

Оглавление

1 О платформе Sherpa RPA	5
1.1 Основные понятия	7
1.2 Компоненты платформы Sherpa RPA	10
1.3 Диаграмма развертывания Sherpa RPA	12
1.4 Безопасность	16
2 Sherpa Robot и Sherpa Assistant	18
2.1 Установка Sherpa Assistant, Sherpa Robot и Sherpa Designer	19
2.2 Регистрация лицензии	21
2.2.1 Автономная активация	23
2.3 Интерфейс Sherpa Assistant	24
2.3.1 Контекстное меню Sherpa Assistant	28
2.3.1.1 Контекстное меню Sherpa Assistant. Логи	29
2.3.1.2 Контекстное меню Sherpa Assistant. Оркестратор	30
2.3.1.3 Контекстное меню Sherpa Assistant. Настройки	30
2.3.1.4 Контекстное меню Sherpa Assistant. Плагины	41
2.3.1.5 Контекстное меню Sherpa Assistant. Инструменты	41
2.3.1.6 Контекстное меню Sherpa Assistant. Удаленная отладка	42
2.4 Работа с Citrix	42
2.5 Внешний мониторинг и логирование	46
2.6 Sherpa Assistant: ответы на часто задаваемые вопросы	50
3 Sherpa Designer	52
3.1 Интерфейс Sherpa Designer	53
3.1.1 Основное меню Sherpa Designer	54
3.1.1.1 Меню «Разработка»	54
3.1.1.2 Меню «Инструменты»	65
3.1.2 Главная панель	73
3.1.2.1 Блоки	73
3.1.2.2Проект	75
3.1.2.3 Библиотека	76
3.1.3 Центральная рабочая область	78
3.1.4 Панель свойств	80
3.1.5 Панель «Переменные»	85
3.1.6 Панель «Лог»	85



3.2 Руководство по веб-селекторам	86
3.2.1 Ручной поиск CSS-селекторов в браузере	87
3.2.2 Ручной поиск CSS-селекторов с помощью Sherpa Explorer	89
3.2.3 Как понять, что селектор не оптимальный?	90
3.2.4 Нотация CSS-селекторов на примерах	91
3.2.5 Справочник по нотации CSS- и XPath-селекторов	94
3.3 Основные функции Sherpa RPA	98
3.3.1 Отладка	98
3.3.2 Запись действий пользователя	102
3.3.2.1 Пример записи действий при работе с Калькулятором	109
3.3.2.2 Особенности работы с Google Chrome	115
3.3.3 Работа с переменными	117
3.3.4 Работа с выражениями	123
3.3.4.1 Функции в выражениях	126
3.3.4.2 Методы в выражениях	129
3.3.5 Примеры использования переменной и работы с выражениями	
3.3.5.1 Простое арифметическое действие	131
3.3.5.2 Сложение числа и строки	132
3.3.5.3 Сложение двух чисел	134
3.3.5.4 Умножение текста на число	135
3.3.5.5 Лог	136
3.3.6 Настройка селекторов	140
3.3.7 Использование ОСК Яндекс	145
3.3.8Примеры автоматизации популярных приложений	150
3.3.8.1 Автоматизация калькулятора	150
3.3.8.2 Точная настройка селектора в Microsoft Word	155
3.3.8.3 Автоматизация выбора шрифта в Блокноте	161
3.3.8.4 Автоматизация поиска в Яндексе	166
3.3.8.5 Извлечения данных из таблиц в Word и Excel	181
3.4 Sherpa Designer: ответы на часто задаваемые вопросы	183
4 Sherpa Orchestrator	186
4.1 Развертывание платформы под управлением Оркестратора	188
4.1.1 Отказоустойчивое развертывание Оркестратора	190
4.2 Vctahorka Sherpa Orchestrator	192



4.2.1 Рекомендации по размещению	192
4.2.2 Установка	193
4.3 Начало работы	195
4.3.1 Использование Attended-роботов в Оркестраторе	195
4.3.2 Использование Unattended-роботов	196
4.3.3 Координатор Unattended-роботов	197
4.3.3.1 Принцип работы Координатора	199
4.3.3.2 Установка Координатора	200
4.3.3.3 Создание работы на экране «Работы»	209
4.3.4 Sherpa RPA Runtime	210
4.3.4.1 Установка Sherpa RPA Runtime	211
4.3 Интерфейс Sherpa Orchestrator	218
4.3.1 Панель «Параметры пользователя»	220
4.3.2 Рабочая область Оркестратора	222
4.3.2.1 Действия с данными таблиц экранов Оркестратора	224
4.3.3 Панель «Экраны»	227
4.3.3.1 Экран «Обзор»	229
4.3.3.2 Экран «Роботы»	234
4.3.3.3 Экран «Процессы»	239
4.3.3.4 Экран «Пакеты»	249
4.3.3.5 Экран «Ресурсы»	252
4.3.3.6 Экран «Хранилище»	261
4.3.3.7 Экран «Работы»	270
4.3.3.8 Экран «Очереди»	279
4.3.3.9 Экран «Действия»	297
4.3.3.10 Экран «Тригтеры»	331
4.3.3.11 Экран «Сообщения Роботов»	339
4.3.3.12 Экран «Аккаунты»	343
4.3.3.13 Экран «Пользователи»	344
4.3.3.14 Экран «Роли»	355
4.3.3.15 Экран «Аудит»	358
4.3.3.16 Экран «Лицензии»	360
4.3.3.17 Экран «АРІ»	370
4.4 Резервное копирование и восстановление	379



4.5 Sherpa Orchestrator: ответы на часто задаваемые вопросы	380
5 Шаблонизатор Sherpa RPA	384
5.1 Интерфейс Шаблонизатора	385
5.1.1 Основное меню Шаблонизатора	387
5.1.1.1 Меню «Файл»	389
5.1.1.2 Меню «Обработка»	390
5.1.1.3 Открыть исходный файл	392
5.1.1.4 Обработать документ	393
5.1.1.5 Проверить	394
5.1.1.6 Добавить атрибут	395
5.1.1.7 Выполнить полное сканирование	396
5.1.1.8 Умный помощник	398
5.1.1.9 Масштаб	400
5.1.1.10 Меню «Отображать»	400
5.1.2 Центральная рабочая область Шаблонизатора	401
5.1.3 Панель настроек	402
5.1.3.1 Настройки	403
5.1.3.2 Якоря	416
5.1.3.3 Атрибуты	421
5.1.3.4 Таблицы	426
5.1.3.5 Чертежи таблиц	431
5.2 Действия в Sherpa Designer после создания шаблона	440
5.2.1 Значения переменных	440
5.2.2 Работа с многостраничными документами	446
5.3 Примеры	449
5.3.1 Создание шаблона на примере документа «Справка о стоимости выполненных работ»	449
5.3.1.1 Предобработка документа	
5.3.1.2 Создание шаблона	
5.3.1.3 Работа с таблицами	
5.3.1.4 Якоря	
5.3.1.5 Создание атрибутов	
5.3.2 Создание шаблона с фиксированными атрибутами на примере док «Счет на оплату»	хумента



1 О платформе Sherpa RPA

Платформа **Sherpa RPA** предназначена для автоматизации рутинных бизнес-процессов с помощью программных роботов.

RPA (Robotic Process Automatisation) — это технология автоматизации бизнес-процессов, которая использует настраиваемых программных роботов.

Программный робот — это программа, способная выполнять стандартные и повторяющиеся операции, которые обычно выполняют сотрудники с использованием клавиатуры, экрана и мыши. Программный робот Sherpa работает с любыми системами на компьютере через графические и командные интерфейсы или прямые обращения к приложениям (офисным и бухгалтерским, веб-ресурсам, базам данных, сервисам электронной почты и т.д.), а также использует сторонние технологии распознавания изображений (OCR-Optical Character Recognition), анализа и синтеза речи, искусственного интеллекта и машинного обучения.

Преимущества использования платформы Sherpa RPA:

- Экономия средств и времени. Платформа Sherpa RPA позволяет экономить средства и время, благодаря передаче программным роботам рутинных бизнес-процессов.
- **Простота внедрения.** Платформа Sherpa RPA не требует существенных изменений в технологической инфраструктуре.
- Простота использования платформы. Платформа Sherpa RPA предоставляет возможность создавать программных роботов без написания программного кода с помощью графического редактора (в режиме No-code), а также создавать программных роботов с помощью Low-Code редактора.
- **Кроссплатформенность.** Платформа Sherpa RPA поддерживает создание и работу программных роботов как на операционной системе Windows, так и на операционной системе Linux.
- **Импортозамещение.** Платформа Sherpa RPA состоит из зарегистрированных и разрешенных реестром российского ПО компонентов.



Платформу Sherpa RPA можно использовать в таких сферах как:

Бухгалтерия

- Оптимизация подготовки актов и счетов;
- Оптимизация выверки счетов;
- Оптимизация формирования отчетов;
- Автоматизация заполнения данных из первичных документов;
- Роботизация обработки транзакций;
- Оптимизация составления финансовых отчетов.

Управление персоналом и HR-менеджмент

- Оптимизация при оформлении приема на работу и увольнении;
- Оптимизация при подготовке рабочих учетных записей для нового сотрудника;
 - Роботизация оформления приказов и справок.

Разработка и тестирование ПО

- Подготовка данных для тестирования;
- Автоматизация тестирования;
- Увеличение скорости анализа данных.

Техническая и клиентская поддержка

- Регистрация поступивших обращений;
- Классификация поступивших обращений;
- Контроль исполнения обращений;
- Аналитика входящих клиентских заявок;
- Улучшение качества клиентского сервиса.

Администрирование и документооборот

- Оптимизация финансового документооборота;
- Оптимизация работы с банковскими выписками;



- Оптимизация работы со сметами;
- Роботизация отчетов отделов предприятия;
- Роботизация взаимодействия с надзорными и госорганами;
- Регистрация и учет входящих документов и обращений;
- Контроль согласования документов;
- Автоматизация проверки договоров на соответствие НПА.

Торговля и закупка

- Оптимизация мониторинга и управления ценами;
- Оптимизация управления остатками на складах и маркетплэйсах;
- Оптимизация взаимодействия с банками;
- Формирование заявок на закупку;
- Проверка закупочной документации;
- Увеличение скорости анализа рынка;
- Сбор и сравнение предложений;
- Автоматизация обработки заказов;
- Синхронизация данных о клиентах в системах предприятия.

Производство

- Оптимизация планирования производства;
- Автоматизация контроля качества на производстве;
- Оптимизация процессов управления цепочками поставок.

1.1 Основные понятия

Платформа Sherpa RPA – платформа, предназначенная для автоматизации рутинных бизнес-процессов с помощью программных роботов.

RPA (Robotic Process Automatisation) — это технология автоматизации бизнес-процессов, которая использует настраиваемых программных роботов.



Sherpa Designer (Дизайнер) — это среда визуальной разработки программных роботов. Именно здесь создается сценарий бизнес-процесса для исполнения роботом.

Sherpa Robot (Робот) — это виртуальный сотрудник, обученный совершать рутинные рабочие задачи в различных информационных системах. Для управления Роботами на локальной машине используется Sherpa Assistant.

Sherpa Assistant (Ассистент) — это программа-агент для выполнения проектов, созданных в Sherpa Designer. Данное приложение позволяет запускать, просматривать список, управлять запуском и планировать исполнение бизнеспроцессов.

Sherpa Orchestrator (Оркестратор) — это инструмент для централизованного подключения и управления программными Роботами, созданными на Платформе Sherpa RPA. Он управляет роботами, определяет их состояние, назначает разным роботам разные задачи, организует очереди выполнения задач и контролирует их исполнение.

Sherpa IDP (Шаблонизатор) — это инструмент создания шаблонов (визуальный редактор) и является неотъемлемой частью Дизайнера Sherpa RPA. Шаблонизатор не привязан к каким-либо облачным сервисам и входит в комплект поставки.

Runtime — это установщик Робота в учетную запись Пользователя для использования без Дизайнера.

Блок — это отдельное действие, шаг бизнес-процесса. Каждый Блок отвечает за уникальное функциональное действие и имеет набор свойств (по сути — настроек), определяющих его поведение. Блоки соединяются между собой с помощью стрелок, которые означают, что после одного действия будет выполнено другое.

Диаграмма — это последовательность действий пользователя, которая автоматизирует определенный бизнес-процесс. Диаграмма состоит из



последовательности Блоков и представляет собой блок-схему выполнения процесса. Диаграммы сохраняются в виде-файлов с расширением «.process».

Проект — это совокупность файлов Диаграмм, одна из которых назначается главной. Обычно Проект состоит из одной главной и нескольких вспомогательных Диаграмм (подпроцессы). Проект может состоять только из одной главной Диаграммы. Проекты сохраняются в виде файлов с расширением «.sherpa».

Процесс — сценарий роботизации, который выполняется Роботами.

Работа – связка из Робота и Процесса либо Робота, Процесса и Задачи. Работу можно создать вручную, с помощью Расписания или с помощью Очереди.

Задача – набор параметров в виде пар Параметр и Значение, которые могут указывать Роботу, что именно ему нужно сделать в рамках выполнения Процесса.

Ресурсы – список объектов, доступный всем Роботам, Группе Роботов или конкретному Роботу. Поддерживаемые типы ресурсов: текст (текстовая строка), реквизиты (пара логин-пароль) и календарь.

Пользователи – это лицо (группа лиц, организация), пользующееся компонентами платформы Sherpa RPA: Sherpa Designer (Дизайнер), Sherpa Robot (Робот) Sherpa Orchestrator (Оркестратор).

Координатор — сервис, устанавливаемый под учетной записью Администратора RDP-сервера (терминального сервера), для запуска Unattended-Роботов, при необходимости автоматического входа в учетные записи Роботов. Для Attended-роботов указывать координатор не требуется. Подробнее о Координаторе в разделе 4.4 Координатор Unattended-Роботов.

Термин Робот имеет несколько определений:

• под программным роботом понимается сценарий в виде последовательности действий пользователя, который создается в среде



визуальной разработки (Sherpa Designer) в виде набора диаграмм (блок-схем) и затем исполняется в программе-агенте (Sherpa Robot).

- **Роботом** называется готовый Проект, который опубликован, то есть упакован в единый файл запуска с расширением «.robot». Файлы готовых Роботов должны быть сохранены в папку «MyRobots» для запуска с помощью-программы агента. Они также могут быть переданы на другой компьютер или запущены на удаленных компьютерах с помощью Оркестратора.
- **Робот** это программа-агент, которая устанавливается на компьютеры и выполняет сценарии.
- Программный робот это программа, способная выполнять стандартные и повторяющиеся операции, которые обычно выполняют сотрудники с использованием клавиатуры, экрана и мыши. Программный робот Sherpa работает с любыми системами на компьютере через графические и командные интерфейсы или прямые обращения к приложениям (офисным и бухгалтерским, веб-ресурсам, базам данных, сервисам электронной почты и т.д.), а также использует сторонние технологии распознавания изображений (ОСR-Optical Character Recognition), анализа и синтеза речи, искусственного интеллекта и машинного обучения.

1.2 Компоненты платформы Sherpa RPA

Платформа Sherpa RPA состоит из следующих компонентов:

Sherpa Designer (Дизайнер) — это среда визуальной разработки программных роботов. Именно здесь создается сценарий бизнес-процесса для исполнения роботом.

Сценарий для робота создается в виде последовательности действий из широкой палитры готовых Блоков или автоматически с помощью инструмента Записи действий пользователя.

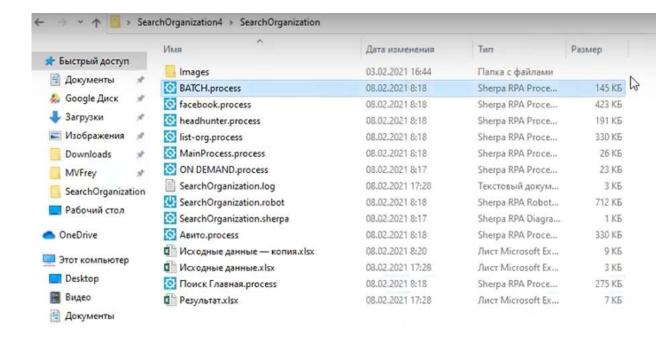
Sherpa Robot (Робот) – программа-агент, которая принимает на вход сценарии бизнес-процесса и исполняет заданные действия. Для управления Роботами на локальной машине используется Sherpa Assistant.



Sherpa Assistant (Ассистент) — это программа-агент для выполнения проектов, созданных в Sherpa Designer. Данное приложение позволяет запускать, просматривать список, управлять запуском и планировать исполнение бизнеспроцессов.

Sherpa Orchestrator (Оркестратор) — это инструмент для централизованного подключения и управления программными Роботами, созданными на Платформе Sherpa RPA. Он управляет роботами, определяет их состояние, назначает разным роботам разные задачи, организует очереди выполнения задач и контролирует их исполнение.

На рисунке 1 представлена типичная папка с проектом робота.



Существуют три типа файлов робота:

Файл с расширением .sherpa – самый главный файл проекта. Для редактирования проекта в Дизайнере необходимо выбрать файл с этим расширением.



Файлы с расширением .process – диаграммы конкретных процессов, из которых состоит проект.

Обратите внимание!

Полностью собранный и отлаженный робот не нуждается в файлах с расширениями .sherpa и .process, так как они нужны только на стадии разработки.

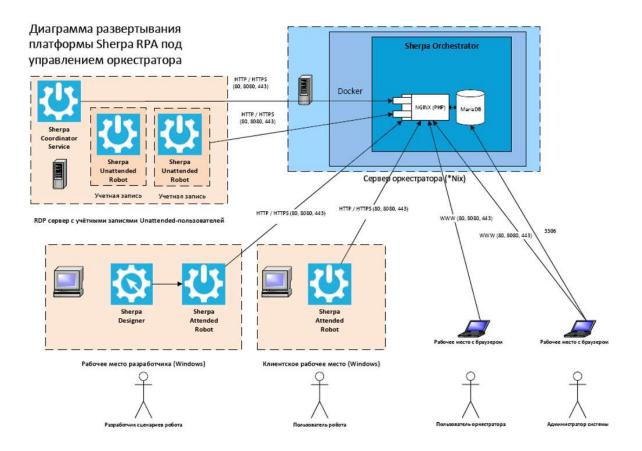
Файл с расширением **.robot** – итоговый файл, представляющий собой собранного робота.

Для получения данного файла необходимо воспользоваться функцией «Опубликовать» (робота), в ходе выполнения которой все исходные файлы собираются в один итоговый. Этот файл подается на вход исполнительной программы – Sherpa Robot.

1.3 Диаграмма развертывания Sherpa RPA

Диаграмма развертывания платформы Sherpa RPA под управлением Оркестратора выглядит следующим образом:





1) Описание

Схема описывает верхнеуровневую реализацию функционала, который обеспечивает автоматизацию бизнес-процессов, выполняемых на пользовательских рабочих местах и терминальных серверах путём роботизации (имитации действий пользователя при работе с информационными системами). Разработчики сценариев роботизации после тестирования передают сценарии в централизованное средство управления – оркестратор.

Бизнес-пользователи на своих рабочих местах по требованию запускают attended-сценарии автоматизации (требующие интерактивного взаимодействия робота или бизнес-процесса с пользователем), запуск осуществляется путём выбора сценария из списка установленных на рабочем месте, нажатия горячей клавиши, также возможен запуск сценариев из локального планировщика Windows и в автозагрузке.

Оркестратор по событиям или входящим данным из внешней среды (через API), по команде робота или бизнес-пользователя или по расписанию запускает 13



unattended-сценарии роботизации (выполняемые в учётных записях терминальных серверов без участия пользователя). Роботы обоих типов обмениваются с оркестратором логами, задачами из очередей, централизованно хранимыми учетными данными и общими данными.

В процессе исполнения сценария робот достигает поставленных задач путём имитации действий пользователя (включая действия с мышью и клавиатурой) в пользовательских интерфейсах десктоп-приложений, вебприложений (исполняемых в браузере), а также путём взаимодействия с операционной системой и установленными приложениями и веб-сервисами с помощью разнообразных программных АРІ.

Уровень доступа робота к информационным системам и контурам клиента ограничивается и определяется уровнем доступа того пользователя, под учётной записью которого робот производит действия, описанные в сценарии. Для unattended-роботов рекомендуется создавать выделенные учетные с уровнями доступа, правами и ролями, определяемыми решаемой данным роботом бизнесзадачей.

2) Конфигурация сети

Конфигурация портов и сетевые протоколы Sherpa RPA могут быть настроены для поддержки всех общих требований брандмауэра. Конфигурация порта по умолчанию выглядит следующим образом:

- *Sherpa Attended Robot, Sherpa Unattended Robot, Sherpa Coordinator исходящие на Sherpa Orchestrator: 80 или 443.
- *Во всех сетевых коммуникациях инициатива установки подключения и первоначального запроса принадлежит только клиентским компонентам, то есть Sherpa Attended Robot, Sherpa Unattended Robot и Sherpa Coordinator. Sherpa Orchestrator по своей инициативе не выполняет запросы к клиентам.
 - *Связь с базой данных: 3306 и 1433-настраивается.



• *Доступ пользователя к веб-интерфейсу Sherpa Orchestrator: 80 или 443.

Для взаимодействия с веб сервером используется https, опционально, возможно, http.

Sherpa RPA поддерживает защищенную связь (с использованием протокола TLS 1.2) между Attended Robot, Sherpa Unattended Robot, Sherpa Coordinator и Sherpa Orchestrator. При установке с помощью TLS клиент должен предоставить необходимые сертификаты, разместив их по пути / opt/app/config/certs/, переименовав их в orchestrator.crt и orchestrator.key.

Все компоненты Sherpa RPA устанавливается локально в сети Заказчика, без связи с внешними серверами или службами SaaS. Возможность и необходимость доступа компонентов платформы к внутренним и внешним системам определяется решаемой в рамках бизнес-процесса задачей.

Развертывание рабочего места разработчика, рабочего места Attended и Unattended-роботов производится вручную с помощью соответствующих ехе-инсталляторов либо автоматически с помощью msi-инсталлятора с использованием механизма GPO.

Установка Sherpa Coordinator Service производится в администраторской учетной записи терминального сервера с помощью ехе-инсталлятора. По умолчанию развертывание Sherpa Orchestrator осуществляется с помощью Docker-контейнера.

3) Механизмы аутентификации

Аутентификация Sherpa Attended Robot, Sherpa Unattended Robot, Sherpa Coordinator в Orchestrator осуществляется с помощью Bearer Token, передаваемого в заголовке запросов. Bearer Token сопоставляется с уникальным GUID каждого экземпляра Sherpa Attended Robot, Sherpa Unattended Robot, Sherpa Coordinator. Для аутентификации пользователей веб-интерфейса Orchestrator применяется



авторизация с помощью пары логинпароль. При повторном входе используется сессионная кука, имеющая ограниченный срок жизни.

Логирование

Для логирования используется компонент Monolog. События аудита и системные ошибки сохраняются в выделенную таблицу базы данных.

1.4 Безопасность

1) Сбор данных

Sherpa RPA не собирает никаких личных данных пользователей и не хранит их у себя в базе, если только это не было прямо предписано сценарием разработанного робота.

Решение поддерживает централизованный сбор и хранение логов, что позволяет проводить мониторинг активности в системе в режиме реального времени. Действия записываются в локальные файлы логов или отправляются в Sherpa Orchestrator, если он был подключен к Sherpa Robot. Логи включают данные об аутентификации, операциях роботов, изменениях конфигурации, а также о любых других действиях, проводимых в системе. Это облегчает процесс аудита и своевременного выявления аномалий.

2) Защита данных

Во всех компонентах решения для защиты конфиденциальных данных используется алгоритм шифрования **AES-256**, который является современным и надежным стандартом для обеспечения безопасности данных.

Для подписания установочных пакетов используется стойкий хэшалгоритм **SHA256**.

Вся передача данных осуществляется по защищенным каналам с использованием протокола TLS 1.3, который обеспечивает высокий уровень защиты от перехвата и атак. Решение также поддерживает совместимость с протоколами SSL3, TLS 1.1, TLS 1.2 и TLS 1.3 для обеспечения максимальной гибкости при интеграции с различными системами.

Для обеспечения безопасного обмена данными между учетными записями системы, роботы могут работать под отдельными, изолированными учетными



записями с минимальными привилегиями. Это исключает необходимость предоставления административных прав, что значительно снижает риск компрометации системы в случае успешной атаки на одну из учетных записей.

Безопасность аутентификации сторонних сервисов обеспечивается компьютером пользователя и поставщиками сервисов. Данные аутентификации сторонних сервисов не передаются серверам Sherpa RPA и не хранятся там.

3) Аутентификация и авторизация

При использовании решения в сочетании с Sherpa Orchestrator применяются продвинутые механизмы аутентификации и авторизации на сервере, что предотвращает несанкционированный доступ. Эти механизмы включают в себя поддержку интеграции с LDAP/OpenID и другие современные методы контроля доступа.

4) Устранение уязвимостей

При обнаружении уязвимости в любом из наших решений, мы, как вендор, незамедлительно уведомляем пользователей, отправляя письма на их электронные адреса. В этих письмах содержится информация о проблеме, сроки её решения, а также инструкции по необходимым действиям для её устранения.

5) Сроки устранения уязвимостей:

- Уязвимости среднего уровня устраняются в течение 30 календарных дней.
- Уязвимости высокого и критического уровня устраняются в течение 7 календарных дней.

6) Работа в закрытом контуре

Все компоненты платформы могут работать в закрытом контуре, без доступа в Интернет. Некоторые отдельные облачные функции и функции сервисов третьих сторон могут требовать доступ в Интернет, если Вы предполагаете их использование. В таком случае необходимо обеспечить доступ в Интернет только для адресов этих сервисов. Для активации лицензий на компоненты платформы в закрытом контуре без доступа в интернет используйте автономную активацию, которая предполагает обмен кодом запроса и кодом ответа с Технической поддержкой Sherpa RPA. Направить код запроса и получить код ответа можно по адресу support@sherparpa.ru. Также Вы можете обеспечить



доступ компонентов платформы к серверу лицензирования Sherpa, расположенному по адресу https://sherpa.wiregeo.com, порт 443. Это позволит проводить автоматическую активацию лицензий Sherpa RPA и пользоваться такими функциями как Умный помощник и запросы к нейросетям **OpenAI**, **Cбер ГигаЧат**, **YandexGPT**, **Groq**, **Claude**.

7) Резервное копирование решения (Sherpa Designer+Sherpa Robot)

Для обеспечения резервного копирования всех данных решения рекомендуется добавить в свою программу резервирования следующий путь: C:\Users\User\AppData\Roaming\Sherpa RPA Data\.

2 Sherpa Robot и Sherpa Assistant

Sherpa Robot принимает на вход сценарии бизнес-процесса и исполняет заданные действия. Робот отвечает за исполнение сценария бизнес-процесса, созданного в Sherpa Designer.

Sherpa Assistant — программа-агент для выполнения проектов, созданных в Sherpa Designer. Приложение позволяет запускать, просматривать список, управлять запуском и планировать исполнение роботов.

Системные требования для установки Sherpa Robot и Sherpa Assistant

Минимальные системные	Рекомендуемые системные требования
требования	
Операционная система: Windows 7	Операционная система: Windows 10 / 11
и выше или Windows Server 2008R2	или Windows Server 2012 и выше
и выше	
Процессор: 1.4GHz	Процессор: 3GHz x 2 ядра
Оперативная память: 2 GB	Оперативная память: 4 GB
Microsoft .NET Framework 4.8+	Microsoft .NET Framework 4.8+ PowerShell
PowerShell 5.1+	5.1+

Системные требования для Sherpa RPA для Linux

Минимальные системные	Рекомендуемые системные требования
требования	

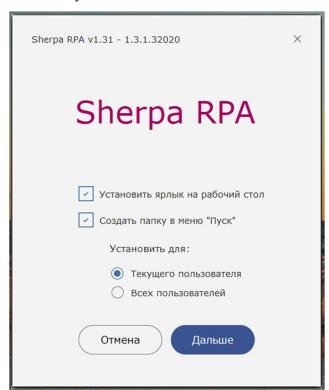


Операционная система: ОС Astra	Операционная система: ОС Astra Linux
Linux Common Edition 2.12.46.6 и	Common Edition 2.12.46.6 и выше
выше	
Прогоссор, 1 АСИ	Промодор: 2СНд у 2 дира
Процессор: 1.4GHz	Процессор: 3GHz x 2 ядра
Оперативная память: 2 GB	Оперативная память: 4 GB
	_
Microsoft .NET Framework 4.8+	Microsoft .NET Framework 4.8+ PowerShell
PowerShell 5.1+	5.1+

2.1 Установка Sherpa Assistant, Sherpa Robot и Sherpa Designer

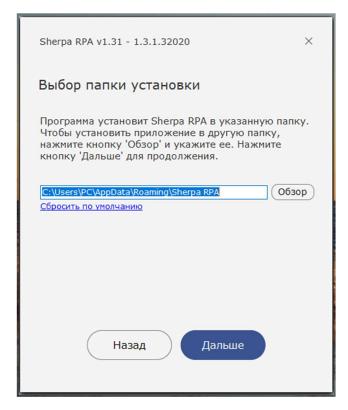
Sherpa Assistant, включая Sherpa Robot, устанавливаются одновременно с Sherpa Designer. Для их установки необходимо скачать исполняемый файл. Установку программ осуществляет инсталлятор, запускаемый файлом «SherpaRPA.exe». Сохраните данный файл на компьютер и запустите его. В открывшемся диалоговом окне необходимо указать параметры установки:

- установить ярлык на рабочий стол;
- создать папку в меню «Пуск»;
- Установить для: текущего пользователя или всех пользователей.

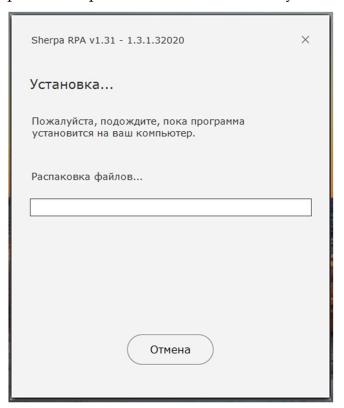




Затем необходимо нажать кнопку «Дальше» и выбрать папку для установки программы.



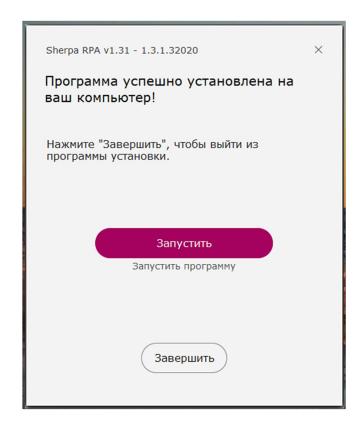
Затем необходимо нажать кнопку «Дальше». После этого инсталлятор начнет установку программ, которая займет несколько минут.



После завершения установки, инсталлятор информирует пользователя с помощью диалогового окна, в котором можно нажать кнопку «Завершить», чтобы 20



завершить процесс установки, либо сразу запустить программу с помощью кнопки «Запустить».



После успешной установки на рабочем столе появятся два ярлыка: Sherpa Designer и Sherpa Assistant.



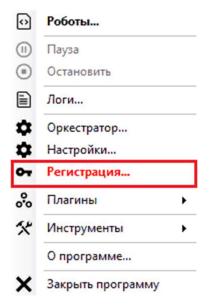
2.2 Регистрация лицензии

Чтобы зарегистрировать лицензию робота, нужно предварительно завершить работу Sherpa Designer (если он был запущен). Затем запустить Sherpa Assistant с помощью ярлыка на рабочем столе.

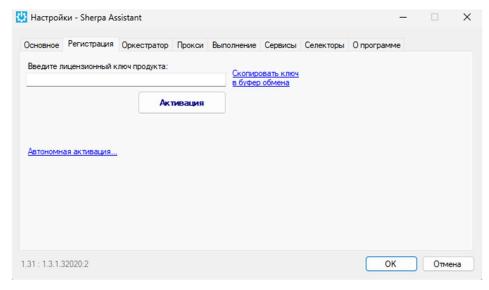




В системном трее необходимо найти значок и нажать на него правой кнопкой мыши, после чего откроется контекстное меню, в нем необходимо выбрать пункт «Регистрация...».

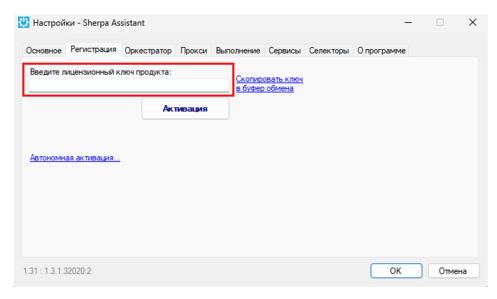


Окно Sherpa Assistant откроется на вкладке «Регистрация».



Затем необходимо ввести лицензионный ключ продукта в предназначенное для этого текстовое поле.

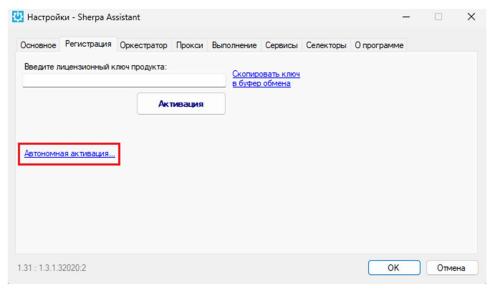




После ввода лицензионного ключа продукта необходимо нажать на кнопку «Активация», а затем на кнопку «ОК». На этом регистрация лицензии завершена.

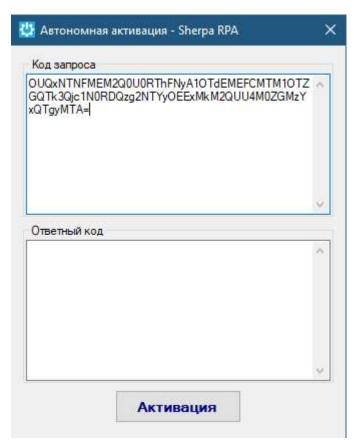
2.2.1 Автономная активация

Автономная активация лицензионного ключа производится также во вкладке «Регистрация» в настройках Sherpa Assistant. Для автономной активации необходимо нажать на ссылку «Автономная активация...».



Затем откроется окно «Автономная активация».





Затем необходимо скопировать значение из верхнего поля «Код запроса» и прислать на e-mail: info@sherparpa.ru. В ответ будет выслан «Ответный код», который необходимо вставить в нижнее поле «Ответный код», а затем нажать кнопку «Активация». Автономная лицензия активирована.

2.3 Интерфейс Sherpa Assistant

После установки и запуска Sherpa Assistant в системном трее отображается значок . Для того, чтобы открыть главный интерфейс Sherpa Assistant с перечнем всех доступных для запуска проектов (роботов) необходимо нажать на в системном трее, либо выбрать в контекстном меню пункт «Роботы».

Внимание! Проекты и роботы при публикации из Sherpa Designer должны быть сохранены в программную папку «MyRobots».

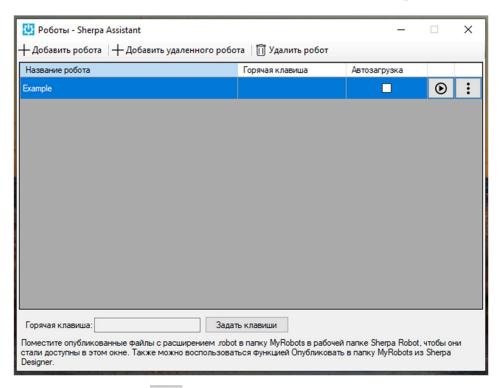
Элементы главного интерфейса Sherpa Assistant.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	кнопка «Добавить	Позволяет добавить робота из папки
	робота»	«MyRobots».

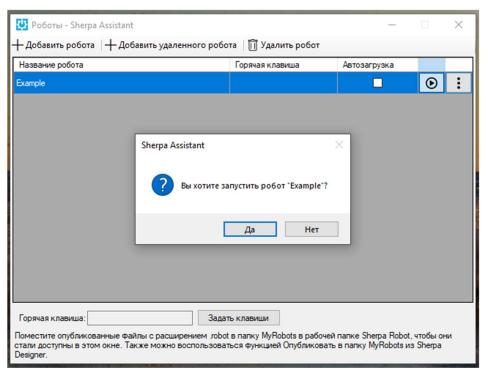


2.	кнопка «Добавить	Позволяет добавить удаленного робота из
	удаленного робота»	папки «MyRobots2.
3.	колонка «Название робота»	Отображает название проекта.
4.	колонка «Горячая клавиша»	Позволяет просмотреть сочетания клавиш, по которому можно запустить проект (робота).
5.	колонка «Автозагрузка»	Позволяет отметить флагом проекты или роботов, которые будут запускаться вместе с запуском программы-агента. Для того, чтобы процесс запускался при запуске Windows требуется добавить программу-агент в список автозагрузки. Если выделить несколько роботов, то они будут запускаться последовательно по очереди.
6.	кнопка	Позволяет запустить процесс или программного робота при нажатии.
7.	кнопка	Позволяет вызвать контекстное меню с доступными опциями для выбранного проекта (робота).
8.	поле «Горячая клавиша»	Позволяет указать горячие клавиши для выбранного проекта (робота).
9.	кнопка «Задать клавиши»	Позволяет сохранить горячие клавиши для выбранного проекта (робота).



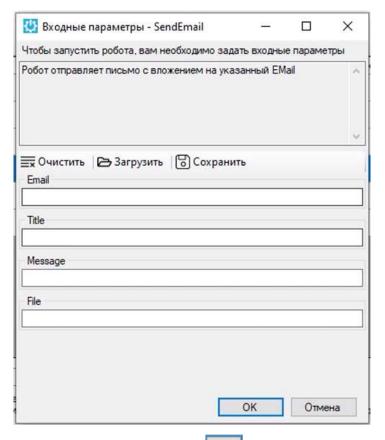


При нажатии на кнопку появится диалоговое окно, предлагающее запустить выбранного робота.

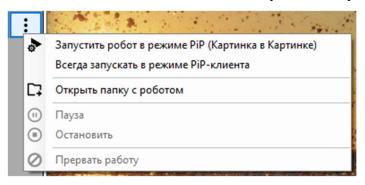


Если в свойствах робота указаны входные параметры, то откроется окно с полями для ввода значений.



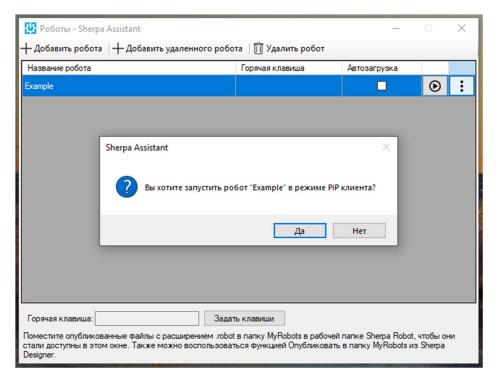


Для каждого робота в списке по кнопке доступны следующие опции.

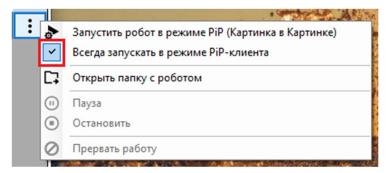


Опция «Запустить робот в режиме PiP (Картинка в Картинке)» позволяет запустить робот в режиме Картинка в картинке с предварительным подтверждением в диалоговом окне.





Опция «Всегда запускать в режиме PiP-клиента» позволяет включить функцию запуска в режиме PiP-клиента всегда. При включенной функции рядом с опцией появляется галочка.



Опция «Открыть папку с роботом» позволяет открыть папку с роботом на Вашем компьютере. Опции «Пауза», «Остановить» и «Прервать работу» активны в случае, если один или несколько роботов уже запущены.

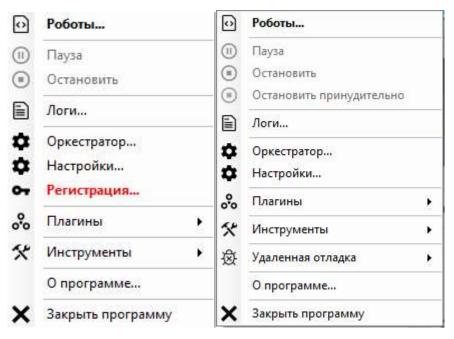
2.3.1 Контекстное меню Sherpa Assistant

При нажатии правой кнопкой мыши на значок в системном трее открывается контекстное меню, которое позволяет получить доступ к следующим пунктам:

- Роботы...;
- Пауза (активен при запущенном проекте (роботе));
- Остановить (активен при запущенном проекте (роботе));

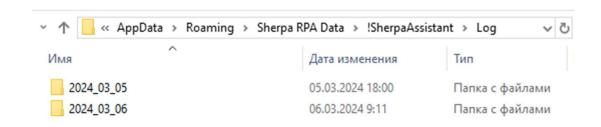


- Остановить принудительно (активен при запущенном проекте (роботе));
 - Логи...;
 - Оркестратор...;
 - Настройки...;
 - Регистрация (если она еще не зарегистирована);
 - Плагины;
 - Инструменты;
 - Удаленная отладка;
 - О программе...;
 - Закрыть программу.



2.3.1.1 Контекстное меню Sherpa Assistant. Логи.

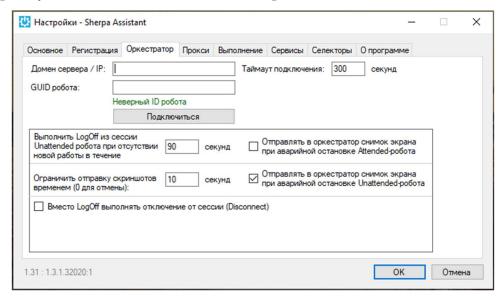
Пункт контекстного меню «Логи...» позволяет открыть программную папку «Sherpa Assistant» с сохраненными в ней логами. Логи, в свою очередь, сохраняются в отдельные папки с указанием даты сохранения.





2.3.1.2 Контекстное меню Sherpa Assistant. Оркестратор.

Пункт контекстного меню «Оркестратор...» позволяет открыть окно настроек на вкладке «Оркестратор». В данном окне можно настроить Оркестратор в случае, если он подключен. Также, вкладку «Оркестратор» можно открыть через пункт контекстного меню «Настройки...».



2.3.1.3 Контекстное меню Sherpa Assistant. Настройки

Пункт контекстного меню «Настройки...» позволяет открыть окно настроек. Окно «Настройки - Sherpa Assistant» содержит следующие вкладки:

- Основное;
- Регистрация;
- Оркестратор (вкладку можно открыть также через пункт контекстного меню «Оркестратор...»);
 - Прокси;
 - Выполнение;
 - Сервисы;
 - Селекторы;
 - О программе.

Контекстное меню Sherpa Assistant. Настройки. Вкладка «Основное»

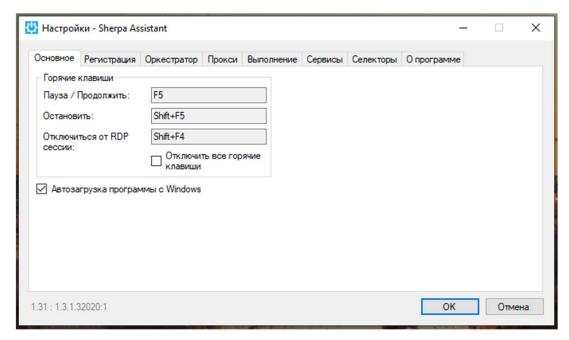
Вкладка «Основное» позволяет настроить «Горячие клавиши» для таких функций как:

- Пауза/продолжить;
- Остановить;



• Отключиться от RDP сессии.

Данная вкладка также позволяет отключить все горячие клавиши. Для этого необходимо установить флаг «Отключить все горячие клавиши». Также, во вкладке «Основное» доступно управление автоматическим запуском программы с Windows. Для этого необходимо установить флаг «Автозагрузка программы с Windows».



Контекстное меню Sherpa Assistant. Настройки. Вкладка «Регистрация»

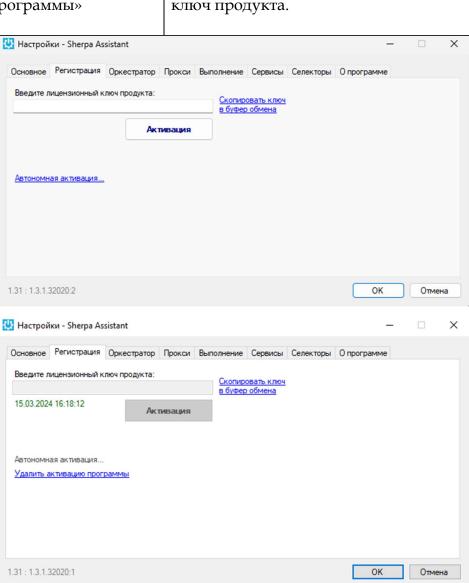
Вкладка «Регистрация» в окне настроек имеет два вида, в зависимости от того, активирована у Вас лицензия или нет.

Элементы интерфейса вкладки «Регистрация».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	текстовое поле «Введите лицензионный ключ продукта»	Поле для ввода лицензионного ключа продукта.
2.	Дата и время	Если у вас уже активирован лицензионный ключ продукта, то под полем для ввода лицензионного ключа будут указаны дата и время окончания срока действия лицензии.



3.	опция «Скопировать	Позволяет скопировать лицензионный ключ
	ключ в буфер	продукта в буфер обмена.
	обмена»	
4.	кнопка «Активация»	Позволяет активировать лицензионный
		ключ продукта.
5.	опция «Автономная	Позволяет перейти к автономной активации
	активация»	лицензии.
6.	опция «Удалить	Позволяет удалить активацию программы,
	активацию	если у вас уже активирован лицензионный
	программы»	ключ продукта.
I		

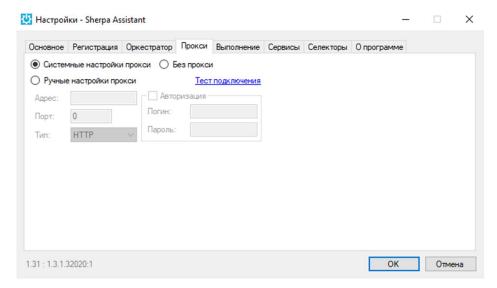




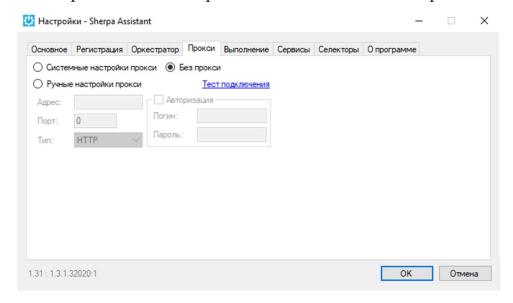
Контекстное меню Sherpa Assistant. Настройки. Вкладка «Прокси»

Вкладка «Прокси» в окне настроек позволяет управлять настройками прокси. Можно использовать системные настройки прокси, а так выбрать режим «Без прокси» или настроить прокси вручную. В зависимости от выбора режима будут активны разные поля, которые необходимо заполнить. При выборе любого из режимов можно запустить проверку настроек прокси с помощью опции «Тест подключения».

Режим «Системные настройки прокси» позволяет включить системные настройки прокси.



Режим «Без прокси» позволяет работать без использования прокси.

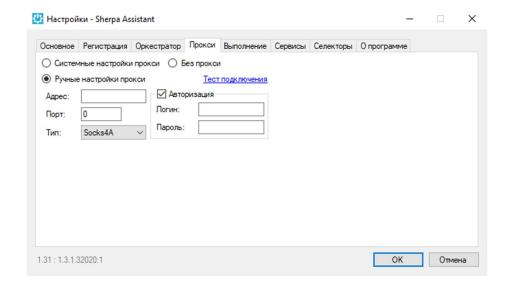


Режим «Ручные настройки прокси» позволяет настроить прокси вручную, используя следующие настройки.



Ручные настройки прокси.

№ п/п	Настройка	Описание настройки
1.	текстовое поле «Адрес»	Поле для записи адреса прокси.
2.	текстовое поле «Порт»	Поле для записи порта прокси.
3.	выпадающий список «Тип»	Позволяет выбрать из выпадающего списка способ прокси-соединения: • HTTP; • Socks4; • Socks4A; • Socks5.
4.	флаг «Авторизация»	Позволяет включить авторизацию при выборе ручных настроек прокси.
4.1	текстовое поле «Логин»	Поле для записи логина.
4.2	текстовое поле «Пароль»	Поле для записи пароля.



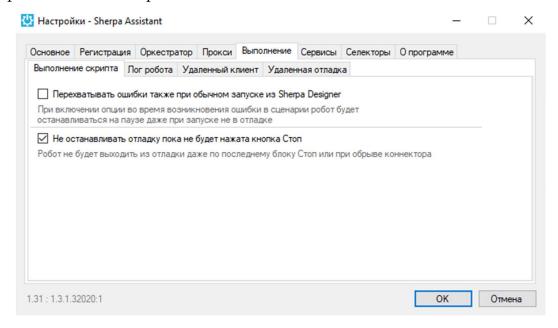


Контекстное меню Sherpa Assistant. Настройки. Вкладка «Выполнение»

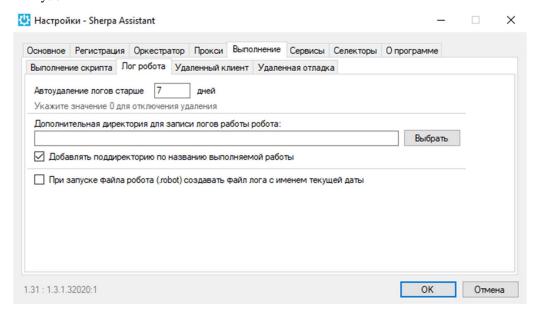
Вкладка «Выполнение» в окне настроек состоит из 4 внутренних вкладок:

- Выполнение скрипта;
- Лог робота;
- Удаленный клиент;
- Удаленная отладка.

Внутренняя вкладка «Выполнение скрипта» позволяет настроить параметры выполнения скрипта.

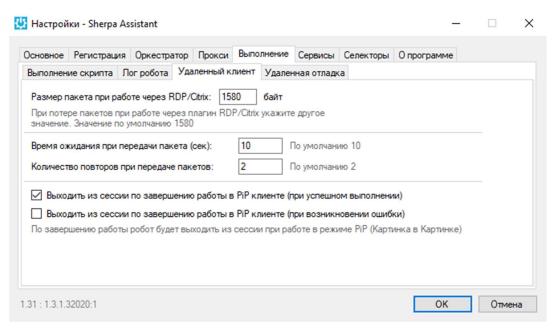


Внутренняя вкладка «Лог робота» позволяет настроить параметры сохранения и удаления логов.

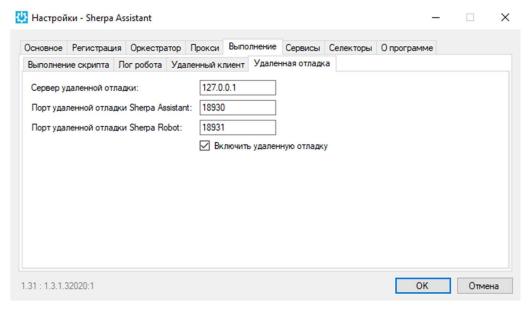


Внутренняя вкладка «Удаленный клиент» позволяет настроить параметры работы передачи пакетов для работы удаленного клиента.





Внутренняя вкладка «Удаленная отладка» позволяет настроить параметры удаленной отладки и ее запуск.



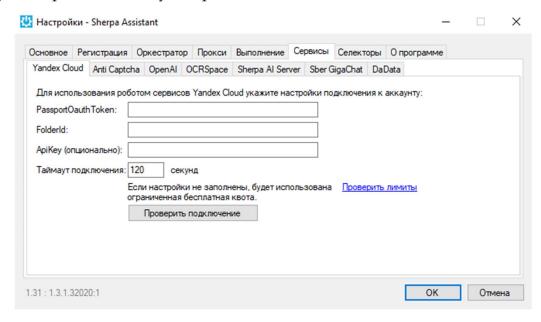
Контекстное меню Sherpa Assistant. Настройки. Вкладка «Сервисы»

Вкладка «Сервисы» в окне настроек состоит из 7 внутренних вкладок:

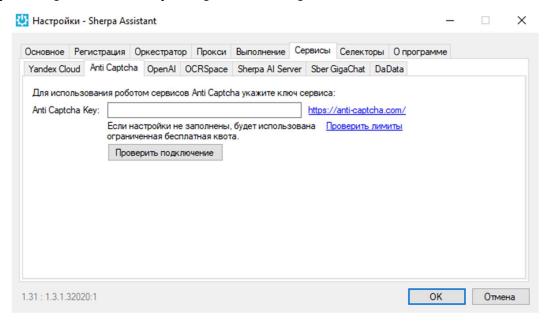
- Yandex Cloud;
- Anti Captcha;
- OpenAI;
- OCRSpace;
- Sherpa AI Server;
- Sber GigaChat;
- DaData.



Внутренняя вкладка «Yandex Cloud» позволяет настроить подключение к аккаунту, если робот использует сервис Yandex Cloud.

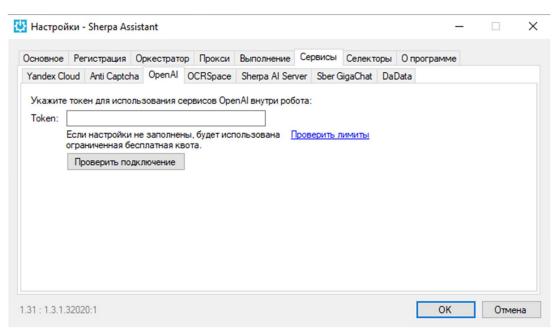


Внутренняя вкладка «Anti Captcha» позволяет настроить подключение к аккаунту, если робот использует сервис Anti Captcha.

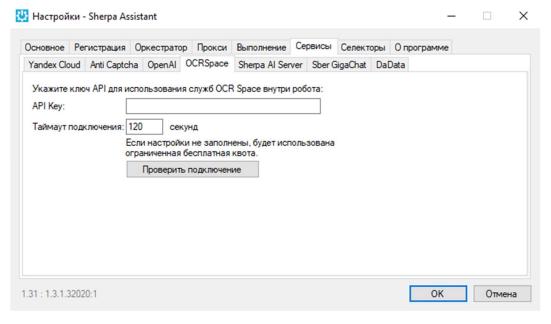


Внутренняя вкладка «OpenAI» позволяет настроить подключение к аккаунту, если робот использует сервис Open AI.



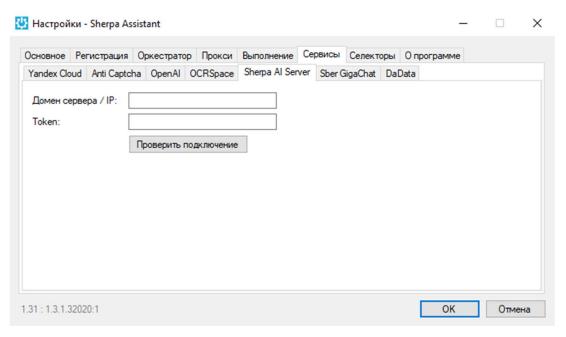


Внутренняя вкладка «OCRSpace» позволяет настроить подключение к аккаунту, если робот использует службы OCR Space.

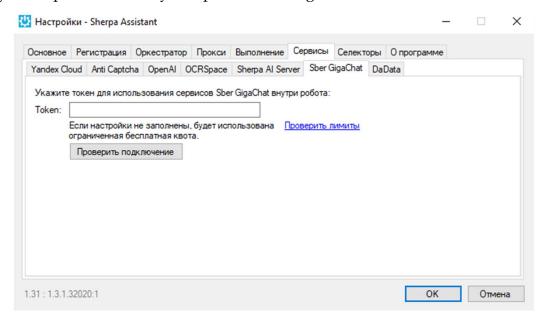


Внутренняя вкладка «Sherpa AI Server» позволяет настроить подключение к аккаунту, если робот использует Sherpa AI Server.



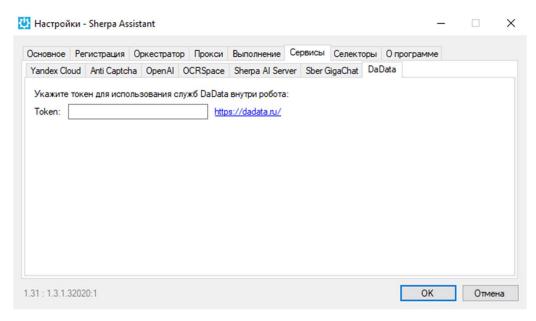


Внутренняя вкладка «Sber GigaChat» позволяет настроить подключение к аккаунту, если робот использует сервисы Sber GigaChat.



Внутренняя вкладка «DaData» позволяет настроить подключение к аккаунту, если робот использует службы DaData.



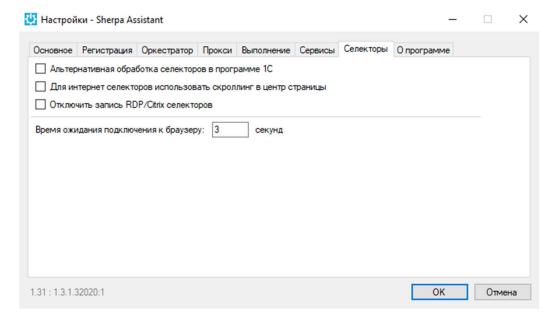


Контекстное меню Sherpa Assistant. Настройки. Вкладка «Селекторы»

Вкладка «Селекторы» в окне настроек позволяет управлять настройками селекторов. В данной вкладке с помощью установки флагов можно выбрать один из параметров настройки селекторов:

- Альтернативная обработка селекторов в программе 1С;
- Для интернет селекторов использовать скроллинг в центр страницы;
- Отключить запись RPD/Citrix селекторов.

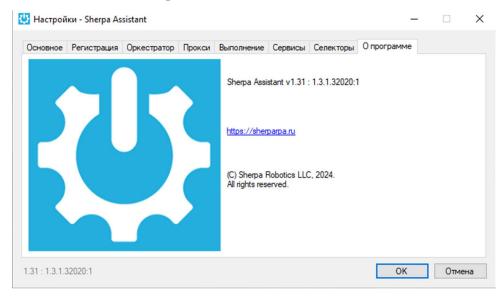
В текстовом поле «Время ожидания подключения к браузеру» при необходимости можно указать время ожидания подключения к браузеру в секундах.





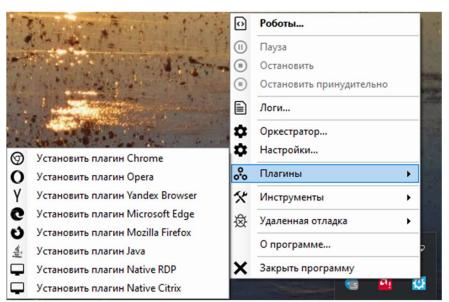
Контекстное меню Sherpa Assistant. Настройки. Вкладка «О программе»

Вкладка «О программе» в окне настроек позволяет просмотреть данные о текущей версии Sherpa Assistant. Также, попасть во вкладку «О программе» можно сразу из контекстного меню Sherpa Assistant.



2.3.1.4 Контекстное меню Sherpa Assistant. Плагины

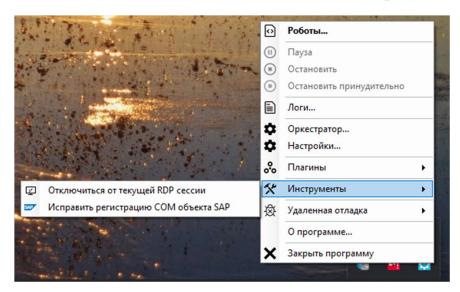
Пункт контекстного меню «Плагины» позволяет добавить плагины в браузеры, если это необходимо.



2.3.1.5 Контекстное меню Sherpa Assistant. Инструменты

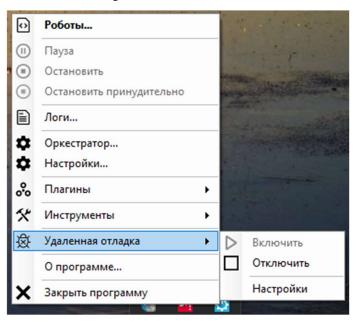
Пункт контекстного меню «Инструменты» позволяет отключиться от текущей RDP-сессии и исправить регистрацию COM объекта SAP.





2.3.1.6 Контекстное меню Sherpa Assistant. Удаленная отладка

Пункт контекстного меню «Инструменты» позволяет включить или отключить удаленную отладку, а также перейти в настройки удаленной отладки во вкладке «Выполнение» окна настроек.



2.4 Работа с Citrix

Для настройки корректной работы роботов через Citrix необходимо иметь клиентский компьютер, на котором будет запускаться Робот и удаленный рабочий стол (или удаленное оконное приложение), доступ к которому будет осуществляться с помощью Citrix. На клиентском компьютере можно установить любой тип Робота (Sherpa Designer + Attended Robot, просто Sherpa Attended Robot или Sherpa Unattended Robot). Именно с клиентского компьютера будет



запускаться и работать сценарий работы. Далее следуйте инструкции.

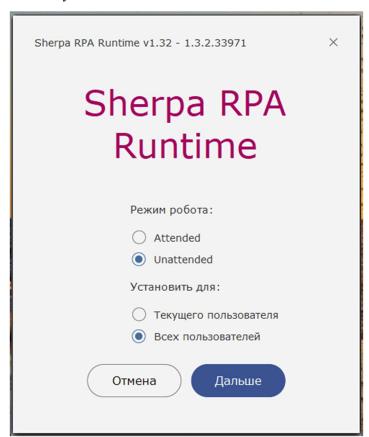
1) Установите SherpaRPARuntime.exe в режиме Attended для всех пользователей на компьютер, к которому будет осуществляться удаленное подключение.

Установку программы осуществляет инсталлятор, запускаемый файлом «SherpaRPARuntime.exe». Сохраните данный файл на компьютер и запустите его.

В открывшемся диалоговом окне отметьте необходимые вам параметры установки:

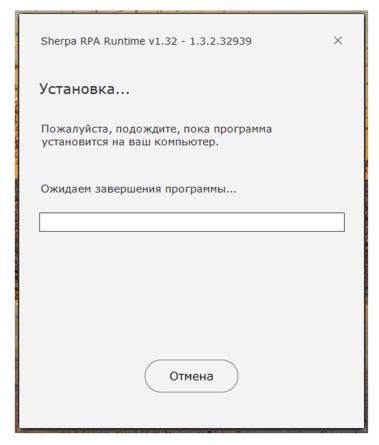
- Режим робота: «Attended»;
- Установить для: «Всех пользователей».

Затем нажмите кнопку «Дальше».

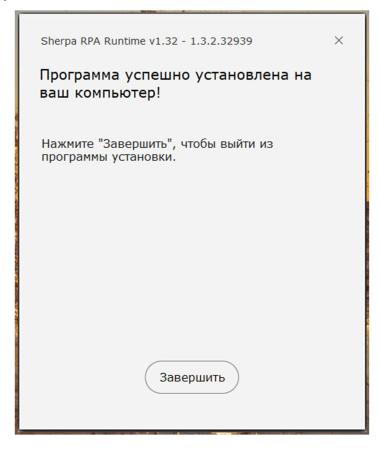


После этого инсталлятор начнет установку программы, которая займет несколько минут.





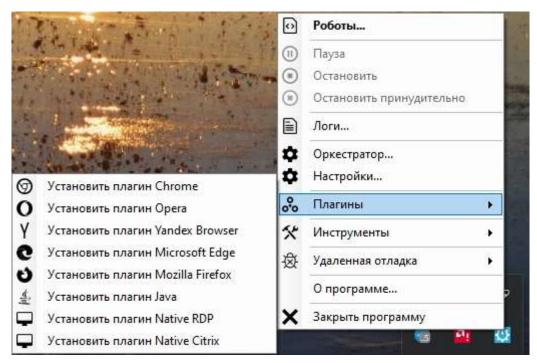
Когда установка будет завершена, инсталлятор проинформирует Вас с помощью диалогового окна, в котором можно нажать кнопку «Завершить», чтобы завершить процесс установки.





Важно! Обязательно перезагрузите ферму (Citrix farm), без перезагрузки селекторы работать не будут.

2) Установите плагин Native Citrix на компьютер, с которого будет запускаться робот. Для этого запустите Sherpa Assistant, затем нажмите в трее правой кнопкой мыши на значок выберите в контекстном меню, пункт «Плагины», далее – «Установить плагин Native Citrix».



Примечание: установку плагина желательно осуществлять от имени администратора (для дальнейшей работы права администратора не нужны).

После этого Робот с клиентской стороны может записывать и воспроизводить действия с помощью селекторов в окнах Citrix.

Далее всё работает автоматически. Например, вы на клиентской стороне запускаете Дизайнер, с помощью инструмента записи действий пользователя пытаетесь работать внутри окна Citrix, либо внутри оконного приложения. Citrix-рекордер это сразу распознает, все элементы внутри Citrix-окна подсвечиваются как обычно и селекторы записываются как обычно. Сам записанный селектор будет содержать параметр Citrix в первой строке, в остальном он будет выглядеть как обычный селектор. В Дизайнере, в палитре блоков «Citrix/RDP» есть блоки, которые позволяют как создавать подключение, так и подключаться к уже существующей Citrix-сессии (однако, если Вы обычно создает Citrix-подключение



через отдельный портал, то Вам необходимо именно подключаться к уже существующему подключению).

Робот при запуске исполняет записанный сценарий как обычно: сам понимает, что надо сделать на клиентской стороне, а что на удаленной стороне Технически и архитектурно, используются именованные каналы передачи данных внутри самого Citrix-соединения, по ним локальный робот транслирует удаленному Sherpa Runtime (фактически это такой же робот) команды, которые нужно исполнить, и они исполняются на удаленной стороне. В данном случае, открытие портов не требуется, всё идёт по самому Citrix-соединению. По такому же принципу работает передача файлов в/из соединения и исполнение сценариев на удаленной стороне - блоки для выполнения этих действий блоки также доступны в Дизайнере, палитре «RDP/Citrix». В целом, реализация этого функционала на платформе Sherpa RPA идентична реализации UIPath.

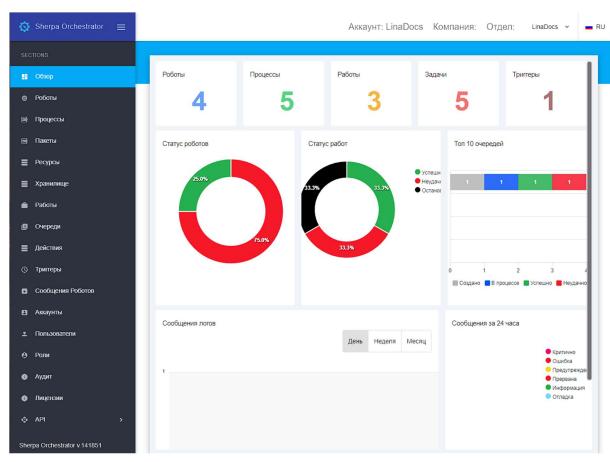
2.5 Внешний мониторинг и логирование

Логирование на платформе Sherpa RPA можно настроить двумя разными способами:

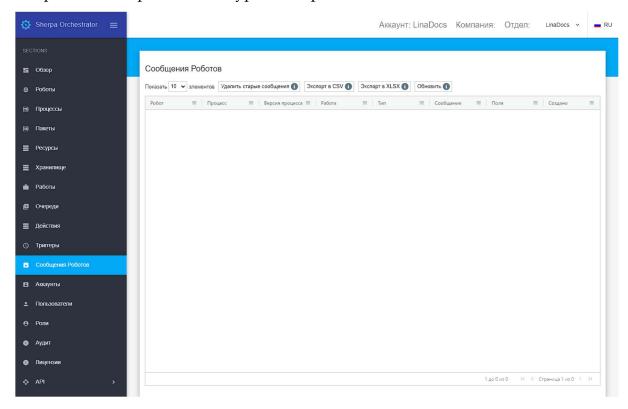
1) Логирование через Sherpa Orchestrator

Для логирования через Оркестратор можно использовать дашборд «Обзор».





или экран «Сообщения Роботов», который позволяет настроить на экране «Пользователи» для каждого отдельного пользователя email-уведомления о событиях роботов с определенным уровнем тревожности.





Также на уровне Оркестратора логирование можно настроить через SQLзапросы в таблицу Оркестратора, где хранятся сообщения роботов.

Важно! При настройке логирования через SQL-запросы необходимо следить за временем выполнения запроса в статистике медленных запросов, чтобы не перегружать Оркестратор.

2) Логирование с помощью NLog

На уровне робота можно использовать логирование с помощью NLog. NLog интегрирован с роботом и позволяет настроить автоматическую отправку логов можно в несколько мест.

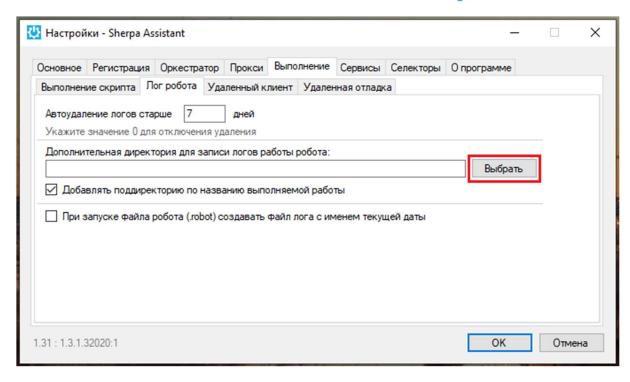
Подробнее об этом можно почитать по ссылке: https://nlogproject.org/config/

Для настройки логирования с помощью NLog:

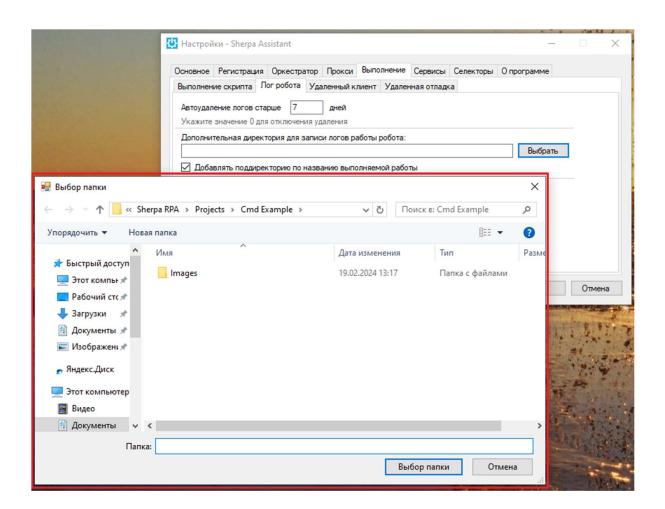
- откройте папку: c:\Users\User\AppData\Roaming\Sherpa RPA
 Data\Setting\
 - найдите в ней файл: nlog.config.xml
 - в файле укажите нужную Вам конфигурацию.

Также функционал платформы Sherpa RPA позволяет указать в настройках робота дополнительную папку для хранения копий текстовых логов. Для этого откройте настройки Sherpa Assistant и во вкладке «Выполнение» выберите дополнительную вкладку «Лог робота». В поле «Дополнительная директория для записи логов работы робота» по кнопке «Выбрать» укажите нужную папку.





и, в открывшемся окне, выберите папку на Вашем компьютере.





Примечание: также можно указать сетевую папку, куда будут сохраняться логи роботов.

Типичный путь к логам координатора выглядит так:

C:\ProgramData\Sherpa RPA Coordinator\Log\SherpaRPAService.log Типичный путь к логам робота:

C:\Users\<имя пользователя>\AppData\Roaming\Sherpa RPA Data\Log
Далее в каждом из вариантов находите папки с текущей датой, где хранятся
текстовые файлы с временем запуска робота.

2.6 Sherpa Assistant: ответы на часто задаваемые вопросы

Вопросы:

- 1) Почему при запуске робота на удаленной машине, после отключения от нее, робот перестает работать?
- 2) Иногда на удаленной машине возникают ошибки работы селекторов. Отключаюсь от RDP с помощью программы closerdpsession, но это не помогает.

Ответ. Возможные решения:

- 1) Запустите ассистент и в трее найдите пункт меню: «Инструменты» -> «Отключиться от текущей RDP сессии».
- 2) На рабочем столе правой кнопкой мышки кликнуть по ярлыку Sherpa Assistant и перейти в папку с программой. Там найти программу closerdpsession.exe и сделать на нее ярлык на рабочий стол. Выходить из подключения не по закрытию окна RDP, а этой программой.

Так можно выйти только с одной машины. Для запуска программы требуются права администратора.

Мы не гарантируем полную и 100% работу робота на RDP машине, когда Вы отключаетесь от нее через Ассистент или программу 'closerdpsession.exe'. При отключении происходит переключение сессии пользователя на консоль. Это, так сказать «костыль», который используется, чтобы контекст пользователя оставался открытым. Однако, как ведет себя консоль и что с ней может случится со временем никто не знает, эту информацию Microsoft не разглашает. Также, следует учитывать, что так, как консоль одна и, если, к примеру другой



пользователь зашел на эту же машину также по RDP или другим методом, то Windows отключает консоль от подключенного пользователя.

Используйте Unattended-роботов, у которых такой проблемы нет.

Вопрос: может ли робот работать при заблокированном экране?

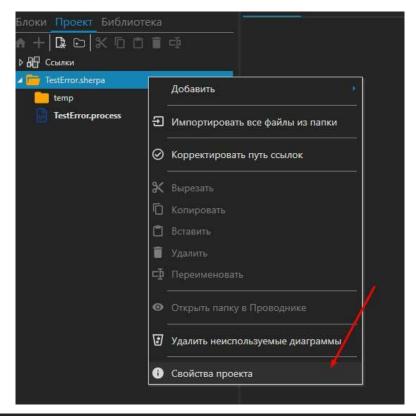
Ответ: да, может. Есть нюансы, в частности, интерактивная работа с GUI невозможна из сценария робота, если самого интерфейса не видно. Но! Если графический контекст сессии переключить на консоль через closerdp, то для пользователя будет выглядеть, как будто, экран заблокирован, но при этом работа с GUI корректно продолжится.

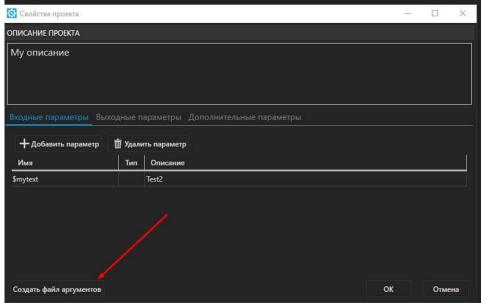
Вопрос: у меня есть приложение написанное на С#. В определенный момент мне нужно запускать робота из кода, и также нужно передать ему параметры определенные. Могу ли я запускать файл «.robot» и передавать параметры для запуска?

Ответ: следуйте инструкции:

- 1) Зайдите в свойство проекта.
- 2) Укажите входные параметры.
- 3) По кнопке «Создать файл аргументов» создайте файл. Это будет json. Переменные должны быть глобальными.
- 4) Пропишите в этом файле нужные Вам параметры и далее запускайте робот так: ProcessAgent.exe -args "путь к файлу аргументов" "путь к файлу .robot".







Путь к файлу ProcessAgent.exe можно найти, кликнув правой кнопкой мыши по ярлыку Sherpa Assistant на рабочем столе и выбрать «Расположение файла».

3 Sherpa Designer

Sherpa Designer — это среда визуальной разработки программных роботов. Именно здесь создается сценарий бизнес-процесса для исполнения роботом.



Sherpa Designer устанавливается одновременно с Assistant и Sherpa Robot. Процесс установки описан в п. 2.1.

Системные требования для установки Sherpa Designer

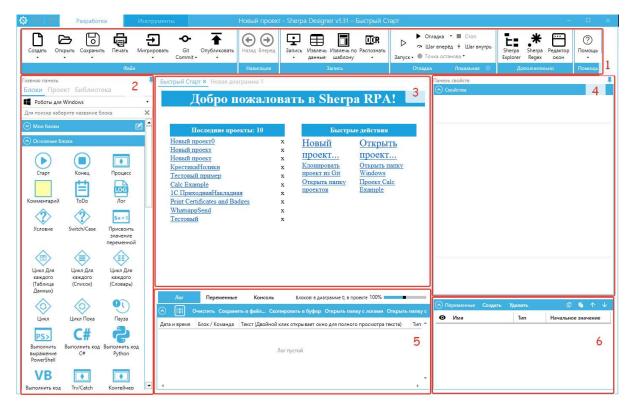
Минимальные системные	Рекомендуемые системные требования
требования	
Операционная система: Windows 7	Операционная система: Windows 10 / 11
и выше или Windows Server 2008R2	или Windows Server 2012 и выше
и выше	
Процессор: 1.4GHz	Процессор: 3GHz x 2 ядра
Оперативная память: 2 GB	Оперативная память: 4 GB
Microsoft .NET Framework 4.8+	Microsoft .NET Framework 4.8+ PowerShell
PowerShell 5.1+	5.1+

3.1 Интерфейс Sherpa Designer

Пользовательский интерфейс Sherpa Designer, открываемый по умолчанию, состоит из основных областей:

- 1. Основное меню (верхняя панель) предлагает основные действия по созданию проекта и диаграмм.
 - 2. Главная панель (левая панель) содержит:
 - Палитры функциональных Блоков для создания проектов;
 - Область работы с Проектами;
 - Библиотека готовых процессов.
- 3. **Центральная рабочая область** область для работы с диаграммой текущего процесса.
 - 4. Панель свойств (правая верхняя панель).
 - 5. Панель «Переменные» (правая нижняя панель).
 - 6. Панель «Лог» лог всех действий программного робота.





3.1.1 Основное меню Sherpa Designer

Основное меню Sherpa Designer содержит две вкладки: «Разработка» и «Инструменты».

3.1.1.1. Меню «Разработка»



Меню «Разработка» содержит основные действия по настройке и управлению проектами и диаграммами роботов. В основное меню входит:

- Меню «Файл»;
- Меню «Навигация»;
- Меню «Запись»;
- Меню «Отладка»;
- Меню «Дополнительно»;
- Меню «Помощь».



Элементы интерфейса меню «Разработка».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	меню «Файл»	Содержит действия, необходимые для работы с проектами.
1.1	кнопка «Создать» Создать Открыть Со Новый проект Новая диаграмма	Позволяет создать новый проект или диаграмму, а также по умолчанию создает новую диаграмму в текущем проекте.
1.2.	кнопка «Открыть»	Позволяет открыть созданные ранее диаграмму или проект.
1.3.	Сохранить диаграмму Сохранить диаграмму как Сохранить проект как Сохранить всё Сохранить и закрыть проект	Позволяет сохранить текущую диаграмму или проект. • «Сохранить диаграмму» позволяет сохранить диаграмму из активной вкладки под текущим именем. Также, для сохранения диаграммы можно воспользоваться сочетанием клавиш Ctrl+S. • «Сохранить диаграмму как» позволяет сохранить диаграмму под новым именем в соответствующей папке. • «Сохранить проект как» позволяет сохранить файл проект под новым именем в соответствующей папке.



1.4.	кнопка «Печать»	«Сохранить всё» позволяет сохранить все открытые диаграммы и проект. «Сохранить и закрыть проект» позволяет сохранить открытый проект и закрывает его. Позволяет отправить на печать
	Kifofika «Fe latb»	текущую диаграмму или сохранить ее в формате PDF-файла.
1.5.	жнопка «Мигрировать» ———————————————————————————————————	Позволяет мигрировать (перенести) проект с UIPath или Blue Prism.
1.6.	кнопка «Git commit» Git Commit Git Push Git Pull Настройки Git Клонировать репозиторий Копировать проект в Git	Позволяет выполнять команды Git Commit:



Опция «Клонировать репозиторий» позволяет клонировать удаленный репозиторий Git на локальный компьютер. При выборе данной опции откроется диалоговое окно, в котором необходимо указать URL и путь для сохранения клонированного репозитория, а также по кнопке «Настройки Git» можно изменить настройки авторизации Git. Опция «Клонировать проект Git» позволяет клонировать текущий проект в существующий репозиторий Git. При выборе данной опции на Вашем компьютере откроется папка для выбора директории с существующим репозиторием. 1.7. кнопка «Опубликовать» Позволяет упаковать текущий проект в один файл запуска с Назад Вперед расширением «.robot». и сохраняет в Опубликовать в оркестратор... папку «MyRobots» или другую Опубликовать в папку проекта выбранную пользователем папку. Опубликовать в папку MyRobots Опубликовать на Рабочий стол «Опубликовать в Опубликовать на удаленную машину оркестратор...» позволяет опубликовать файл робота в Sherpa Orchestrator. «Опубликовать в папку проекта» позволяет опубликовать



		файл робота в папку текущего проекта. • «Опубликовать в папку МуRobots» позволяет опубликовать файл робота в папку МуRobots. • «Опубликовать на
		Рабочий стол» позволяет опубликовать файл робота на Рабочий стол. • «Опубликовать на удаленную машину» позволяет опубликовать файл робота на выбранном удаленном компьютере.
2.	меню «Навигация» Назад Вперед Навигация	Позволяет осуществлять навигацию между диаграммами роботов.
3.	меню «Запись»	Содержит действия, необходимые при работе с записью действий пользователя в приложениях и на веб-ресурсах, а также с извлечением данных Шаблонизатором.
3.1.	кнопка «Запись»	Позволяет записать действия пользователя в приложениях или вебсайтах и вставить их в текущую диаграмму как блоки робота. Функция Записи действий пользователя для



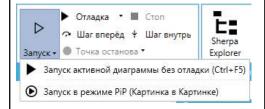
		автоматизированного создания
	Запись Извлечь Извлечь п	бизнес-процессов является важной
	данные шаблону	частью Sherpa Designer.
	 Запись Записать изображение 	По клику на иконку вызывается
	2 & Записать IE	полное меню инструмента «Запись» с
	③ Записать Chrome	возможностью записывать клики на
		кнопки в программах или нужных
		элементах экрана, эмулировать
		нажатия клавиш клавиатуры и т.д.
		Меню инструмента Записи
		появляется в левой части экрана, при
		этом окно Sherpa Designer
		сворачивается.
		Инструменты «Записать
		изображение», «Записать IE»,
		«Записать Chrome», «Записать SAP
		GUI», «Записать Native RDP» и
		«Записать Java» позволяют
		записывать действия пользователя в
		соответствующих программах.
		Важно! Для работы этих
		инструментов необходимо
		предварительно установить
		соответствующие плагины через Sherpa
		Assistant.
3.2.	кнопка «Извлечь данные»	Позволяет извлечь
		структурированную информацию
		(таблицы, списки, похожие элементы)
		из приложений (Word, Excel и т.д.) и
		веб-страниц в браузере и затем
		вставить в текущую диаграмму блок,
		y ,y - , , Fyy



		который автоматизирует этот процесс.
3.3.	кнопка «Извлечь по шаблону»	Позволяет создавать интеллектуальные шаблоны для извлечения структурированной и полуструктурированной информации (таблиц, атрибутов) из документов (PDF, PNG, JPG и т.д.). Запускает Шаблонизатор. Подробнее о Шаблонизаторе в разделе 5.
3.4.	КНОПКА «Распознать» Отладка	Позволяет распознать документ, область экрана или окно приложения с помощью одного из доступных движков ОСR, затем найти и кликнуть на заданное слово или записать распознанный текст в переменную. • Клик по тексту на экране (ОСR); • Распознать текст с экрана (ОСR); • Распознать текст из изображения (ОСR).
4.	меню «Отладка» Отладка	Содержит действия, необходимые при запуске отладки робота, а также удаленной отладке робота.



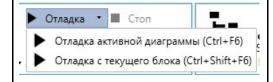
4.1. кнопка «Запуск»



Позволяет запустить выполнение проекта (главной и вложенных Диаграмм) роботом без отладки или приостановить его выполнение. По данной кнопке активируется работа агента Sherpa Robot, а окно Sherpa Designer сворачивается. Также, при нажатии на кнопку "Запуск" происходит автоматическое сохранение всего проекта.

- «Запуск активной диаграммы без отладки» позволяет активную диаграмму без отладки. Также, для выполнения запуска активной диаграммы без отладки можно воспользоваться сочетанием клавиш Ctrl+F5.
- «Запуск в режиме PiP» позволяет запустить выполнение проекта с отображением в отдельном окне запуска робота.

4.2. кнопка «Отладка»



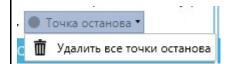
Позволяет запустить текущий проект в режиме отладки. В отличие от режима обычного запуска, доступна остановка выполнения процесса на точках останова и выполнение процесса по шагам. Также, при нажатии на кнопку «Отладка» происходит автоматическое сохранение всего проекта.



• Вариант «Отладка активной диаграммы» позволяет запустить отладку активной диаграммы. Также, для выполнения отладки активной диаграммы можно воспользоваться сочетанием клавиш Ctrl+F6.

• Вариант «Отладка с текущего блока» позволяет запустить отладку с текущего блока. Также, для выполнения отладки с текущего блока можно воспользоваться сочетанием клавиш Ctrl+shift+F6.

4.3. кнопка «Точка останова»



Позволяет пометить текущий блок точкой остановки или снять пометку. **Точка останова** — это блок, на котором робот приостановит выполнение проекта. С помощью кнопки «Точка останова» до или во время запуска проекта можно пометить нужные блоки, на которых процесс должен быть приостановлен - и процесс автоматически встанет на паузу, когда его выполнение дойдет до этих блоков. Помеченные точкой останова блоки обводятся красным прямоугольником. Дополнительно, тот блок, на котором выполнение программы остановлено в данный момент, обводится синим прямоугольником.



Пока исполнение приостановлено, можно просмотреть текущие значения переменных на панели «Переменные». Во время остановки можно убрать точку останова с одного блока и поставить на другой, или же поставить точки останова на несколько блоков. Во время приостановки процесса можно совершать и другие отладочные действия, такие как «Шаг внутрь», «Шаг вперед», или «Стоп», чтобы завершить работу робота и вернуться в режим разработки. 4.4. кнопка «Шаг вперед» Позволяет запустить или продолжить выполнение проекта до следующего Отладка Шаг вперёд Шаг внутрь блока и, затем, поставить выполнение Запуск 🕶 🛑 Точка останова на паузу, но для блоков Подпроцессов следующая остановка будет произведена в текущей диаграмме на следующем после подпроцесса блоке (то есть весь подпроцесс выполнится полностью). 4.5. кнопка «Шаг внутрь» Позволяет запустить или продолжить выполнение проекта до следующего Шаг внутрь блока и, затем, поставить выполнение Точка останова на паузу, но для блоков подпроцессов Отладка следующая остановка будет произведена на первом блоке диаграммы подпроцесса.

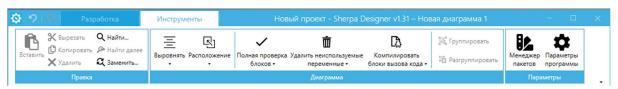


4.6. 5.	кнопка «Стоп» Отладка Стоп Шаг вперёд У Шаг внутрь Запуск Точка останова Отладка Меню «Дополнительно»	Позволяет завершить работу робота и вернуться в режим разработки. Содержит дополнительные действия
J.	меню «дополнительно»	для работы с диаграммами роботов.
5.1.	кнопка «Sherpa Explorer»	Sherpa Explorer – инструмент для ручного подбора и тестирования селекторов. В нём, как и в обычном редакторе селекторов можно записать любой элемент интерактивно, а также исследовать иерархическое дерево любого Desktop- и вебприложения, его доступные атрибуты, подобрать и протестировать нужное сочетание атрибутов.
5.2.	кнопка «Sherpa Regex»	Sherpa Regex – это мощный инструмент для работы с регулярными выражениями. Он предоставляет удобные средства для составления и проверки регулярных выражений. Инструмент разделяет различные группы выражений для облегчения выбора соответствующего шаблона. Кроме того, Sherpa Regex включает умного помощника, который способен составить регулярное выражение в



		соответствии с запросом пользователя.
5.3.	кнопка «Редактор окон»	Позволяет создавать свои собственные окна и отображать их в модальном и немодальном режиме, в том числе асинхронно по отношению к ходу сценария.
6.	меню «Помощь»	Позволяет обратиться к обучающим материалам: обучающим видео, примерам по работе с программой, демонстрационным проектам.

3.1.1.2. Меню «Инструменты»



Меню «Инструменты» содержит действия необходимые при работе с блоками диаграммы робота и при настройки интерфейса Sherpa Designer:

- Меню «Правка»;
- Меню «Диаграмма»;
- Меню «Параметры»;
- Меню «Инструменты».

Элементы интерфейса меню «Инструменты».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	меню «Правка»	Содержит действия,
		необходимые при работе с
		блоками диаграммы робота.



	Вставить Ж Вырезать Q Найти Копировать Я Найти далее Удалить З Заменить	
1.1.	кнопка «Вставить»	Позволяет вставить ранее скопированные блоки в текущую диаграмму. Также, можно воспользоваться сочетанием клавиш Ctrl+V.
1.2.	кнопка «Вырезать»	Позволяет вырезать выделенные блоки в буфер обмена. Также, можно воспользоваться сочетанием клавиш Ctrl+X.
1.3.	кнопка «Копировать»	Позволяет скопировать выделенные блоки в буфер обмена. Также, можно воспользоваться сочетанием клавиш Ctrl+C.
1.4.	кнопка «Удалить»	Позволяет удалить выделенные блоки. Также, можно воспользоваться клавишей Delete.
1.5.	кнопка «Найти…»	Позволяет найти блок, переменную или значение выражения. Также, можно воспользоваться сочетанием клавиш Ctrl+F.



1.6.	кнопка «Найти далее»	Позволяет найти далее по ранее начатому поиску. Также, можно воспользоваться клавишей F3.
1.7.	кнопка «Заменить»	Позволяет заменить значение в названии блока, переменную, значение в выражении. Также, можно воспользоваться сочетанием клавиш Ctrl+H.
2.	меню «Диаграмма»	Содержит действия, необходимые при построении с диаграммы робота.
2.1.	КНОПКА «Выровнять» Инструменты Новый проек Выровнять Расположение Полная проверка Удал блоков ▼ Выровнять по горизонтали Выровнять по вертикали Выровнять по сетке	Позволяет выровнять выделенные блоки по заданному правилу.
2.2.	КНОПКА «РАСПОЛОЖЕНИЕ» Расположение Полная проверка Удалить неисполь блоков переменны Назад Позади всех Авторасстановка всех блоков Автосоединение всех блоков в стиле "Дракон"	Позволяет изменить порядок пересекающихся блоков.
2.3.	кнопка «Полная проверка блоков»	Позволяет выполнить полную проверку свойств у блоков.

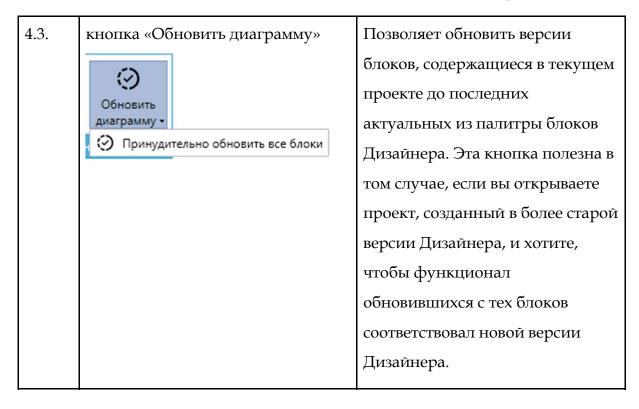


		Г
	Полная проверка Удалить неисполь блоков ▼ переменные ✓ Только в текущей диаграмме	
2.4.	кнопка «Удалить неиспользуемые переменные» Г. Удалить неиспользуемые Компил блоки вы: Только в текущей диаграмме	Позволяет найти и удалить неиспользуемые переменные.
2.5.	кнопка «Компилировать блоки вызова кода» Компилировать блоки вызова кода Разгрупг Только в текущей диаграмме	Позволяет компилировать все блоки вызова кода в проекте. Данное действие ускоряет работу этих блоков.
2.6.	кнопка «Группировать»	Позволяет сгруппировать выделенные блоки, чтобы перемещать их как единое целое.
2.7.	кнопка «Разгруппировать»	Позволяет разгруппировать ранее сгруппированные блоки.
3.	меню «Параметры» Менеджер Параметры пакетов программы Параметры	Содержит действия, необходимые при работе с параметрами программы.
3.1.	кнопка «Менеджер пакетов»	Позволяет открыть окно «Менеджера пакетов». Эта



		функция предназначена для управления пользовательскими палитрами блоков как в локальном варианте, так и загружать их из Sherpa Orchestrator. Подробнее описано
3.2.	кнопка «Параметры программы»	ниже в данном разделе. Позволяет открыть окно «Параметры программы». Подробнее описано ниже в данном разделе.
4.	меню «Инструменты»	Содержит действия, необходимые для удобной настройки интерфейса.
4.1.	Русский (Россия) – язык Русский Тема Обновить диаграмму • Английский (США) Португальский (Португалия) Русский (Россия)	Позволяет изменить язык интерфейса для Дизайнера диаграмм и инструмента Записи, а также программы-агента, установленной на Вашем компьютере, выбрав один из доступных языков: английский, португальский и русский. По умолчанию стоит русский язык.
4.2.	кнопка «Тема» Тема Темная	Позволяет переключить тему оформления интерфейса. Доступны светлая и темная темы. Светлая тема стоит по умолчанию.





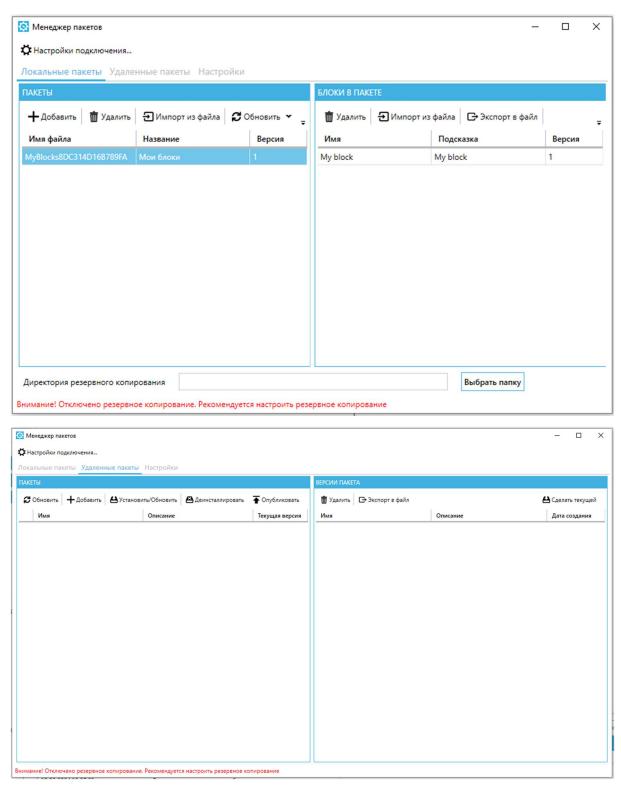
Менеджер пакетов

Кнопка «Менеджер пакетов» в меню инструментов позволяет открыть окно «Менеджера пакетов». Эта функция предназначена для управления пользовательскими палитрами блоков как в локальном варианте, так и загружать их из Sherpa Orchestrator.

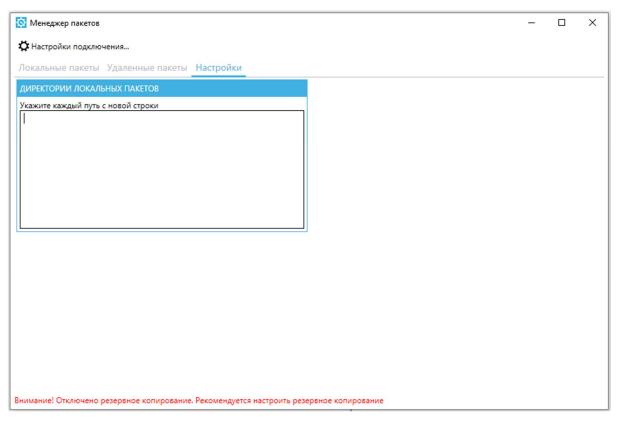
Окно «Менеджера пакетов» содержит три вкладки:

- Локальные пакеты;
- Удаленные пакеты;
- Настройки.







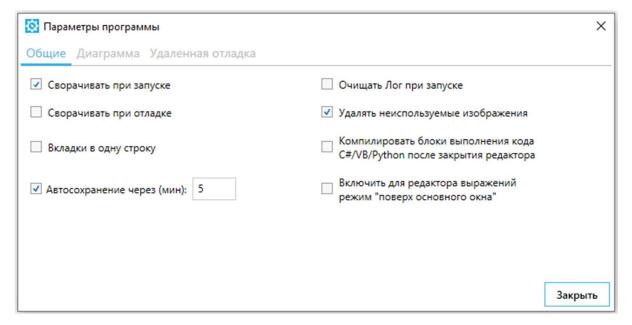


Параметры программы

Кнопка «Параметры программы» в меню инструментов позволяет открыть окно «Параметры программы».

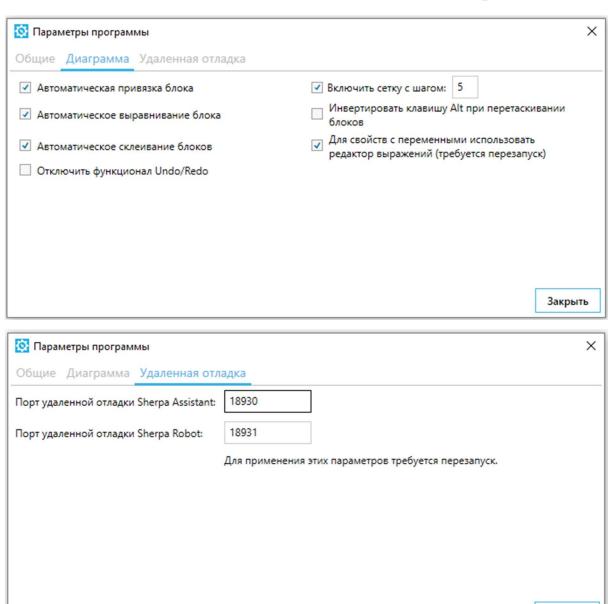
Окно «Параметры программы» содержит три вкладки:

- Общие;
- Диаграмма;
- Удаленная отладка.





Закрыть



3.1.2 Главная панель

3.1.2.1 Блоки

Блоки - основные функциональные единицы программного робота.

Блоки, из которых будет состоять процесс, можно выбрать и перенести в диаграмму процесса по двойному щелчку левой кнопкой мыши или с помощью перетаскивания. Некоторые блоки также создаются автоматически, если вы воспользуетесь инструментом «Запись» в верхней панели разработки Sherpa Designer.

Доступно более сотни блоков, которые позволяют выполнять разнообразные действия в операционной системе и других программах, манипулировать данными, производить вычисления и принимать решения. Для



удобства блоки сгруппированы в палитры, также пользователь может создавать свои блоки и собирать свои палитры блоков.

Сложность блоков варьируется от простого задания задержки перед кликом до полной замены всех вхождений одного текста на другой внутри документа Word без потери форматирования документа.

Примечание: подробное описание каждого блока представлено в «Справочнике блоков» и в программе.





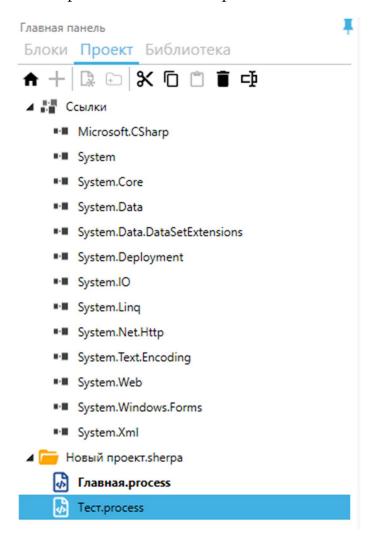
Обратите внимание! Для работы с блоками необходимо выбрать для какой системы будут создаваться роботы — Windows или Linux.



3.1.2.2 Проект

Проект — совокупность файлов диаграмм, в котором одна из диаграмм назначена главной. Проект описывается файлом с расширением «.sherpa». В нем содержится перечисление путей (относительных файлу проекта) с входящими в него диаграммами, а также помечено, какая из диаграмм является главной, и содержатся другие настройки проекта.

На вкладке «Проект» можно просматривать диаграммы текущего проекта, а также добавлять, удалять, переименовывать диаграммы и изменять их порядок.





Одновременно в программе может быть открыт только один проект. При открытии другого проекта предыдущий будет закрыт. Допускается открытие в программе файлов диаграмм, которые не являются частью текущего проекта.

Существуют три типа файлов робота:

- Файл с расширением .sherpa файл проекта. Для редактирования проекта в Дизайнере необходимо выбрать файл с этим расширением.
- Файлы с расширением .process диаграммы конкретных процессов, из которых состоит проект.
- Файл с расширением .robot итоговый файл, представляющий собой собранного робота. Для получения данного файла необходимо воспользоваться функцией "Опубликовать", в ходе выполнения которой все исходные файлы собираются в один итоговый. Этот файл подается на вход исполнительной программы Sherpa Robot.

Обратите внимание! Полностью собранный и отлаженный робот не нуждается в файлах с расширениями .sherpa и .process, они нужны только на стадии разработки.

3.1.2.3 Библиотека

Библиотека содержит готовые стандартные наборы блоков — предустановленные шаблоны для популярных действий в различных программах, а также предоставляет возможность создавать и сохранять собственные шаблоны для использования в текущем или будущих проектах автоматизации.

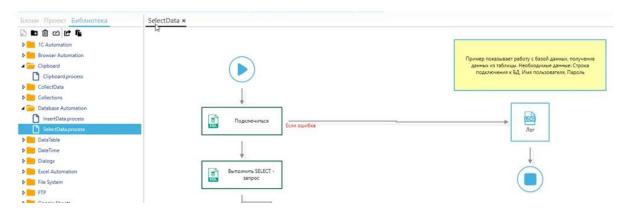




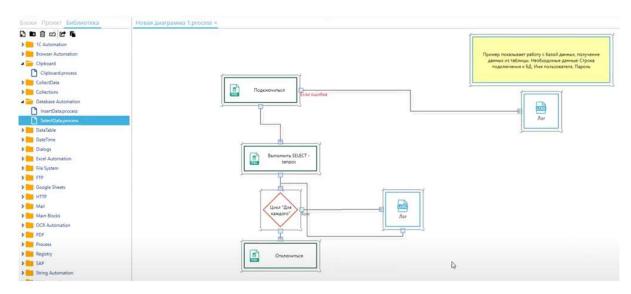
В библиотеке могут быть сохранены не только последовательности блоков, но и целые процессы с переменными и свойствами. Готовы фрагменты проектов позволяет повторно использовать созданные ранее удачные последовательности блоков, тем самым увеличивая скорость разработки новых проектов. В библиотеку по умолчанию добавлены процессы автоматизации, которые охватывают определенные виды действий, например, действия, связанные с манипулированием файлами PDF, Excel, почтой и т.д.

Вы можете добавить заготовки из библиотеки в текущую диаграмму, просто перетащив элемент библиотеки на свободное место диаграммы. При перетаскивании заготовки из библиотеки размещаются внутри текущей диаграммы.





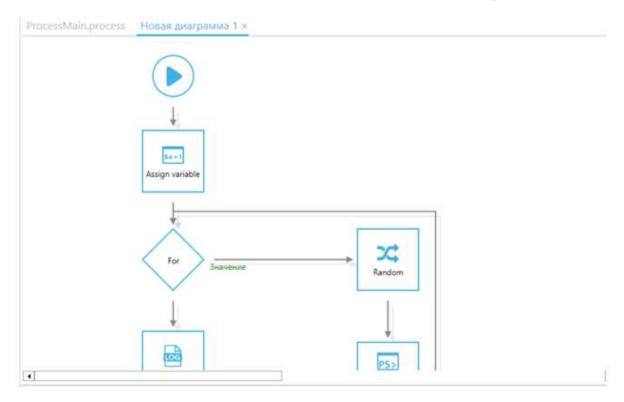
Вы также можете открыть заготовку библиотеки как самостоятельную диаграмму в отдельной вкладке Дизайнера, дважды кликнув по ней. В этом случае заготовка откроется как отдельная диаграмма в составе проекта. Затем вы можете добавить эту заготовку в ваш текущий проект.



3.1.3 Центральная рабочая область

В центральной рабочей области Дизайнера создается диаграмма процесса автоматизации, то есть сценарий автоматизируемого бизнес-процесса в виде блоксхемы.





Для создания диаграммы применяется так называемое «визуальное программирование». Пользователь добавляет в рабочую область функциональные блоки, каждый из которых выполняет отдельное действие бизнес-процесса.

Блоки добавляются в проект перетаскиванием из палитры блоков или по двойному клику. Каждый блок содержит набор свойств, которые определяют поведение блока, и настраиваются в соответствии с задачами бизнес-процесса.

В центральной области отображается вкладка с диаграммой процесса текущего проекта. В случае создания нового проекта данная область будет пустой. Если вы открываете проект, над которым уже работали вы или кто-то из ваших коллег, то в данной области вы увидите ранее разработанную последовательность действий робота.

Порядок действий робота определяется связями — стрелками между блоками.

- Чтобы изменять порядок действий робота, перетащите блок в необходимое место с использованием левой кнопки мыши.
- Чтобы создать связь между блоками, кликните и потащите стрелку от выхода одного блока (обозначенного как квадратная точка) к входу другого блока.



Каждый блок имеет определенное количество входов и выходов. Ко входу блока может быть протянуто неограниченное количество связей от других блоков, но от каждого конкретного выхода блока может быть протянута только одна связь к другому блоку.

Обычно блок имеет только один вход (сверху блока) и один выход (снизу блока). Однако блоки циклов, условных операторов и некоторых других блоков могут иметь несколько выходов, что позволяет в зависимости от выполнения или невыполнения заданных условий продолжать сценарий по одному или другому маршруту, а также закольцовывать части сценария, чтобы они повторялись.

Если необходимо удалить блок, то это можно сделать, нажав кнопку «Удалить» на панели инструментов, либо щелкнув правой кнопкой мыши на блоке и выбрав пункт «Удалить» из выпадающего меню.

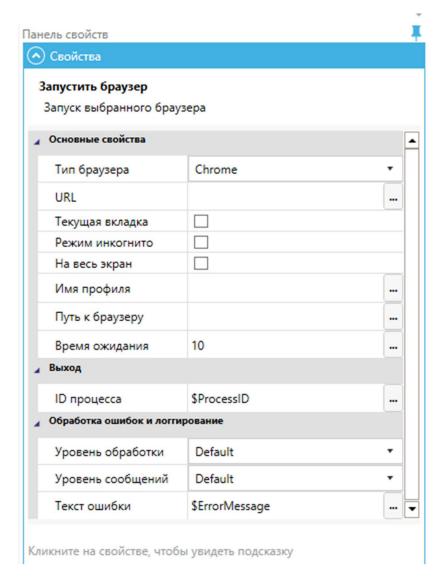
Можно одновременно создавать несколько разных диаграмм в «Рабочей области» и переключаться между ними с помощью закладок в верхней части области.

3.1.4 Панель свойств

При создании нового блока диаграммы необходимо задать определенные свойства блока, которые определяют его поведение. Панель свойств блоков позволяет просматривать и изменять свойства выбранного блока.

На скриншоте приведен пример свойств блока «Запустить браузер».





Свойства блоков в их большинстве относятся к двум категориям:

- Main properties предназначены для получения значений, необходимых для работы блока (входные данные);
- Output предназначены для указания того, куда поместить выходные данных, в таких свойствах нужно указать переменные, в которые будут сохраняться результаты работы блока.

Свойства блоков, определяющие их работу, могут быть заданы:

- Константами. Числа записываются как есть, а текстовые значения записываются в двойных кавычках.
 - Переменными, имена которых должны начинаться со знака \$.
- Выражениями константами и переменными, функциями, методами их сочетаниями.



Для того, чтобы изменить свойство необходимо ввести значение в строке справа от названия свойства или использовать более удобный вариант – окно «Редактировать выражение». Для вызова окна «Редактировать выражение» необходимо нажать на кнопку с многоточием в правой части строки «Свойства».

Окно «Редактировать выражение» содержит три вкладки:

- Быстрая вставка;
- Умный помощник;
- Переменные.

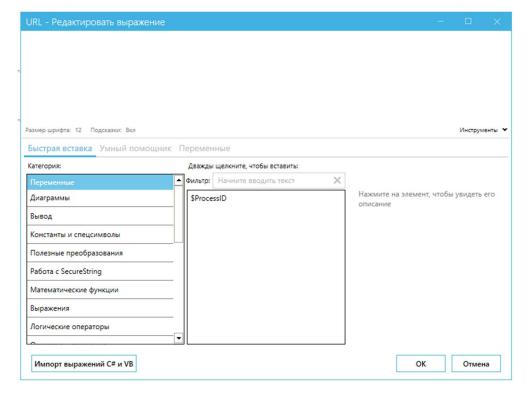
Элементы интерфейса окна «Редактировать выражение».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	кнопка 🖡	Позволяет свернуть или развернуть панель свойств и панель «Переменные» и сохранить это положение.
2.	кнопка	Позволяет свернуть панель свойств.
3.	Размер шрифта	Позволяет выбрать удобный для Вас размер шрифта в окне «Редактировать выражение».
4.	Подсказки	Позволяет включить или выключить подсказки в окне «Редактировать выражение». Данная функция указывает на ошибки в выражениях, если они есть.
5.	Инструменты	Позволяет вставить в строку значения свойства путь к файлу(ам), директории или вставить с экранированием символов.
6.	Импорт выражений С# и VB	Позволяет импортировать выражения С# и VB. При нажатии на кнопку открывается окно с текстовым полем для вставки выражения, а также, с возможностью выбора



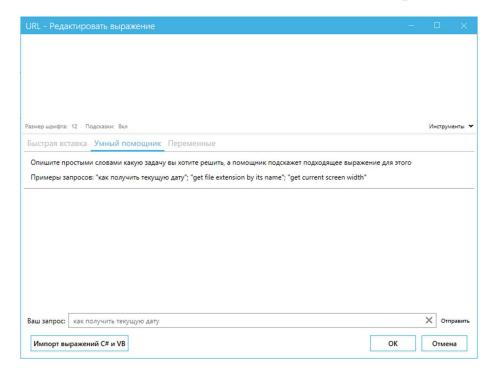
языка выражения: С#, Blue Prism, Visual
Basic.

Вкладка «Быстрая вставка» позволяет выбрать категорию выражения и само выражение из предложенного перечня. А также, при выборе выражения можно воспользоваться текстовым полем «Фильтр:».

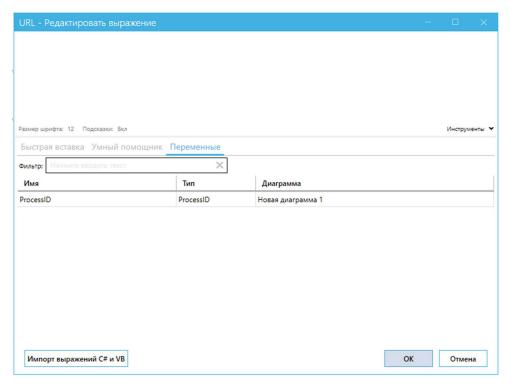


Вкладка «Умный помощник» позволяет описать простыми словами задачу, которую необходимо решить в текстовом поле «Ваш запрос:», после чего Умный помощник подскажет подходящее для этого выражение.





Вкладка «Переменные» позволяет указать необходимую переменную. Для поиска можно воспользоваться текстовым полем «Фильтр:» В списке переменных указаны: имя переменной, ее тип и название диаграммы, в которой используется переменная.



Не во все свойства блоков можно записать выражения. В те свойства, у которых нет кнопки с многоточием, можно записывать только константные значения, или выбирать их из предложенного списка.



3.1.5 Панель «Переменные»

Переменные – это ячейки памяти робота, места для хранения необходимой в процессе его работы информации.

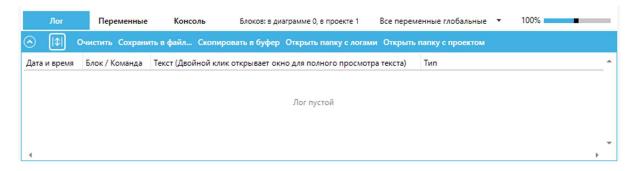
Переменная — это контейнер, содержащий какое-либо значение. При этом в зависимости от типа переменной в ней могут храниться числа, текст, таблицы и т.д. Панель «Переменные» в Дизайнере предназначена для создания и удаления переменных текущей диаграммы, а также изменения их типа и начального значения. В режиме отладки данная панель также показывает текущее значение каждой из переменных. Подробнее смотрите в разделе «3.3.2.3 Работа с переменными».

Имя	1	Тип	Начальное значение	
\$Переменная9		SecureString		
\$Переменная8		Money	0	
\$Переменная7		DataTable	Редактировать	
\$Переменнаяб		List	Редактировать	
\$Переменная5		DateTime	07.05.2024 19:02:53	
\$Переменная4		Hashtable	Редактировать	
\$Переменная3		Text	B.H.	
\$Переменная2		Logical	False	
\$Переменная10		Object		
\$Переменная1		Number	0	

3.1.6 Панель «Лог»

В нижней панели расположен лог важных действий программного робота — журнал работы робота. Автоматически в логе появляются записи об ошибках, а также важные предупреждения от Дизайнера. Вы можете использовать блок «Лог» для создания собственных записей в панели «Лог» во время работы сценария робота. Это удобно для отладки сценария, например, чтобы сообщить себе о том, что выполнение сценария достигло определенной точки, или о результате вычисления некоторого выражения.





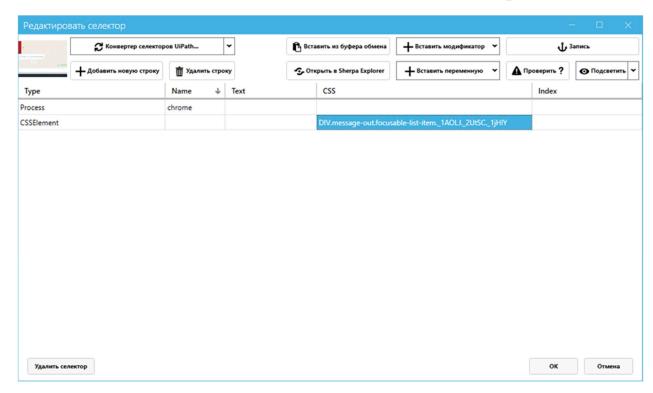
3.2 Руководство по веб-селекторам

В данном разделе мы не рассматриваем базовые способы записи и использования селекторов в Sherpa RPA с помощью кнопки и панели «Запись». Если Вы не знакомы с базовыми приемами записи селекторов – обратитесь к «Руководству пользователя Sherpa RPA» и обучающим видео на нашем официальном YouTube-канале. Также, в данном разделе, мы не рассматриваем десктоп-селекторы. Здесь мы рассказываем о способах подбора веб-селекторов в сложных случаях.

Для поиска элементов на веб-странице в блогах из палитры «Автоматизация браузеров» могут использоваться CSS- или XPath-селекторы. В Sherpa RPA инструмент записи действий пользователя (рекордер) по умолчанию записывает CSS-селекторы.

Если Вы хотите вручную использовать XPath-селектор вместо CSSселектора, допишите к селектору в колонке CSS в окне «Редактировать селектор» префикс «xpath:»





Если Вы задались вопросом, какой тип селекторов лучше – CSS или XPath то, на самом деле, в большинстве случаев нет никакой разницы. Почти любой CSS-селектор может быть преобразован в XPath, и наоборот, это просто два разных способа записи. CSS-селекторы для одних и тех же элементов, как правило, короче, чем XPath, и их удобнее читать, поэтому в Sherpa RPA по умолчанию используются именно они. XPath-нотация обладает дополнительным функционалом для работы с текстовым содержимым элементов, однако в Sherpa RPA для этой цели существует отдельная колонка в окне селекторов – Text, которая даёт те же самые возможности CSS селекторам, и более удобным способом.

Перечень возможных префиксов перечислен в выпадающем меню кнопки Вставить модификатор в окне «Редактировать селектор». В частности, доступен префикс «regex:», который позволяет в текстовых колонках селектора (например, в колонке Text) использовать регулярные выражения. Также в текстовых колонках селектора доступны подстановочные символы * (замена любой последовательности символов) и ? (замена ровно одного символа).

3.2.1 Ручной поиск CSS-селекторов в браузере

В случае, если инструмент записи действий пользователя (кнопка «Запись») записал не оптимальный селектор, вы можете подобрать альтернативный селектор вручную, ориентируясь на исходный код нужного элемента управления 87



в веб-странице, который можно увидеть с помощью «Инструментов разработчика» вашего браузера. Например, в Google Chrome в основном меню вы можете выбрать More Tools -> Developer Tools или нажать Ctrl+Shift+I. В появившейся панели «Developer Tools» Вы увидите текущий исходный код вебстраницы на вкладке «Elements». Вы также можете быстро открыть этот инструмент и перейти к коду нужного вам элемента управления, если на вебстранице кликните правой кнопкой мыши и выберите пункт контекстного меню «Inspect». Код, соответствующий нужному элементу, будет подсвечен на вкладке «Elements» в «Developer Tools».

Некоторые веб-страницы блокируют отображение стандартного контекстного меню браузера и вместо него по правому клику показывают собственное меню. Также, в некоторых случаях, клик по элементу меняет состояние элемента так, что у элемента пропадает нужный нам атрибут. Например, вы хотите записать селектор непрочитанного сообщения в мессенджере, но, когда Вы кликаете на него правой кнопкой мыши, сообщение становится прочитанным, и код этого элемента в Inspect меняется. Чтобы увидеть код нужного элемента в этих двух ситуациях – используйте кнопку «Select Element» – в Google Chrome это самая первая кнопка на панели «Elements».

```
Elements Console Recorder A Performance insights A Soumar range en or recorder and 
 Elements Console Recorder
                                                                                                                                                                                                                        Sources Network Performance Memory Application
    <head> ... </head>
    ▼ <body style="overflow: visible;">
         ▼ <div id="__docusaurus"> flex
                 <div id="docusaurus-base-url-issue-banner-container"></div>
              ▶ <div> · · · (div>
              ► <nav class="navbar navbar--fixed-top"> ··· </nav> flex
               ▼ <div class="main-wrapper">
                     ▼ <div class="home-wrapper">
                           ▼<header class="hero heroBanner_1ZBZ"> flex
                                ▼ <div class="container">
                                      ▼ <div class="hero-inner">
                                              <img src="/img/logo/logo-sm.png" alt="SourceAI Logo" class="heroLogo_17HU">
                                          <div class="heroText_3rZi"> ... </div>
                                         ▼ <div class="buttons irzW"> flex
                                                 <a href="https://www.youtube.com/watch?v=N1iFmqI608A" target="_blank" rel="noopener noreferrer" class="button"</pre>
                                                   </a> == $0
html body div#_docusaurus div.main-wrapper div.home-wrapper header.hero.heroBanner_1ZBZ div.container div.hero-inner div.buttons_irzW
```

В других браузерах (Edge, Firefox и т.д.) инструмент «Developer Tools» выглядит и работает похожим образом, но конкретные названия пунктов меню и комбинации клавиш для его вызова могут отличаться.

Важно понимать, что современные веб-страницы являются динамическими структурами, и они часто меняют сами себя (названия и само наличие и

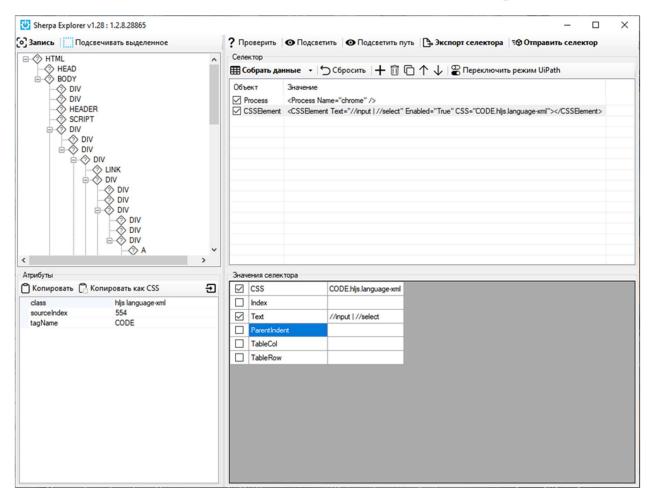


отсутствие тех или иных элементов управления, их порядок и иерархию, их внешний вид и атрибуты), уже после того как были загружены с сервера в кэш вашего веб-браузера. Кроме того, страницы часто загружаются по частям, с помощью фреймов или динамического изменения DOM-дерева элементов (DOM расшифровывается как «Document Object Model» – то есть модель веб-страницы, представленная в виде иерархического дерева тегов, именно такое дерево вы видите в Developer Tools браузера), в том числе, с помощью таких технологий как Ајах. Чтобы увидеть исходный HTML-код страницы в том виде, как он был первоначально получен браузером с сервера, нужно выбрать пункт «View page source» в контекстном меню страницы браузера. Однако этот код для целей автоматизации почти бесполезен, так как с момента загрузки страницы браузер, следуя заложенным в страницу скриптам, уже успел много раз её изменить. Developer Tools браузера показывает вам динамический, обновляемый в реальном времени слепок того, как DOM-дерево страницы выплядит прямо сейчас, поэтому для ручного подбора селекторов следует пользоваться именно им.

3.2.2 Ручной поиск CSS-селекторов с помощью Sherpa Explorer

Ещё один мощный инструмент ручного подбора и тестирования селекторов «Sherpa Explorer», он доступен в самом Sherpa Designer и вызывается по соответствующей кнопке «Sherpa Explorer». В нём, как и в обычном редакторе селекторов, Вы можете записать любой элемент интерактивно, а также исследовать иерархическое дерево любого Desktop и веб-приложения, его доступные атрибуты, подобрать и протестировать нужное сочетание атрибутов. Подобранный и проверенный селектор можно отправить в редактор селекторов самого Дизайнера с помощью нажатия на соответствующую кнопку.





3.2.3 Как понять, что селектор не оптимальный?

Для одного и того же элемента веб-интерфейса можно подобрать много разных селекторов, все из которых будут в той или иной степени рабочими. Например, в веб-клиенте WhatsApp селектор

#main > footer > div._21SWV._3cjY2.copyable-area > div >
span:nth-child(2) > div > div._1VZX7 > div._2xy_p._3XKXx >
button

и селектор

button[aria-label="Отправить"]

будут обозначать один и тот же элемент управления – кнопку «Отправить». И они оба будут работать в моменте. Первый селектор опирается на длинный путь от самого начала (корневого элемента) веб-страницы до конкретного элемента управления «вниз» по дереву иерархии элементов управления. Второй селектор опирается на тип тега элемента управления и один характерный атрибут этого тега. Несмотря на то, что в момент записи оба селектора работают корректно, первый селектор с гораздо большей вероятностью «сломается», то есть



перестанет работать, при очередном обновлении веб-приложения или веб-сайта. Например, по пути селектора появятся или наоборот исчезнут промежуточные элементы управления, и, следовательно, изменится сам путь от корневого элемента до нужного нам.

Также, если в селекторе присутствуют псевдослучайные комбинации букв и цифр (такие как 1VZX7 или 3XKXх в примере выше), это является признаком потенциально нестабильного селектора. Например, эта комбинация букв и цифр может меняться при каждом заходе пользователя на сайт, и для другого пользователя она будет уже другой. Проверить это можно, разлогинившись из сайта, закрыв браузер, открыв его и снова перейдя на нужную страницу, либо просто зайдя на эту страницу с другого компьютера или браузера, и проверив – те же буквы и цифры вы видите в этом месте селектора, или уже другие. Также, может помочь (но не всегда) комбинация Ctrl+F5, которая вызывает полную перезагрузку страницы. Но, даже если после всех этих действий, комбинация букв и цифр не поменялась, она может поменяться после следующего билда (обновления) веб-сайта или веб-сервиса, т.к. современные фреймворки веб-сайтов (такие как Angular, React и т.д.) могут генерировать эти названия классов случайным образом при каждой пересборке страницы.

Таким образом, чтобы робот в продуктиве работал надежно, и разработчику не приходилось слишком часто исправлять селекторы в блоках автоматизации браузеров, очень важно на этапе разработки подобрать как можно более стабильный селектор. Не всегда, но очень часто, самый стабильный селектор – это самый короткий из всех возможных для данного элемента.

Если инструмент рекордера не выдаёт вам достаточно стабильный селектор, его нужно подобрать или отредактировать вручную. Для этого нужно знать нотацию CSS-селекторов.

3.2.4 Нотация CSS-селекторов на примерах

Давайте рассмотрим длинный селектор из WhatsApp, который мы записали выше:

```
#main > footer > div._21SWV._3cjY2.copyable-area > div >
span:nth-child(2) > div > div._1VZX7 > div._2xy_p._3XKXx >
button
```



И сразу же посмотрим на участок кода в самой веб-странице, которая соответствует этому элементу.

```
🖟 🗖 Elements Console Recorder 🗸 Performance insights 🗸 Sources Network Performance Memory Application Security Lighthouse EditThioCookie
          (html class="is serviceworker adownload cssanimations csstransitions webp exiforientation webp-alpha webp-animation webp-lossless wf-loading" dir="LTR" loc="en" lang="ru">
          <span></span>
                                       <div tebindexm*-1" class="two _ij770"> ⟨sie
<div id="wa-popover-sbuckt">⟨/div⟩
<div class="snyj76hw an6tjent jbm6vef4 bb19m3t3 ora14ekb nv3qcefw">⟨/div⟩
<div class="snyj76hw an6tjent jbm6vef4 bb19m3t3 ora14ekb nv3qcefw">⟨/div⟩
<div class="_20g5f"> (div> fie)
</div class="_20g5f"> (div> fie)
</div class="_2756i _2x4Qv"></div> fie)
</div class="_2756i _2x4Qv"></div class="_2756i _2
                                                         \(\forall \) 
                                                                                                                                              daly teashnock* -1* clash* _0001/ (mix)

(for clash="2001/ (mix)

(for
e"> @ </svg>

4 xos-win div.two_1j/70 div_27s6i_2xAQV div#main_2Ex_b footer_3E8Fg div_2ISWV_3cjY2.copyable-area div_4/9rJ span div_2lyq div_1VZX7 div_2xy_p_3XXXX button.tvf2evcx.oq44ahr5.lb5m6g5c.sv/sagor.p2rjqpw5.epia9gcq
```

Для вашего удобства, мы подчеркнули красным все участки кода и идентификаторы, которые входят в этот селектор, чтобы вы могли сопоставить их в селекторе и в HTML-коде страницы. Начнем прослеживать селектор с самого начала и спустимся вниз по иерархии вложенных тегов веб-страницы до нужного нам элемента управления. Запись #main означает, что нам нужно найти элемент, у которого атрибут id имеет значение main. В отличие от имен классов, которые могут повторяться много раз, id элементов встречаются на странице только по одному разу. В данном случае эта запись нашла нам тег <div id="main">, при этом сам тег div или его класс никакой роли не играли. Символ > означает, что мы должны спуститься вниз по иерархии, к дочерним элементам найденного тега, и последующую часть селектора искать среди них.

Далее следует название тега footer, здесь мы нашли соответствующий тег просто по его названию, без всяких дополнительных уточнений. Далее видим ещё один символ >, значит спускаемся ещё ниже по иерархии элементов. Следующая запись div. 21SWV. 3cjY2.copyable-area означает, что мы должны найти тег div, среди классов которого должны быть одновременно 21SWV, . Зсју2 и copyable-area. Классов у самого искомого элемента может быть и больше, мы указываем только тот необходимый минимум, который нам нужен. Таким образом мы находим тег <div class=" 21SWV 3cjY2 copyable-area">,



который подходит под этот селектор. Обратите внимание, что в селекторе имена классов соединены через точку (которая в данном случае означает логический оператор И), а в тексте атрибута class в самом теге эти же имена классов перечислены через пробел – это важное отличие записи имен классов в атрибутах тегов и в селекторах.

Спускаемся ещё ниже, находим следующий элемент просто по названию тега div. Т.к. тегов div на этом уровне иерархии много, будет взят первый попавшийся. Спускаемся ещё ниже. Запись span:nth-child(2) говорит нам о том, что мы должны найти тег span, и далее перейти во второй по счёту дочерний элемент этого тега. Затем ещё раз находим элемент div просто по имени элемента. Затем находим тег <div class="_1vzx7"> который соответствует селектору div._1vzx7 - по названию тега и имени класса в этом теге. Затем находим элемент <div class="_2xy_p _3XKXx"> по селектору div._2xy_p._3XKXx, ну а в конце находим дочерний тег button просто по названию самого тега. Это была достаточно длинная цепочка, и по этому пути многое может пойти не так, если сайт будет обновляться. Зато на этом примере мы посмотрели, какие разные варианты нотации используются в CSS-селекторах.

Сравните с коротким селектором button [aria-label="Отправить"], который выделяет тот же элемент управления, и который мы предложили, как альтернативу. Почему мы вручную составили именно такой селектор? Сначала мы посмотрели на финальный тег, который нам нужен:

<button data-tab="11" aria-label="Отправить" class="tvf2evcx
oq44ahr5 lb5m6g5c svlsagor p2rjqpw5 epia9gcq">

Имена классов нам сразу не нравятся – их много, и они выглядят как произвольные сочетания букв и цифр, а это означает что в новой версии вебстраницы они почти наверняка изменятся. Название тега button выглядит перспективно. Не так уж много кнопок на этой странице. Но тем не менее само по себе название button может быть неуникальным – на странице могут быть и другие кнопки. Конечно мы можем использовать модификатор вроде button: nth-child(1), который укажет нам на точный порядковый номер однотипного элемента на странице. Или то же самое можно сделать, занеся число



1 в колонку Index справа от колонки CSS в окошке редактора селекторов Sherpa RPA – это даст тот же эффект.

Но кто даст гарантию, что при следующем обновлении страницы порядок или количество кнопок на странице не изменятся? Текст «Отправить» в атрибуте aria-label выглядит достаточно уникальным – это тот текст, которым будет замещен элемент управления в «экранных читалках» и других accessibility-инструментах для лиц с ограниченными возможностями. Очень маловероятно что разработчики сайта будут менять это название в следующих версиях вебприложения, поэтому его-то мы и возьмем за характерный отличительный атрибут. Чтобы указать, что нам нужен только такой тег button, у которого атрибут aria-label имеет значение «Отправить», используем квадратные скобки – button [aria-label="Отправить"]. Мы получили достаточно надежный селектор, который не стыдно использовать в продуктиве нашего робота.

3.2.5 Справочник по нотации CSS- и XPath-селекторов

Ниже мы перечислим популярные типы CSS-селекторов и соответствующие им селекторы XPath. Обратите внимание, что некоторые устаревшие версии браузеров могут не поддерживать отдельные типы селекторов, описанные ниже. Также, не забывайте, что все перечисленные ниже элементы селекторы можно легко комбинировать друг с другом в рамках одного большого селектора.

1) Значение ID элемента.

Идентификатор элемента в CSS определяется с помощью #, а в XPath с помощью [@id='example']. Идентификаторы должны быть уникальными в пределах DOM-дерева страницы.

Примеры:

CSS: #example

XPath: //div[@id='example']

2) Название тега.

В предыдущем примере мы использовали запись **//div** в варианте для XPath. Это название тега, например, **input** для текстового поля или кнопки, **img** для изображения или **a** для ссылки. Можно выделять элемент просто по названию его тега.

Примеры:



CSS: input

XPath: //input

3) Название класса.

В НТМL-коде названия классов пишутся внутри значения атрибута **class**, и, если классов у атрибута несколько, они разделяются (или, точнее, объединяются) пробелом. Но в селекторах имена классов указываются немного по-другому – в CSS названия классов перечисляются через точку, либо отделяются точкой от других частей того же элемента селектора, например, от названия тега. В XPath классы указываются в квадратных скобках в атрибуте @**class**.

Примеры:

```
CSS: div.example
XPath: //div[@class='example']
```

4) Значение атрибута

У HTML-тегов может быть множество атрибутов, и можно находить теги просто по названиям и значениям этих атрибутов. В примерах ниже мы найдем тег **input** по атрибуту **name** и его значению **«username»**:

```
CSS: input [name='username']
XPath: //input [@name='username']
```

Также есть возможность искать элементы по значениям сразу нескольких атрибутов.

Примеры:

```
CSS: input[name='login'] [type='submit']
XPath: //input[@name='login'and @type='submit']
```

5) Частичное значение атрибута.

Если значение атрибута, с помощью которого вы хотите найти элемент, частично изменяется в процессе работы сайта, вы можете использовать для поиска только ту часть значения, которая остаётся неизменной. Такая возможность есть только в CSS-селекторах:

Поиск элемента по префиксу значения атрибута:

```
CSS: a[id^='id prefix ']
```

В данном случае мы найдем гиперссылку, у которой атрибут **ID** начинается на **«id_prefix_»**. Поиск элемента по суффиксу значения атрибута:



```
CSS: a[id$=' id sufix']
```

В данном случае мы найдем гиперссылку, у которой атрибут **ID** заканчивается на **«_id_sufix»**. Поиск элемента по подстроке в значении атрибута:

В данном случае мы найдем гиперссылку, у которой атрибут ID содержит подстроку «id_pattern».

6) Объединение результатов нескольких селекторов.

В примере с WhatsApp выше все исходящие сообщения можно найти с помощью селектора div. message-out, а все входящие сообщения с помощью селектора div. message-in. А что делать, если мы в одном списке хотим получить или перебрать и входящие, и исходящие сообщения, причем в том порядке, в котором они идут в чате? В CSS для этих случаев можно легко объединить списки элементов, полученные разными селекторами с помощью запятой. В XPath для этих же целей используется символ вертикальной черты.

Пример:

```
CSS: div.message-out, div.message-in
XPath: //div[@class=' message '] | //div[@class= 'message-
in']
```

7) Непосредственный дочерний элемент.

Страницы HTML имеют структуру, подобную XML, с дочерними элементами, вложенными в родительские. Прямой дочерний элемент определяется в CSS с помощью символа >, а в XPath с помощью символа /.

Примеры:

```
CSS: div > a
XPath: //div/a
```

8) Дочерний элемент ниже по иерархии.

Описание цепочки всех непосредственных дочерних элементов, во-первых, утомительно, а во-вторых, приводит к ненадежным селекторам. Если элемент может находиться внутри другого или внутри одного из его дочерних элементов, он определяется в CSS просто с помощью пробела, а в XPath с помощью //.

Примеры:

CSS: div a



XPath: //div//a

9) Следующий потомок.

Этот тип селектора полезен, чтобы перебирать однотипные элементы на одном уровне иерархии, например, строки таблиц, элементы нумерованных или ненумерованных списков. Следующий потомок – это следующий элемент однородного списка элементов на том же уровне иерархии что и текущий.

Представьте себе, что у вас в форме идут подряд два тега **input**, вот таких:

```
<input type = "text" class = "form-control" id = "username"
name = "username" placeholder = "username" required
autofocus></br>
```

По какой-то причине вы не можете адресоваться ко второму **input** напрямую, но у вас есть устойчивый селектор для первого **input**. В такой ситуации, для выделения второго **input** в CSS вам нужно будет записать вот так:

```
#username + input
```

А в XPath вот так:

```
//input[@id='username']/following-sibling:input[1]
```

10) Псевдоклассы для однородных элементов.

Существует несколько псевдоклассов, которые позволяют выделить нужный по счёту элемент просто по его порядковому номеру на нужном уровне иерархии (nth-child), либо же по порядковому номеру и типу элемента (nth-of-type).

Селектор ниже выделит четвертый по счёту тег **LI** на заданном уровне иерархии:

```
CSS: li:nth-of-type (4)
```

Если же мы хотим выделить четвертый по счёту тег, являющийся потомком тега **LI**, безотносительно типа этого тега, можно сделать вот так:

```
CSS: li:nth-child(4)
```

В XPath есть возможность указывать нужный по счёту дочерний элемент относительно текущего элемента просто приписав порядковый номер в квадратных скобках:

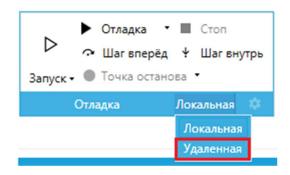
```
//li[4]
```



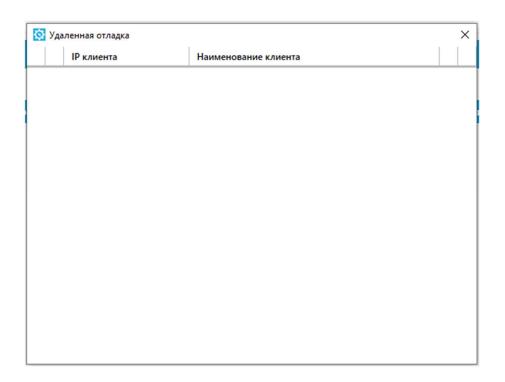
3.3 Основные функции Sherpa RPA 3.3.1 Отладка

B Sherpa Designer доступны два вида отладки: локальная и удаленная.

Для работы в режиме удаленной отладки в меню «Отладка» нажмите на кнопку «Локальная» (локальный режим отладки стоит по умолчанию) и, в выпадающем списке, выберите режим «Удаленная».



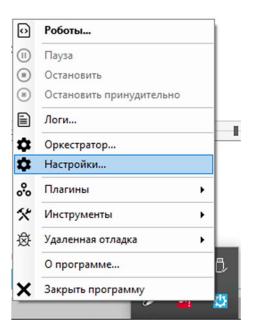
Затем нажмите на кнопку , после чего откроется окно «Удаленная отладка». После настройки удаленной отладки робота в этом окне появятся IP и наименование клиента, доступные для удаленной отладки.



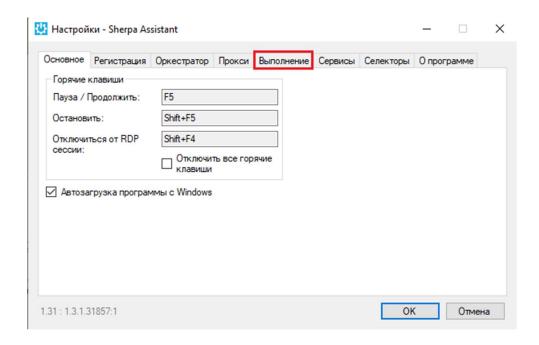
После включения режима удаленной отладки в Sherpa Designer, необходимо настроить удаленную отладку в Sherpa Assistant. Запустите



программу, затем в трее правой кнопкой мыши нажмите на и выберите пункт контекстного меню «Настройки...».

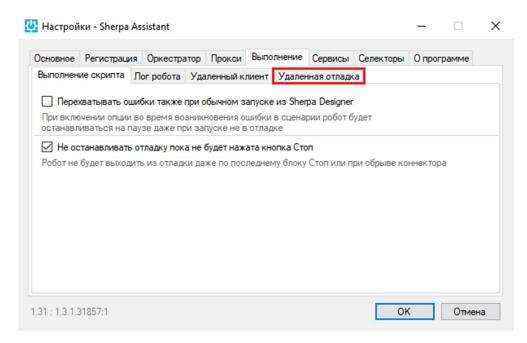


В открывшемся окне, выберите вкладку «Выполнение».



В меню вкладки «Выполнение» выберите вкладку «Удаленная отладка».



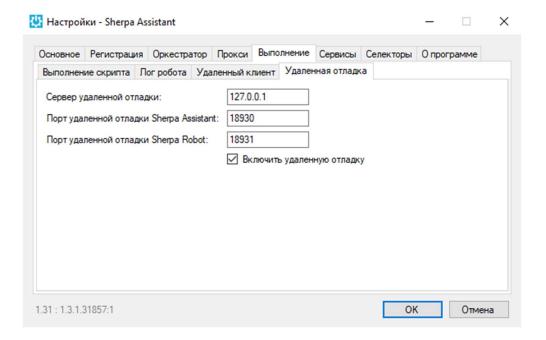


В меню вкладки «Удаленная отладка» укажите:

- сервер удаленной отладки;
- порт удаленной отладки Sherpa Assistant;
- порт удаленной отладки Sherpa Robot.

Поставьте флаг в поле «Включить удаленную отладку».

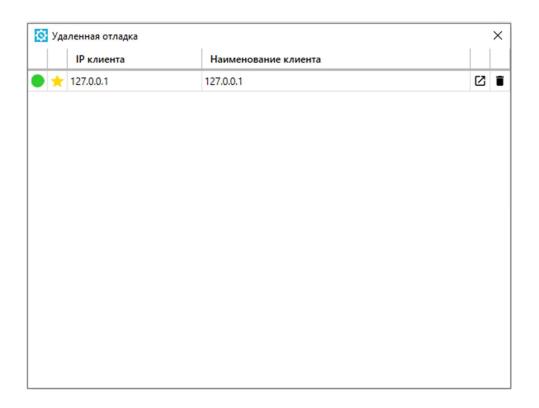
Нажмите кнопку «ОК». Удаленная отладка робота на платформе Sherpa RPA включена.





Проверьте в программе Sherpa Designer подключение удаленного клиента.

Для этого снова нажмите на кнопку ²². В открывшемся окне появилась строка с IP и наименованием клиента.



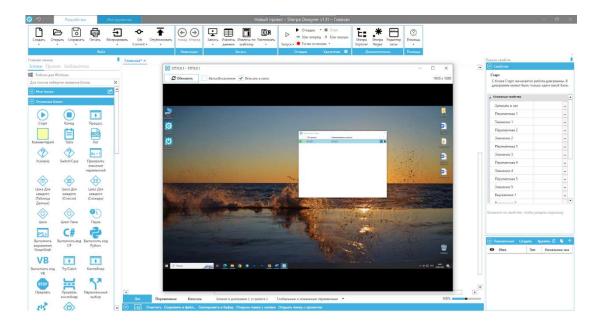
Элементы интерфейса окна «Удаленная отладка».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	индикатор	Клиент подключен.
2.	кнопка	Клиент по умолчанию. Если доступны для удаленной отладки несколько клиентов, то с помощью этой кнопки можно переключать клиента, на котором будет производиться удаленная отладка.
3.	столбец «IP клиента»	IP клиента, доступного для удаленной отладки.



4.	столбец	Наименование клиента, доступного для
	«Наименование	удаленной отладки.
	клиента»	
5.	кнопка 🗹	Просмотр рабочего стола подключенного
		клиента.
6.	кнопка	Удаление клиента из списка доступных для
		удаленной отладки.

Чтобы проверить работу режима удаленной отладки, нажмите на кнопку просмотра рабочего стола подключенного клиента.



В окне просмотра рабочего стола клиента можно включить автообновление, поставив флаг в поле «Автообновление». Или использовать ручное обновление с помощью кнопки «Обновить». Флаг в поле «Вписать в окно» стоит по умолчанию, его также можно убрать.

3.3.2 Запись действий пользователя

Инструмент Записи (рекордер) является важной частью Sherpa Designer, которая может помочь сэкономить большое количество времени при автоматизации бизнес-процессов. Эта функция позволяет легко записывать



действия пользователя на экране, такие как запуск программ, клики на кнопках и гиперссылках веб-страниц, выбор элементов меню, вызов контекстного меню и т.д., и преобразовывать их в блоки бизнес-процесса.

Рекордер используется для интерактивного захвата элементов управления в приложениях и браузерах, таким образом происходит автоматизированное создание сценария. Можно записать всю последовательность действий пользователя (работает только для левого, правого и двойного клика) или использовать пошаговую запись (доступен более широкий список записываемых действий). Выбор элементов управления в браузере происходит без использования кода веб-страниц, интерактивно.

Кнопка «Запись» В основном меню Дизайнера или «Запись» из окна «Редактировать селектор» вызывают меню инструмента записи.

Меню записи появляется у левой границы экрана, при этом окно Sherpa Designer сворачивается.





Элементы интерфейса Меню записи.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	кнопка	Позволяет отменить запись действий пользователя и выйти без сохранения действий.
2.	кнопка	Позволяет удалить последнее действие пользователя при записи.
3.	кнопка	Позволяет сохранить запись действий пользователя и закрыть меню записи.
4.	кнопка	Позволяет записать последовательность кликов, совершаемых пользователем.
5.	кнопка	Позволяет запустить программу с помощью дополнительного диалогового окна.
6.	кнопка	Позволяет эмулировать нажатие кнопок мыши.
6.1.	кнопка	Позволяет эмулировать нажатие левой кнопки мыши.
6.2.	кнопка	Позволяет эмулировать нажатие правой кнопки мыши.
6.3.	кнопка	Позволяет эмулировать двойное нажатие левой кнопки мыши.
6.4.	кнопка	Позволяет эмулировать задержку курсора мыши над элементом управления.



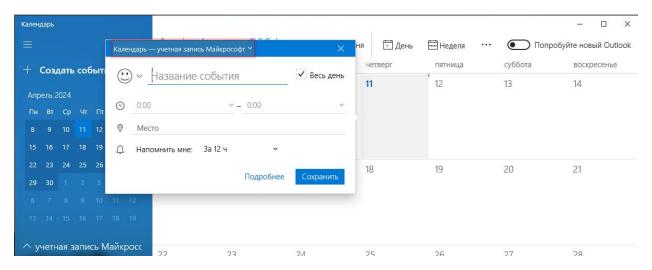
6.5.	кнопка	Позволяет эмулировать перетаскивание элемента.
7.	кнопка	Позволяет записывать клики по изображению.
7.1.	кнопка	Позволяет записать клик по заданному изображению на экране. Эту же функцию позволяет выполнять пункт «Записать изображение» из выпадающего списка меню Записи.
7.2.	кнопка	Позволяет записать клик по заданному изображению внутри окна или элемента управления.
8.	кнопка	Позволяет записывать действия пользователя при работе с текстом.
8.1.	кнопка	Позволяет получить видимый текст элемента управления.
8.2.	кнопка	Позволяет записать видимый текст в элемент управления.
8.3.	кнопка	Позволяет кликнуть по месту расположения заданного текста на экране, предварительно распознав все текстовые надписи с экрана.
8.4.	кнопка	Позволяет получить текст с экрана методом распознавания образов.



8.5.	кнопка	Позволяет извлечь данные из документа в приложении (Word, Excel, PDF и т.д.) или из веб-страницы, открытой в браузере. Эту же функцию позволяет выполнять кнопка «Извлечь данные» на панели «Запись».
9.	кнопка	Позволяет записывать действия, связанные с копированием и вставкой.
9.1.	кнопка	Позволяет скопировать в буфер обмена текст из элемента управления.
9.2.	кнопка	Позволяет вставить из буфера обмена текст в заданный элемент управления
10.	кнопка	Позволяет записывать задержки в выполнении процесса.
10.1.	кнопка	Позволяет записать ожидание появления элемента на экране.
10.2.	кнопка	Позволяет записать ожидание исчезновения элемента с экрана.
11.	кнопка	Позволяет эмулировать нажатие клавиш клавиатуры с помощью дополнительного диалогового окна.

Во время записи элементы пользовательского интерфейса выделяются красной рамкой, как показано на приведенном ниже снимке экрана, что позволяет быть уверенным в правильности выбора соответствующих кнопок, полей и меню.



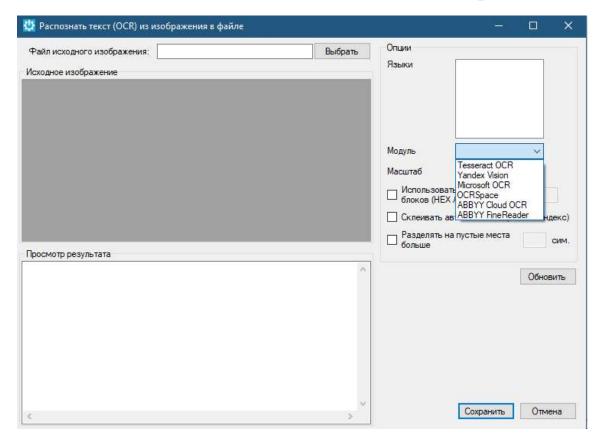




Блоки бизнес-процесса в Диаграмме, созданные с помощью Записи интерфейса пользователя, отличаются от Блоков, добавленных из палитры. Отличие заключается в наличие значка «фото», при наведении на который отображаются информативные снимки экрана, сделанные программой в момент выбора элемента управления. Маленьким крестиком показана точка, на которую пользователь кликнул при записи.

Функция Get OCR Text — получить текст с экрана методом распознавания образов.





Во время Записи выделите элементы (область экрана) на которой нужно произвести поиск текста – то есть выделите область, в которой будет проводиться распознавание. Область ограничьте красной рамкой, которая появляется зажатием кнопки мыши. Распознавание текста проводится с помощью встроенных и внешних модулей распознавания:

- Tesseract OCR;
- Yandex Vision;
- Microsoft OCR;
- OCR Space;
- ABBYY Cloud OCR;
- ABBYY FineReader.

Выберите нужный модуль распознавания. Вы можете сразу видеть результаты распознавания в окне слева. Если результаты распознавания верны, нажмите «Сохранить». Если результаты распознавания не удовлетворяют, выберите другой модуль распознавания из списка и нажмите «Обновить» или измените доступные настройки распознавания.

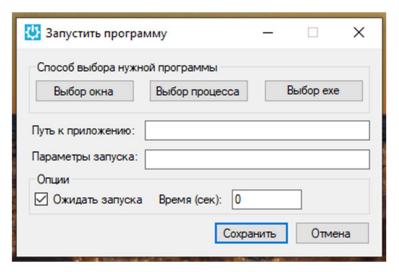
Доступны для подключения следующие OCR:



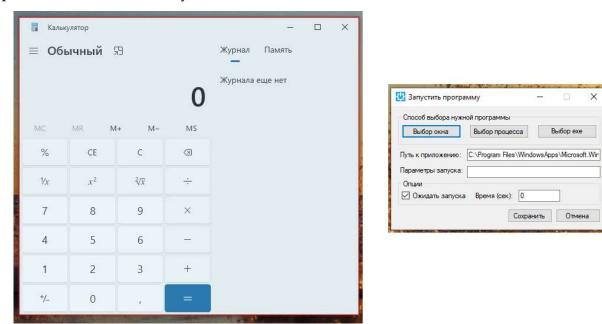
- Tesseract OCR;
- ABBYY OCR;
- Microsoft OCR;
- Yandex Vision.

3.3.2.1 Пример записи действий при работе с Калькулятором

Допустим, перед работой с Калькулятором нужно, чтобы робот сначала его запустил. Для этого нажмите кнопку (запустить программу) в меню записи и ознакомьтесь с вариантами.



При нажатии кнопки «Выбор окна» просто подсвечивается окно нужной программы. Нажмите кнопку Ctrl.



Робот сам нашёл путь к приложению. При желании можно дописать параметры запуска (параметры командной строки). Можно указать, что робот



должен ожидать запуска в течение какого-то времени. При нажатии на кнопку «Сохранить» блок запуска программы добавляется в диаграмму.

При нажатии кнопки «Выбор процесса» можно выбрать процесс из списка запущенных в данный момент процессов.

😃 Выбор проце	cca		×	(
ID процесса (Идентификатор окна)	Имя процесса 🔺	Файл	Полный путь	^
1408 (0)	RtkAudUService64	RtkAudUService64.exe	C:\Windows\System32\RtkAudUService64.exe	
9924 (0)	RuntimeBroker	RuntimeBroker.exe	C:\Windows\System32\RuntimeBroker.exe	
12836 (0)	RuntimeBroker	Runtime Broker.exe	C:\Windows\System32\RuntimeBroker.exe	
11156 (0)	RuntimeBroker	RuntimeBroker.exe	C:\Windows\System32\RuntimeBroker.exe	
11392 (0)	RuntimeBroker	RuntimeBroker.exe	C:\Windows\System32\RuntimeBroker.exe	
13824 (0)	RuntimeBroker	RuntimeBroker.exe	C:\Windows\System32\RuntimeBroker.exe	
13524 (0)	RuntimeBroker	Runtime Broker.exe	C:\Windows\System32\RuntimeBroker.exe	
6652 (0)	RuntimeBroker	RuntimeBroker.exe	C:\Windows\System32\RuntimeBroker.exe	
<			0.D D (00%)	. ~

Третий вариант - мы выбираем нужный ехе-файл из проводника.

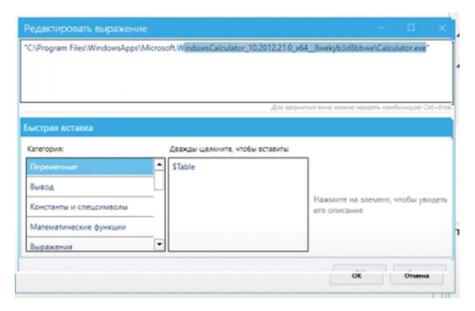
Выбор окна, как правило, наиболее удобный способ.

В диаграмме появился блок «Запустить приложение», настроенный на запуск соответствующего приложения.



Обратите внимание, что робот умеет разбирать сложные случаи. Например, калькулятор в Windows10 не является настоящим приложением, запускается через вспомогательный механизм и через командную строку его запустить не получится. Поэтому робот находит правильный путь для этого приложения.

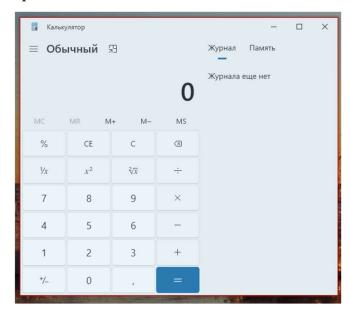




Далее в палитре записи действий:

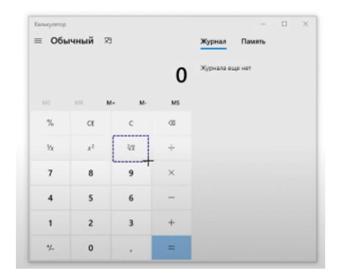
Можно задерживать курсор над элементом управления в течение какого-то времени, можно делать одинарные и двойные клики, клик по заданному изображению на экране или клик по изображению внутри окна или элемента управления. Здесь работает такая же логика, как при распознавании текста с экрана, только в данном случае просто ищется некий визуальный образ какого-то элемента на экране. Например, нужно найти кнопку калькулятора просто по внешнему виду, а не по её селектору или тексту на ней. Нажмите на инструмент «Клик по заданному изображению внутри окна или элемента управления». Далее нужно подсветить желаемый элемент управления и нажать Ctrl.

Мы просто подсвечиваем окно калькулятора, чтобы робот не искал нужное изображение на всём экране.

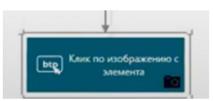




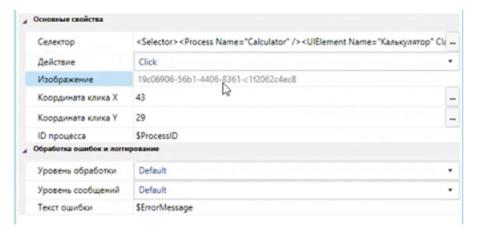
Нажмите Ctrl и затем выберите образец того изображения, которое вы будете искать.



Далее необходимо указать точку внутри или снаружи найденного изображения, в которой действительно будет производится клик. Т.е. найденную картинку можно использовать как якорь и кликать на другую точку в окне относительно этого якоря. Кликнем в центр. Действие записалось, его можно будет потом запустить.



Когда вы работаете с блоками, которые каким-либо образом записывают изображение, то свойства, в которых сохраняются эти изображения, выглядят как уникальный набор символов (идентификатор):



Под этим именем робот сохраняет картинки с образцами изображений в папке Images внутри вашего проекта (такая папка есть в каждом проекте).

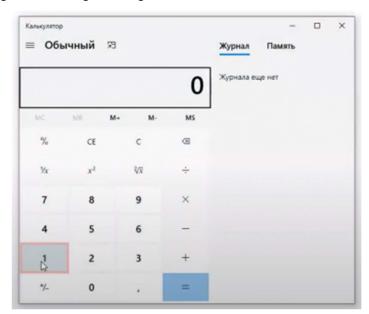


При нажатии на кнопку «Опубликовать», внутрь единого архива публикуется не только диаграмма, но и все картинки из папки Images. Это значит, что робота со всеми его ресурсами потом можно перенести с виде одного файла с расширением .robot на другой компьютер и там без проблем запустить.

Функция ожидания появления элемента на экране бывает очень полезна. Иногда может быть нужно задержать работу робота, пока мы не дождёмся, чтобы какое — то окно исчезло с экрана или какой-то элемент не появился на экране. Нажимаем на этот инструмент.



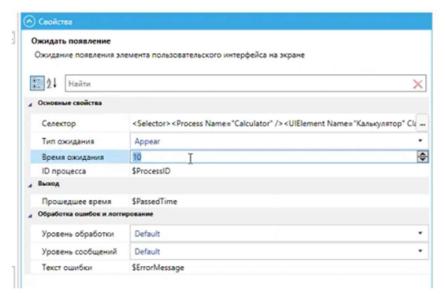
И как обычно, укажите на нужный элемент управления, относительно которого будет задерживаться работа робота.



Нажатием Ctrl записали действие. Посмотрите свойства добавленного в диаграмму блока.

Можно задать предельное время ожидания в секундах. Робот продолжит выполнять сценарий после этого блока, либо когда указанный элемент управления по селектору появится на экране, либо когда истечёт время ожидания.





Если время ожидания истекло, но элемент ещё не появился, есть возможность подключить справа какой-нибудь блок по коннектору «Если ошибка», и если он подключен в случае, если элемент на экране так и не появился, сценарий пойдет дальше не вниз, а вправо. Если коннектор «Если ошибка» не подключен, сценарии в любом случае пойдёт вниз.

Свойство «Прошедшее время» — переменная числового типа, в которую записывается количество секунд, которое на самом деле прошло до момента появления этого элемента. Например, указанное максимально время ожидания 10 минут, а элемент появился через 5 минут, в переменной PassedTime будет записано число 5.

Блок «Подсветить элемент» – полезно при создании интерактивных обучающих руководств. Есть блок, позволяющий захватить и перетаскивать элементы, блоки, позволяющие выполнять какие-то действия в браузерах, например, обновлять страницы, нажимать на кнопку Назад. Есть возможность не только запускать браузер с заданной вкладкой, но и подключаться к браузеру и к его вкладке, которая уже находится на экране.

Напомним, что при клике по элементу управления, который находится за пределами экрана, во всех случаях робот постарается автоматически проскроллить какую-то страницу или список в приложении до указанного элемента, если он находится не на виду.

С кликами мышью в автоматизации браузеров связан ещё один интересный момент: как правило, во всех блоках, которые работают с программами и



браузерами, есть свойство «ID процесса». В нём записана переменная ProcessID, это некая переменная-указатель на ту конкретную программу, с которой мы сейчас работаем. Это важно, т.к. одновременно могут быть открыты два приложения Word, и в каждом из них по два документа. Вы можете переносить что-то из документа в документ. Т.е. когда вы поставили блок для работы с Word или Excel, или какой-то другой программой, вам нужно указать, каким именно приложением, т.е. конкретным экземпляром приложения мы сейчас пользуемся.

Аналогично происходит с браузерами. Можно открыть Chrome или Explorer и в нём несколько вкладок. Возможно, нужно будет переключаться между вкладками. Поэтому переменная ProcessID содержит информацию о том, с каким конкретным экземпляром приложения мы сейчас работаем, а также о том (если это браузер), с какой вкладкой этого браузера мы сейчас работаем.

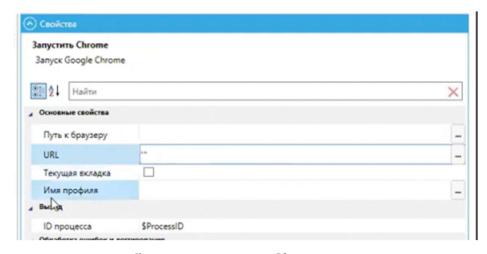
Откуда берется значение этой переменной? Она появляется в блоке «Запустить приложение». У блока «Клик мышью» ID процесса – входное свойство, определяющее работу блока. А у блока «Запустить приложение» ID процесса – выходное свойство, т.е. блок запишет в эту переменную нужное значение после запуска приложения.

3.3.2.2 Особенности работы с Google Chrome

Также есть отдельные блоки для запуска Chrome, где мы можем указать URL, а также указать, открываем мы страницу в текущей вкладке или в новой вкладке. Здесь же в выходных свойствах тоже есть ID процесса.

Если вы сначала, допустим, запустили Chrome, а потом сделали клик мышью, и у вас название переменной ProcessID совпадает с названием переменной ProcessID в следующем блоке, это значит, что эти два блока будут работать с одним и тем же браузером и с одной и той же вкладкой.



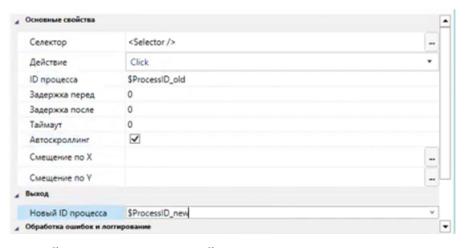


Если вы поставили два блока для запуска Chrome, и один запустил станицу на одной вкладке, а другой – на другой вкладке, можно развести по разным ID процесса. В одном случае переменная будет называться ProcessID1, в другом – ProcessID2.

При клике мышью по какому-то элементу внутри этой страницы (работает с любыми блоками, где есть селекторы), в поле ID процесса вы можете указать правильную переменную, которая будет определять и конкретный экземпляр браузера, и конкретную вкладку, с которой мы сейчас работаем.

Блок клика мышью для разных браузеров обладает некоторыми интересными возможностями. Одна из них заключается в том, что здесь ID процесса является и входным свойством, и выходным свойством. Точнее, выходное свойство называется «Новый ID процесса». Часто бывает, что, когда вы кликаете на какой-то ссылке внутри страницы, сама страница так устроена, что она открывает новую страницу не в той же вкладке, а в новой. И получается, что у вас есть старая вкладка со страницей, с которой вы только что ушли, и новая вкладка со страницей, на которую вы перешли. Бывает полезно иметь возможность работать как с новой, так и с предыдущей вкладкой. Тогда в переменную, указанную в выходном свойстве «Новый ID процесса» будет записан идентификатор той вкладки, на которую мы перешли. Это значит, например, что эти две переменные можно сделать разными, на входе допишем к имени переменной old, а на выходе пеw.





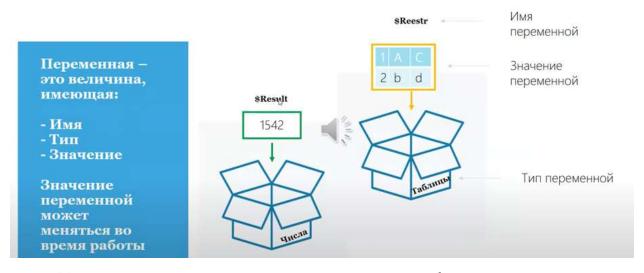
В следующих блоках вы можете работать с новой только что открытой вкладкой, закрыть её и продолжить работать со старой вкладкой, или даже переключаться между ними, работая в одном блоке с одной вкладкой, а в другом блоке с другой вкладкой. Если этот функционал не требуется, по умолчанию оставляем на входе и выходе имя переменной ProcessID. Это значит, что, если вы открыли при клике на какую-то ссылку новую вкладку, в ProcessID запишется идентификатор новой вкладки, и вы продолжите работать в этой новой вкладке.

3.3.3 Работа с переменными

Переменная — это ячейка памяти, своеобразный контейнер, содержащий какое-либо значение. При этом в зависимости от типа переменной значениями могут быть числа, текст, изображения, таблицы и даже более комплексные объекты, такие как списки, словари, коллекции т.д. Переменные в Дизайнере используются для хранения данных различных типов информации и её передаче между блоками.

Можно представить себе переменную как некую коробочку, у которой есть название, тип и значение. Название переменной всегда начинается со знака \$. На рисунке ниже первая переменная называется Result, её значение – 1542, тип данных – число. Если при создании переменной указать определенный тип данных, она будет работать именно с ним. Пременная Result, соответственно, может хранить только числа.



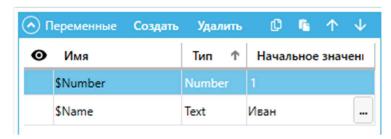


Имя переменной является ее уникальным идентификатором, и оно определяет способ ее отображения и использования.

В Sherpa RPA предусмотрена специальная панель Переменные, на которой пользователь может создавать, переименовывать и удалять переменные.

Объявление переменной

Чтобы использовать переменную, требуется сначала ее создать, или, если использовать терминологию программистов, объявить переменную. Чтобы сделать это, нужно ввести символ \$ за которым следует имя, которое вы хотите дать своей переменной. Имя переменной может быть любым, однако не должно содержать пробелов и специальных символов.



Однако желательно, чтобы оно было описательное, и было интуитивно понятно, какие данные содержит эта переменная. Не используйте только отдельные буквы, цифры или большие длинные фразы.

У переменных могут быть русскоязычные и англоязычные названия, но в названии не может быть пробелов. Можно выбирать тип данных для переменной (число, текст, выражение и пр.). Некоторые из типов — простые (число, логический, текстовый тип), другие являются составными (словарь, список, таблица данных). Также можно задавать начальные значения переменных (значение, которое переменная примет в момент запуска сценария).



Во многих Свойствах Блоков по умолчанию уже указаны названия Переменных (например, переменная \$ProcessID на картинке). Вы можете использовать их в своих проектах, изменять или вводить собственные переменные.

Типы переменных

Поддерживается большое количество типов данных, которые хранятся в переменной: число, логические, текст, выражение (универсальное значение), картинка, массив, словарь, дата и время и другие.

- Текст. Хранит текстовые строки. При задании константных значений текстовых переменных эти значения всегда заключаются в кавычки.
- Число. Числа могут быть обычными или с плавающей точкой.
 Пишутся без кавычек.
 - Таблицы. Могут хранить некоторое количество строк и столбцов.
- Списки. Могут хранить некоторое количество строк. Можно получить значение той или иной строки по её индексу.
 - Дата и время. Хранит только дату или дату и время.
- Словари (хэш-таблицы). Список, котором каждая строка представляет собой некое имя и присвоенное ему значение (название ключа и значение ключа).
- Логический тип. Принимает два значения (хранит одну из двух констант) true и false (истина/ложь).

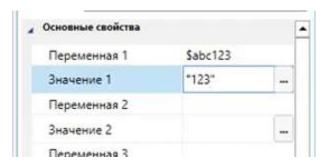




При создании переменной на панели «Переменные» её начальное значение может быть задано в специальном окне.



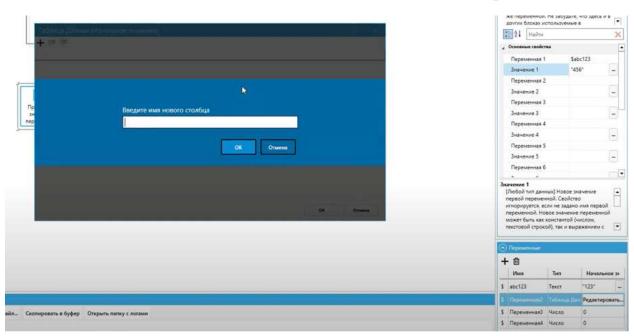
При работе с текстовыми типами данных или с текстовыми значениями константы всегда заключаются в кавычки. Если вы выберете переменную в свойствах и захотите присвоить ей текстовое значение, оно должно быть заключено в кавычки:



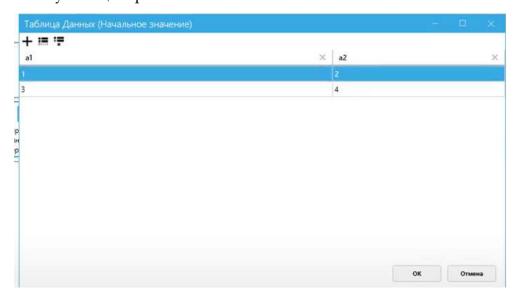
Если переменной присваивается числовое значение, оно указывается без кавычек.

При выборе составного типа данных для переменной есть возможность сразу задать структуру этого типа данных. Например, в таблице можно задать количество и заголовки строк и столбцов и присвоить им значения.





Данная таблица содержится в нашей переменной с именем «Переменная2» на момент запуска сценария.



В режиме отладки при проходе по шагам сценария и остановке на отдельном блоке на панели Переменные отобразится текущее значение переменных. Таким образом, если ранее переменная не была объявлена на панели «Переменные», при отладке невозможно будет посмотреть текущее значение переменной.

Если вы хотите использовать переменную, установленную по умолчанию, то добавьте ее в панель «Переменные», то есть явным образом укажите использование данной переменной. Назначьте тип и начальное значение каждой



переменной. Этот шаг не является обязательным, то есть переменная будет работать даже без добавления переменной в панель «Переменные».

Существует два вида переменных:

- Системные переменные, которые по умолчанию уже существуют при создании диаграммы;
- Пользовательские переменные, которые можно создавать самостоятельно.

При создании собственных переменных необходимо указать тип данных новой переменной. Sherpa Designer использует неявную типизацию. Это значит, что, если вы попытаетесь присвоить переменной выражение или константу с другим типом данных, это в большинстве случаев не приведет к ошибке, и переменная изменит свой тип данных автоматически. Однако это может привести к неправильному отображению текущего значения переменной на панели «Переменные» при отладке, а также к неявным ошибкам в работе алгоритмов. Поэтому рекомендуем всегда явным образом объявлять и устанавливать правильный тип данных используемым переменным.

С простыми типами данных платформа Sherpa проводит т.н. неявные преобразования. Например, если вы сложите две простые переменные, получится число. Если к числовой переменной прибавить текстовую, платформа автоматически распознает, что складываются переменные с разными типами данных и преобразует число в текст. Если вы решили присвоить текстовую константу числовой переменной, это, как правило, автоматически приведет не к ошибке, а к изменению типа данных.

В результате неявных преобразований не только значения, но даже типы переменных могут меняться во время работы сценария.

Использование переменной





Если вам нужно передать данные между разными блоками, воспользуйтесь переменными, например, установите пароль в одном блоке и проверьте правильность его ввода в другом блоке. Для этого сначала создайте переменную в панели «Переменные». После создания данную переменную можно будет использовать в свойствах любого Блока в данном проекте, в том числе в других диаграммах.

Как только переменная объявлена (то есть на панели «Переменные» заданы имя, тип и начальное значение переменной), ей можно присвоить значение с помощью блока «Assign Variable». С помощью этого же блока можно присвоить одной переменной значение из другой переменной или вычислить результат арифметического выражения, использовать математические, логические, строковые и другие функции и методы.

Если вы объявите переменную и присвоите ей значение, заключенное в кавычки, то Sherpa Designer будет обрабатывать переменную как строку: \$Текст = "Привет"

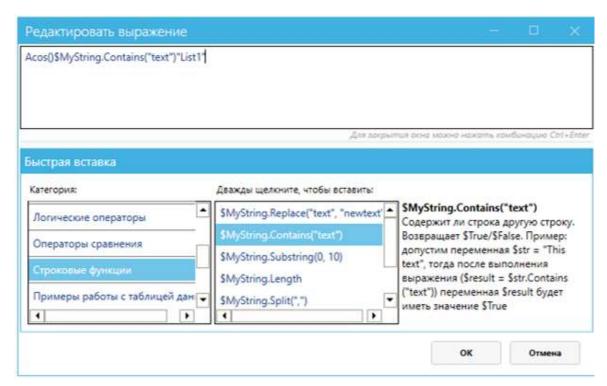
3.3.4 Работа с выражениями

Выражения – важный и мощный инструмент, позволяющий не использовать дополнительные лишние блоки, а прямо в том месте, где возникла такая потребность, производить разные арифметические вычисления, выполнять операции со строками и использовать встроенные функции (и всё это внутри свойства).

Редактировать выражения

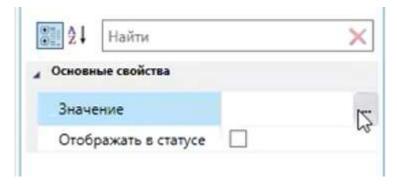
Для удобства работы с выражениями используйте окно «Редактировать выражения». Окно вызывается из панели Свойства блока нажатием на многоточие рядом с любым свойством, которое поддерживает использование выражений.





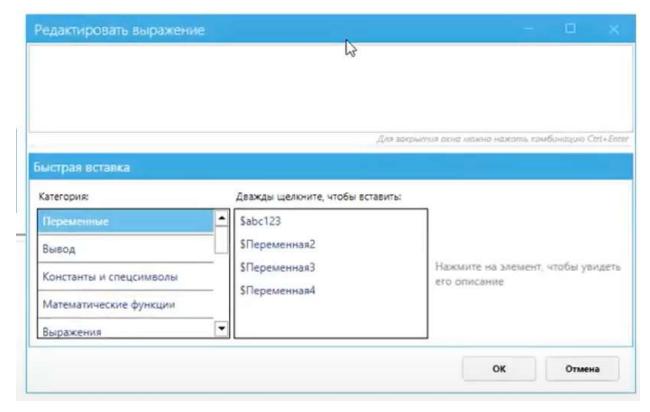
Окно «Редактировать выражение» предназначено для формирования выражений, которые могут быть использованы в качестве условий для проверки, формул для вычисления значений и т.д. Редактор позволяет создавать и редактировать ранее созданные выражения без привлечения разработчиков системы, силами подготовленного пользователя или администратора. В нижней части данного окна есть справка по функциям. Выберите категорию функций и саму функцию, чтобы увидеть подсказку по ее использованию в выражении. Дважды кликните на функции или примере использования выражения чтобы вставить его в текущее редактируемое выражение.

Выражения можно использовать в любом блоке и в любом свойстве, у которого справа есть кнопка с тремя точками.



При клике на такую кнопку открывается окно редактирования выражения. Здесь можно задать константу, соответствующую тому типу данных, который ожидает свойство.





Структура выражений

Выражения в системе используются для наложения различных условий на данные при поиске данных, отборе данных для отчетов, проверке правильности ввода данных и т.д. Выражения могут быть записаны в виде математических функций, даты и произвольных выражений (написанных на языке выражений, который похож на язык PowerShell). В выражениях могут быть использованы константы, переменные, математические функции или методы.

Когда вы используете выражение в свойстве Блока, то в момент, когда выполнение сценария дошло до этого Блока, вычисляются значения выражений, при этом берутся текущие значений всех используемых в выражении переменных. Результат вычисления выражения используется как значение свойства, в котором это выражение было записано.

Свойства, которые размещены в разделе Output, не допускают использования выражений. В них необходимо записать имя переменной, в которую Блок должен поместить результат своей работы.

Блок «Выполнить выражение» (InvokeExpression) в свойствах выражения (Expression) позволяет использовать выражения расширенного формата, которые похожи на выражения на скриптовом языке PowerShell. В них в том числе можно



использовать явные присвоения. Выражение записывается \$Имяпеременной, затем следует знак равенства (=), а за ним значение, которое вы хотите присвоить. Для указания переменной используйте ключевой символ \$. Для присваивания значений переменным использоваться оператор присваивания «=».

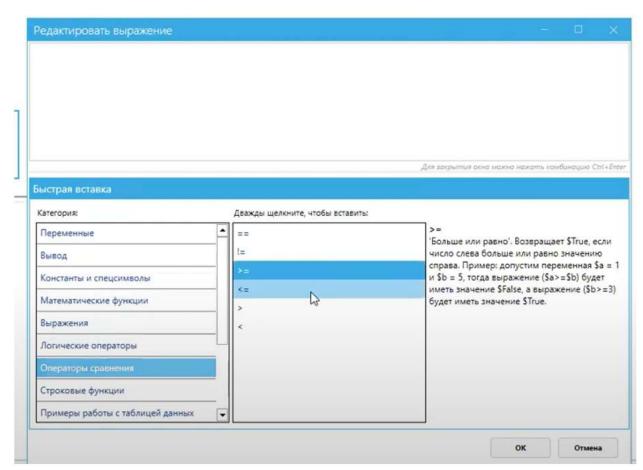
3.3.4.1 Функции в выражениях

Функции предназначены для удобных манипуляций над значениями переменных внутри выражений. Некоторые из задач, которые решаются функциями, могут быть также решены отдельными блоками, однако использование функций внутри выражений позволяет сократить количество блоков и промежуточных переменных в диаграмме, а также ускорить разработку сценариев роботов.

Sherpa RPA предоставляет несколько десятков функций для выполнения математически операций, работы со строками, с датой и временем, со списками, словарями и другими структурами данных. При использовании функции в выражении нужно написать название функции, а затем в скобках указать один или несколько её аргументов, то есть входных данных, с которыми она будет работать.

Для удобной работы с функциями разработаны подсказки. В любом месте, где можно вызвать редактор выражений, в нижней части окна редактора приводятся категории выражений, и для каждой категории приведены возможные варианты записи функций с пояснениями, как пользоваться этими функциями.



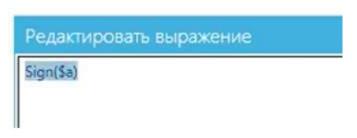


При двойном клике по строке списка функций данная функция появится в окне редактора и с её помощью можно будет составить выражение:



Аргумент функции (находящийся в скобках) можно поменять на значение какой-либо переменной. В примере на скриншоте выше – так мы узнаем, положительное или отрицательное значение у данной переменной.





Аргументами функции могут быть любые корректные выражения, однако их тип должен соответствовать тому типу данных, который данная функция ожидает увидеть в данном аргументе.

Например, функция Round() позволяет округлить значение числа, которое было передано ей в качестве аргумента, то есть указано в скобках. При этом вместо числа можно указать имя переменной, в которой содержится нужное число, или даже целое выражение, состоящее из констант, переменных и других функций. Ниже приведены примеры корректного использования функции Round().

Round(5.5) – округлит число 5.5 и вернёт результат 6. Обратите внимание, что разделителем десятичных дробей является точка, а не запятая. Если вы попробуете вызвать функцию вот так: Round(5,5), то результатом будет число 5, а не 6. Объясняется это тем, что запятая в функциях разделяет аргументы. Программа предположит, что в функцию было передано два аргумента вместо одного. Это не приведет к ошибке, но так как функция Round() ожидает только один аргумент, то будет использовать число 5 как входное значение, соответственно и на выходе также будет выдано число 5.

Round(\$МояПеременная) – если в переменную \$МояПеременная предварительно было записано число 5.5 – также вернёт значение 6.

Round(\$МояПеременная + 5) – следуя предыдущему примеру, вернет число 11.

Round(\$МояПеременная + Round(\$МояПеременная)) -следуя предыдущим примерам, вернет число 12.

Round("Привет!") – приведет к ошибке, так как в качестве аргумента функция Round() ожидает число, а не текстовую строку.

Если вы подставите функцию Round() или какую-либо другую функцию в выражение, то результат работы функции будет использован при вычислении



значения выражения в том месте, в котором вы подставили функцию. То есть результат вычисления выражения «5 + Round(5.5)» будет 11.

Список всех доступных функций вместе с описаниями их работы и примерами применения вы можете найти в справочнике в нижней части окна «Редактировать выражение».

3.3.4.2 Методы в выражениях

Методы похожи на функции, однако они всегда относятся к переменной определенного типа (например, к строке, к числу, к словарю) и служат для работы именно с этим типом переменной. При этом название метода записывается после названия переменной через точку. Метод может не иметь аргументов, и тогда после него не нужно писать скобки. Некоторые методы, как и функции, могут иметь один или несколько аргументов, которые записываются в скобках после имени метода. Методы могут только возвращать значение, а могут изменять значение и состояние той переменной или того объекта, к которому принадлежат.

Примеры использования методов:

\$МояТекстоваяСтрока.Length – метод Length используется с текстовыми строками, списками, словарями и некоторыми другими составными типами данных. При использовании с текстовыми строками метод Length возвращает их длину в символах (количество символов, из которых состоит строка), а при использовании с составными типами данных (списками, словарями) – возвращает количество их элементов. Метод Length не требует указания аргументов и поэтому используется без скобок.

\$MyString.Replace(«text», «newtext») - метод Replace используется с текстовыми строками, ищет в строке все вхождения текста, заданного в первом аргументе, и меняет их на текст, заданный во втором аргументе. Как и с функциями, в качестве аргументов можно использовать не только константы, но и другие переменные, а также выражения с ними. Метод Replace не изменяет исходную строку, а возвращает результат своего выполнения. То есть его, например, можно использовать в блоке Assign, чтобы присвоить возвращенную в результате работы этого метода строку новую другой переменной.



Методы могут быть использованы друг за другом «цепочкой». Например, следующее выражение корректно, и его результатом будет длина новой строки, полученной после замены всех вхождений «text» на «newtext» в строке, хранящейся в переменной \$MyString:

\$MyString.Replace(«text», «newtext»).Length

Некоторые методы дают доступ к отдельным элементам составных объектов, и тогда их называют «свойствами». Например, если вы используете блок Get POP3 Mail Messages для получения новых email-сообщений, полученные сообщения будут сохранены в переменной, указанной в свойстве Result. Допустим, там указана переменная \$Result. Тип этой переменной – список объектов типа MailMessage. Допустим, вы хотите получить доступ к адресу отправителя первого из полученных сообщений. Так как \$Result в данном случае - это список, можно получить доступ к первому его элементу через указание индекса в квадратных скобках: \$Result[0]. Не забывайте, что индексы всех состаных элементов начинаются с нуля, а не с единицы. Так мы получим доступ к составному объекту MailMessage, в котором хранится вся информация о каждом сообщении в отдельности. Справочник в нижней части окна «Редактировать выражение» нам подсказывает, что адрес отправителя сообщения содержится в свойстве From и его подсвойстве Address. Таким образом, чтобы получить доступ к адресу отправителя в первом полученном сообщении, нам нужно воспользоваться вот таким выражением:

\$Result[0].From. Address

Результат вычисления этого выражения будет представлять собой текстовую строку. Её можно затем вывести на экран, записать в файл, в ячейку таблицы и т.д. Вспомнив о том, что методы и свойства можно «нанизывать» друг на друга, мы можем узнать длину адреса отправителя письма в символах с помощью вот такого выражения:

\$Result[0].From. Address.Length

Или же мы можем заменить в адресе отправителя письма символ «@» на строку «-at-» с помощью вот такого выражения:

\$Result[0].From. Address.Length.Replace(«@», «-at-«)



Как и в случае с функциями, список всех доступных методов, их аргументов и примеры их использования вы можете посмотреть в справочнике окна «Редактировать выражение». Там же вы можете найти примеры формирования аргументов некоторых специальных функций. Например, в разделе «DataTime Formats» содержатся примеры аргументов для функций и блоков форматирования даты и времени. И там же вы сможете найти примеры разложения на отдельные свойства составных объектов. Например, все свойства упомянутого выше объекта MailMessage содержатся в разделе «Mail Message examples» справочника.

3.3.5 Примеры использования переменной и работы с выражениями 3.3.5.1 Простое арифметическое действие

Допустим, есть переменная, и в неё подставлять значение и выполнять арифметическое действие, указанное на иллюстрации ниже, то есть: \$Имя Переменной+100.

Примеры:					
Задача	Выражение				
К результату предыдущего шага всегда прибавлять 100	\$ИмяПеременной +100				
Посчитать количество строк в таблице	\$MyTable.Rows.Count				
	1				
	Имя Функция переменной				

Если в первом примере цифру 100 заключить в кавычки, это будет уже не числовая операция, а операция над строками, т.е. к одной строке прибавляется другая строка, и к текстовому значению, содержащемуся в переменной, будет приписано значение «100» как часть строки.

Свойства составных типов данных

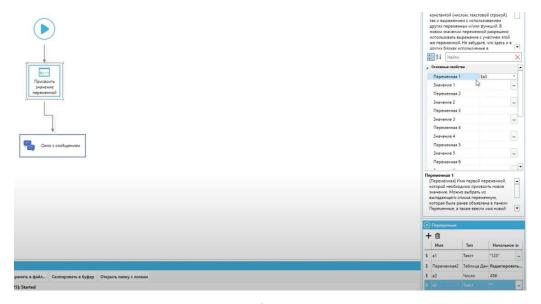
У некоторых составных типов данных есть возможность вызывать свойства или функции. Например, у таблицы MyTable есть свойство Rows (Строки), а у этого свойства есть свойство Count (количество). Запись выражения MyTable.Rows.Count позволяет получить количество строк в данной таблице.



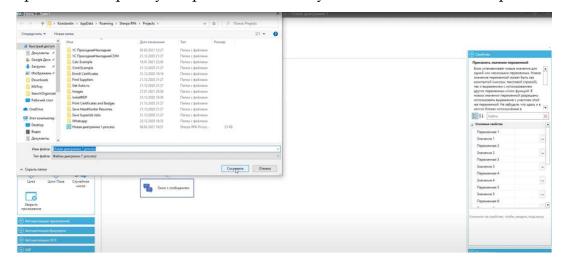
3.3.5.2 Сложение числа и строки

Существует блок, выводящий на экран сообщение, и в сообщении можно записать переменную любого типа (строку, число, таблицу или список). Также есть блок, позволяющий присваивать значение переменной, причём даже не одной, а можно использовать до 10 переменных.

Имеется переменная а1, в которой хранится какой-то текст, допустим, 123. Чтобы обозначить значение как текст, заключим его в кавычки. Также есть переменная а2, в которой хранится число, присвойте ей значение 456. Требуется соединить блоки таким образом, чтобы сначала присвоить значение переменным, а потом вывести полученный результат на экран в виде сообщения. Для этого потребуется вспомогательная текстовая переменная а3, у которой нет никакого начального значения.

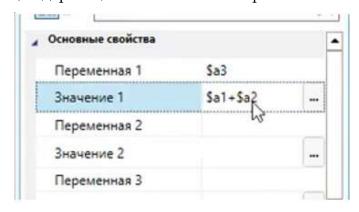


Сохраните диаграмму и проект, чтобы запустить созданный сценарий.



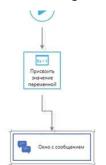


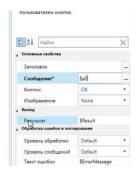
Требуется соединить содержимое переменных а1 и а2. Логично, что при сложении числа и строки в результате получится текстовая строка. Выберите в качестве целевой переменной а3, в которую будет помещен результат вычисления. В качестве значения переменной а3 напишите а1+а2, т.е. результатом будет текстовая строка, содержащая значения этих переменных: 123456.



Обратите внимание: т.к. используются не абсолютные значения и не константы, а именно переменные, то перед именем каждой переменной требуется ставить символ \$.

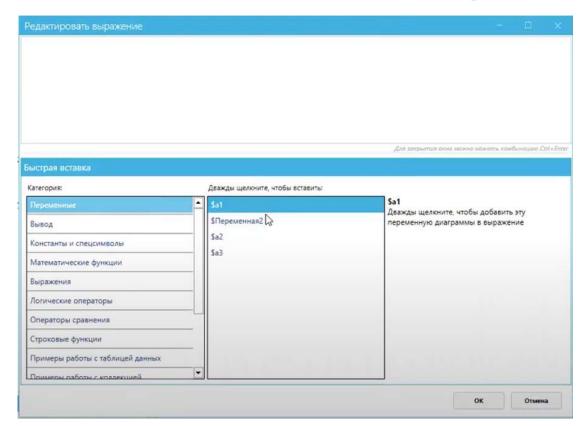
Далее в окне с сообщением нужно вывести значение переменной а3. Можно вписать переменную в поле «Сообщение» в свойствах блока:





А можно выбрать из списка в окне редактора выражений, нажав кнопку с тремя точками справа от поля. Все пользовательские переменные указаны для категории «Переменные».





Далее необходимо дважды кликнуть по строке с переменной а3 и нажать «ОК». После этого сохраните и запустите сценарий. Результат:

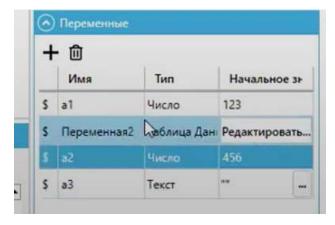


Таким образом, вычисления проводятся в первом блоке, а второй блок только выводит результат выражения на экран.

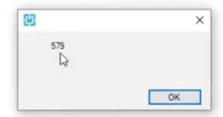
3.3.5.3 Сложение двух чисел

Для эксперимента измените тип переменной а1 на число и уберите кавычки из значения переменной.





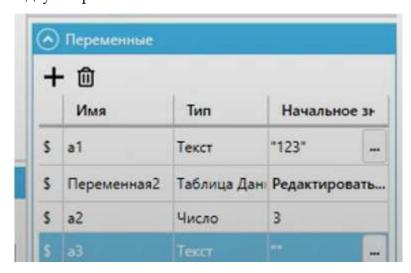
Больше в сценарии ничего не меняйте. Запустите сценарий и посмотрите результат:



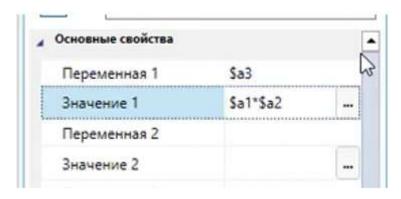
Таким образом, сменив тип переменной и начальное значение, мы получаем результат вычисления числового выражения (сложения двух чисел – значений переменных a1 и a2).

3.3.5.4 Умножение текста на число

Снова измените тип переменной а1 на текст и заключите значение 123 в кавычки. а2 останется числом, но значение измените на небольшое, допустим, на 3. В блоке «Присвоить значение переменной» в качестве результата укажите перемножение этих двух переменных.

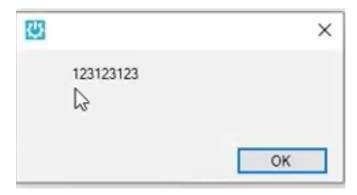






Т.е. пытайтесь умножить текст на число.

Запустите сценарий.



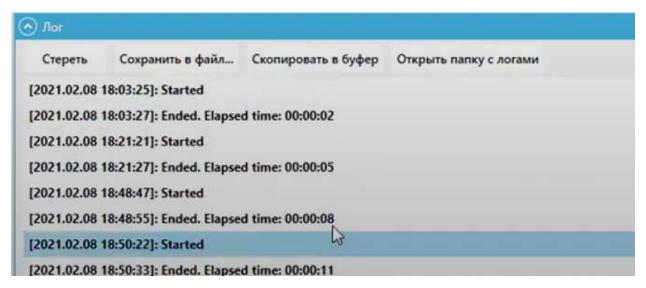
Текст «123» повторился три раза, т.е. строку умножили на 3. На практике это бывает нужно редко, но в качестве примера интересно посмотреть, как робот обрабатывает те или иные типы данных в разных комбинациях.

3.3.5.5 Лог

В нижней части окна дизайнера расположен заголовок «Лог».

Здесь выводятся сообщения о процессе запуска робота и о процессе его исполнения. Здесь же можно видеть сообщения, которые можно отправить в лог напрямую посредством блока «Лог», а также сообщения о возникающих в процессе работы ошибках.



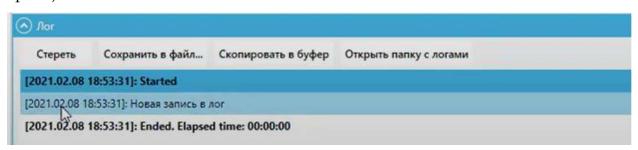


Отправьте сообщение в лог. Подключите блок «Лог», задайте в свойствах значение «Новая запись в лог».



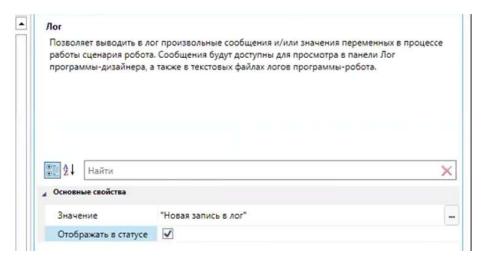
Сохраните. В логе нажмите кнопку Стереть, чтобы удалить старые записи. Запустите робота.

В логе появились три новые записи: о том, что робот стартовал, далее заданный текст, и запись о том, что робот закончил работу (указано затраченное время).

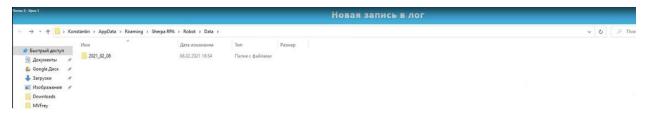


У блока «Лог» есть свойство «Отображать в статусе».

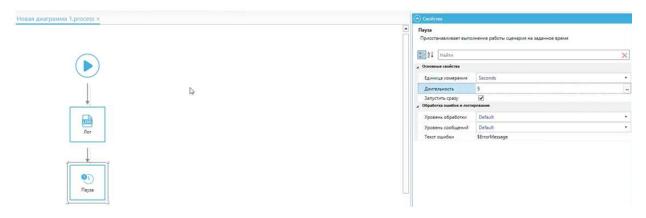




Это галочка и, если её проставить, то в верхней части экрана будет появляться сообщение, дублирующее заданное в свойствах значение. Запустите робота и убедитесь в этом.



Сообщение мелькает вверху экрана очень быстро и, если необходимо его задержать, можно использовать блок «Пауза». Он приостанавливает выполнение действия сценария на заданное количество секунд или миллисекунд. Приостановите выполнение сценария на 5 секунд.



Важное свойство данного блока – «Запустить сразу», т.е. задержка будет применена сразу при первом проходе этого блока с момента запуска робота. Получается, что при отсутствии галочки задержка применяется только со второго прохода блока. Оставьте галочку и запустите сценарий. Сообщение «Новая запись в лог» задерживается вверху экрана на 5 секунд. Напомним, что т.к. свойство значения имеет рядом с полем кнопку с тремя точками, то нажатием на эту



кнопку можно вызвать редактор выражений и прямо здесь производить различные вычисления и пр.



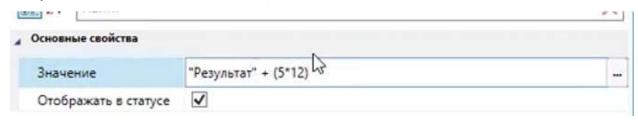
Можно ввести, допустим, выражение 5*12.



Результат:



Можно добавить текст. Добавьте к значению текстовую константу "Результат" (кавычки обязательны).

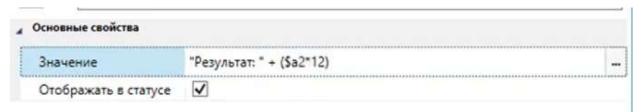


Скобки указывают на первоочередное выполнение действия. Сохраните и запустите робота.



Т.е. робот вначале выполнил арифметическое действие, а потом при попытке сложить текст и число, преобразовал результат в текст.

Также можно взять переменную a2, присвоить ей значение 5, и в свойствах лога в выражении заменить число 5 на значение переменной a2.





Переменная а2 является числом и хранит начальное значение 5, таким образом, результат на выходе меняться не должен.



Результат не изменился.

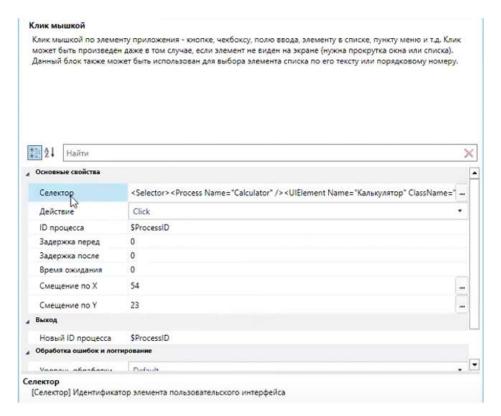
3.3.6 Настройка селекторов

Программному роботу для имитации конкретных действий в интерфейсе пользователя (операционной системе, программах, веб-страницах) необходимо взаимодействовать с различными окнами, кнопками, раскрывающимися списками, гиперссылками и так далее. В действиях с экранными формами Sherpa RPA осуществляет поиск (перебор) положения элементов на экране. При этом выбор нужного элемента на экране – это сложная задача, перебор может осуществляться среди нескольких тысяч элементов, которые присутствуют на экране. Достаточно часто признаки элементов управления, по которым их можно найти на экране (названия, подписи, системные свойства и дескрипторы) подвержены динамическим изменениям, например, заголовок веб-страницы может измениться во время процесса автоматизации.

Для решения этой проблемы Дизайнер использует так называемые Селекторы. Селекторы хранят атрибуты элемента графического интерфейса пользователя в виде набора инструкций в формате XML, которые подсказывают роботу, как найти тот или иной элемент управления на экране.

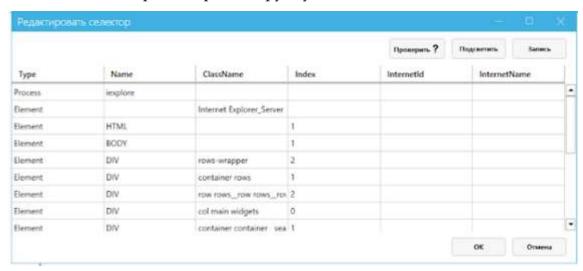
В большинстве случаев Селекторы автоматически генерируются Дизайнером в процессе записи действий пользователя и не требуют дополнительного ввода от пользователя, особенно если автоматизируемые приложения имеют статический интерфейс пользователя.





Селектор – подсказка, руководство к действию для робота, как ему найти на экране тот или иной элемент управления. Робот записывает эти селекторы автоматически при записи каких-либо действий.

Однако в некоторых программах используются изменяемые макеты и атрибуты с изменчивыми значениями, например в веб-приложениях. Дизайнер не может предсказать эти изменения, поэтому некоторые селекторы, возможно, придется создавать или редактировать вручную.



Окно «Редактировать селектор»



Окно «Редактировать селектор» вызывается из правой панели свойств селектора (только для тех блоков, в которых есть селекторы) нажатием на кнопку с многоточием. Все блоки, связанные с графическими элементами, обладают этим свойством. Селектор доступен и в случае, когда блоки добавляются в проект с помощью инструмента записи, и в готовых блоках для операций с экранными формами (UI Automation) из палитры.

Окно «Редактировать селектор» позволяет просмотреть автоматически сгенерированный селектор и изменить сам селектор и его атрибуты. Вы можете включать или исключать определенные атрибуты или добавлять подстановочные знаки в динамически изменяющиеся значения.

При нажатии на кнопку с тремя точками справа от поля селектора, появится окно «Редактировать селектор».

100				1777	0700	100		-0.00
+ Добавить новую стро	ку 🕶 🧻 Удыничь строму			+ Вставить подстановку =	+ Вставить переменную 👻	▲ Проверить ?	Подсветить	Ū Barrieci
Туре	Name	ClassName	ControlType	Index	ld		Text	
Process	Calculator							
UlElement	Калькулятор	ApplicationFrameWindow	Window					
UiElement	Калькулятор	Windows.UI.Core.CoreWindow	Window					
UlElement		LandmarkTarget	Group	1				
JI Element	Числовая панель	NamedContainerAutomationPeer	Group	6	NumberPad			
UlElement	Шесть	Button	Button	6	num6Button			

В данном окне можно посмотреть, как выглядит селектор, проверить и исправить его при необходимости, а также обогатить селектор вспомогательными инструментами, например, вставкой подстановочных символов, вставкой значения переменной. Также можно удалить лишние строки или очистить лишние значения ячеек.

Итак, селектор – это таблица, состоящая из колонок и срок, при этом каждая строка – некая команда, шаг, который робот должен выполнить. Команды читаются сверху вниз. Верхняя строка выполняется самой первой, нижняя самой последней. Робот идёт по шагам сверху вниз и выполняет каждую строку как команду. Он ищет на экране элемент, у которого все свойства одновременно равны тому, что указано в колонках строки селектора.

Например, рассмотрим первую строку селектора. Тип – процесс. Т.е. эта строка ищет необходимое приложение для запуска. Имя процесса здесь – Calculator, оно записано так, как фигурирует в диспетчере задач.





Итак, на первом этапе робот найдет процесс с названием Calculator среди запущенных процессов в памяти компьютера.

Следующая строка обозначается как UIElement, это уже окно или какой-то конкретный элемент управления этого селектора. Для того, чтобы найти нужный элемент управления, робот одновременно будет искать элемент с именем Калькулятор, указанным именем класса и элементом управления Window, отсюда понятно, что речь идёт об окне. Данная строка позволяет найти конкретное окно калькулятора.

Спускаемся на строку ниже. Теперь в найденном окне необходимо найти последовательно все контейнеры, которые приведут к нужной кнопке. Таких контейнеров здесь три. Данные контейнеры могут визуально не отображаться в приложении, т.к. они были созданы для разработчиков. Однако по названию «Числовая панель» можно догадаться, что это контейнер, содержащий все кнопки калькулятора. И далее видим, что самый последний элемент – та самая цифра 6, на которую кликал пользователь.

Кнопка «Проверить» проверяет заданное в селекторе позиционирование элемента, то есть в результате проверки может выдаваться два состояния:

- Ошибка при проверке;
- Проверка прошла успешно, элемент с заданным в селекторе положением найден на экране.

Кнопка «**Подсветить**» — указывает на элемент пользовательского интерфейса, подсвечивает сам элемент и элементы верхнего уровня на пути к нему красными рамками. Выделение пропадает автоматически через несколько секунд.

Кнопка «Запись» – запускает процедуру записи селектора путем клика на нужном элементе управления на экране.

Зайдите в Selector и нажмите Запись. Окно SherpaDesigner (с открытым окном редактирования селектора) сворачивается. В верхней части экрана



появится подсказка для работы в режиме записи. Выбирайте нужное действие для вашего бизнес-процесса, то есть нужный элемент на экране и кликайте на него. Обратите внимание, элементы перед кликом должны подсветиться (то есть должна появиться красная рамка). Если подсветка не появляется автоматически или пропадает, то для подсветки нужного элемента, необходимо нажать клавишу Shift на клавиатуре. Подсветка нужна, чтобы точно выбрать элемент под мышкой.

Варианты записи клика:

- кликнуть левой кнопкой мыши;
- нажать левый Ctrl, что эмулирует нажатие клика левой кнопкой мыши;
- нажать правый Ctrl, что эмулирует нажатие правой кнопки мыши, то есть, во многих приложениях, вызов контекстного меню;
- эмуляция задержки курсора мыши над элементом записывается с помощью клавиши Shift.

Из окна «Редактировать селектор» записывается только одно действие (клик, вставка, зависание над кнопкой и т.д.), в отличие от использования инструмента Записи через основное меню, когда вы можете записать несколько действий сразу. То есть после совершения действия (клика) пользователь автоматически возвращается в окно «Редактировать селектор».

После возврата в окно редактирования Селектора рекомендуем использовать кнопки «Подсветить» или «Проверить», чтобы проверить правильно ли выбран элемент. Если элемент выбран неправильно, то можно повторить попытку, нажав кнопку «Запись», или найти и исправить ошибку в строках селектора вручную.

Настройка селектора



Туре	Name ClassName			Проверить ?	Подсветить	одсветить Запись	
		Index	InternetId	InternetNa	ame		
Process	iexplore						
Element		Internet Explorer_Server					
Element	HTML		1				
Element	BODY		1				
Element	DIV	rows-wrapper	2				
Element	DIV	container rows	1				
Element	DIV	row rows_row rows_rox	2				
Element	DIV	col main widgets	0				
Element	DIV	container container sea	1				

Каждая строка селектора имеет один или несколько атрибутов, помогающих правильно идентифицировать точное местоположение выбранного участка экрана. Атрибуты отображают, в какой программе идет запись, в каком окне и элементе, а также конкретный уровень выбранного элемента с помощью таких атрибутов как Туре, Name, ClassName, ControlType, ID. Каждый атрибут может имеет присвоенное значение, по которому будет осуществляться поиск. Важно выбрать атрибуты с постоянным значением. Если значение атрибута изменяется при каждом запуске приложения, то селектор не сможет правильно идентифицировать элемент. В этом случае рекомендуем отказаться от использования данного атрибута (оставить его значение пустым).

Кроме того, в сложных случаях вы можете воспользоваться символами замены или подстановкой значения атрибута или части атрибута из переменной:

- Звездочка (*) заменяет любое количество символов в значении атрибута;
 - Знак вопроса (?) заменяет один символ в значении атрибута;
- \$ИмяПеременной использование имени переменной, начинающегося на символ \$, позволяет вставить значение этой переменной вместо значения атрибута или его части.

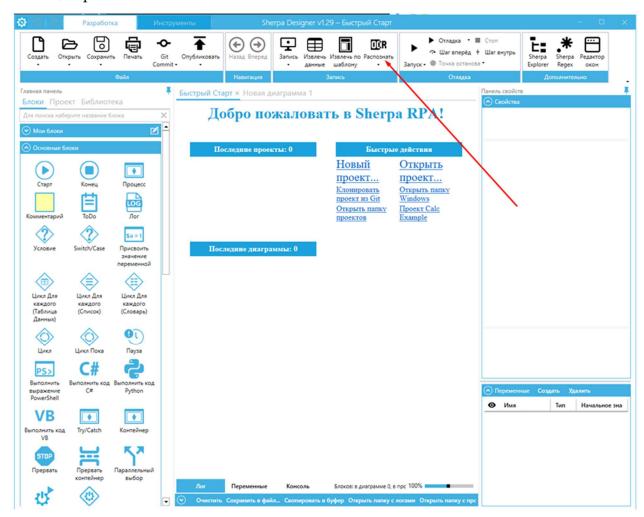
3.3.7 Использование ОСК Яндекс

Использование OCR-движка Yandex Vision в проектах

При создании сценариев роботов, в которых нужно распознавать сканкопии, фотографии (формата JPEG, PDF, TIFF или других форматов), Вы можете применять на выбор один из пяти ОСR-движков, встроенных в Платформу Sherpa 145



RPA. ОСR-движки распознают текст на изображении и возвращают в виде блоков текста, строк и слов.



Одним из лучших ОСR-движков по качеству распознавания является Yandex Vision. https://cloud.yandex.ru/docs/vision/

Требования для использования Yandex Vision в роботах

Yandex Vision – облачный сервис компьютерного зрения для анализа изображений, для его использования нужен интернет. Политиками безопасности должны быть открыты доступы на исходящие вызовы робота Yandex Vision

https://iam.api.cloud.yandex.net/iam/v1/tokens
https://vision.api.cloud.yandex.net/vision/v1/batchAnalyze

Стоимость Yandex Vision

Для тестирования и ваших экспериментов, Sherpa RPA дает 100 страниц бесплатного лимита. То есть на старте для использования вам нужно только выбрать этот движок и распознавание пройдет автоматически (то есть через аккаунт Sherpa RPA).



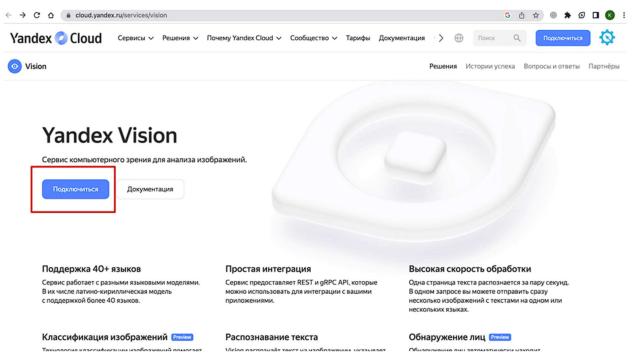
Для постоянного использования Yandex Vision вам нужно оформить оплату сервиса. При этом вы можете выбрать один из способов:

- Оплатить кредиты страниц через нас. Шерпа Роботикс как партнер Яндекса предложит вам договор. Стоимость распознавания в настоящий момент 0,13 руб за страницу (т.е. 13 копеек).
- 2) Создать собственный аккаунт в Yandex Cloud и прописать в настройках робота токен вашего аккаунта.

Инструкция по настройке

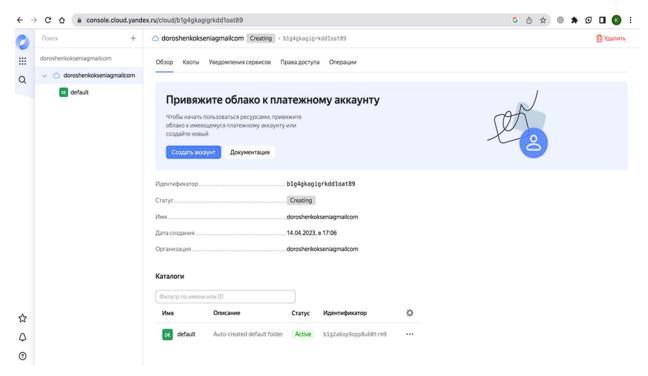
1) Для работы должен быть создан аккаунт Yandex Vision для вашей компании

https://cloud.yandex.ru/services/vision



2) Привяжите облако Yandex Vision к вашему платежному аккаунту:





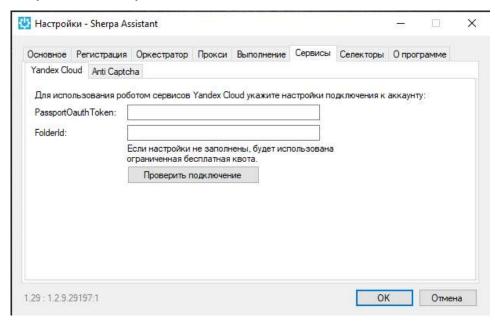
3) Список серверов, к которым должен быть открыт доступ (т.е. исходящие запросы должны проходить к этим трём доменам):

https://iam.api.cloud.yandex.net/

https://vision.api.cloud.yandex.net/

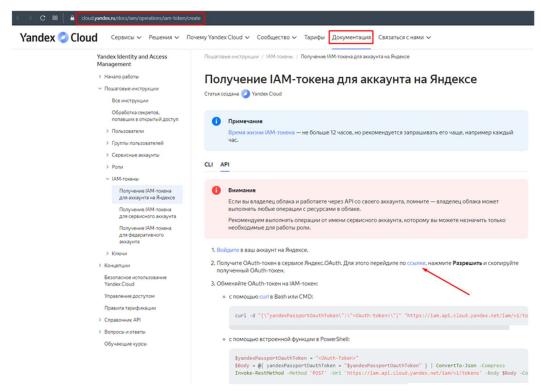
http://sherpa.wiregeo.com/

4) На стороне Sherpa RPA (в настройках) Нужно заполнить два параметра: как указано на скриншоте — это заполняет разработчик, после того, как ему передадут соответствующие данные.

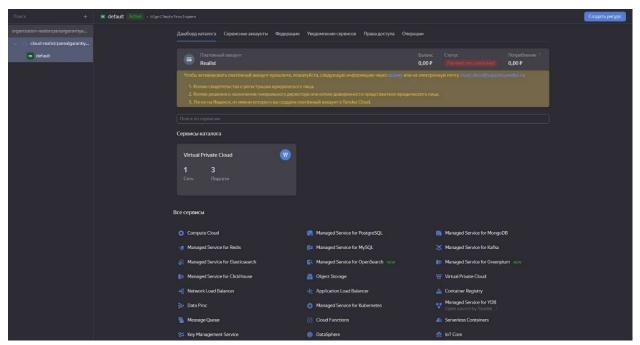




5) Для получения PasswordOauthToken (из аккаунта Яндекса — из Cloud Console) необходимо перейти по адресу, который подчеркнут в адресной строке и далее нажать на выделенную ссылку, там будет указан данный токен.



6) Получение FolderId описано вот в этой статье: https://cloud.yandex.ru/docs/resource-manager/operations/folder/get-id Пример, как в нашем аккаунте выглядит FolderId:





Безопасность Yandex Cloud

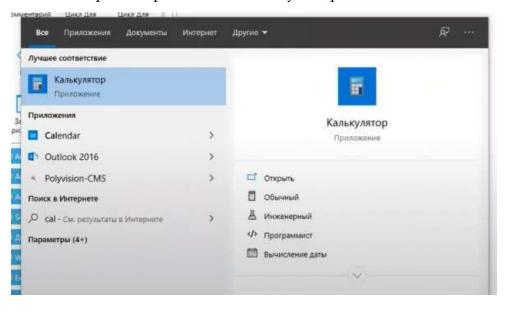
В Yandex Cloud действуют меры по защите персональных данных (ПДн), указанные в постановлении №1119 и приказе ФСТЭК №21 в соответствии с требованиями к 1 уровню защищенности (УЗ-1).

Когда клиент размещает на ресурсах Yandex Cloud персональные данные, в отношении которых он выступает оператором, он поручает Яндексу обработку этих персональных данных. Yandex Cloud обязуется соблюдать конфиденциальность ПДн, обеспечивать их безопасность при обработке и выполнять все требования к защите обрабатываемых ПДн, установленные законодательством.

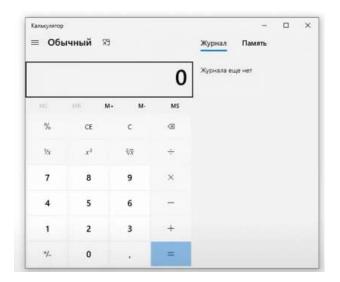
3.3.8 Примеры автоматизации популярных приложений

3.3.8.1 Автоматизация калькулятора

Рассмотрим на практике работу с записью на примере калькулятора. Запустите стандартное приложение Калькулятор.





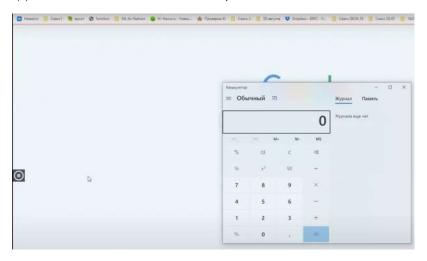


Задача: роботом нажимать на кнопки Калькулятора.

Нажмите кнопку «Запись последовательности кликов». В верхней части экрана появляется подсказка насчёт использования дополнительных клавиш:



Внешний вид панели записи изменился, осталась лишь кнопка Отмена.



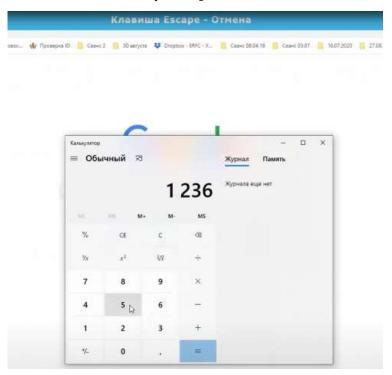
По мере того, как курсор мыши перемещается по экрану, элемент под курсором выделяется красной рамкой.





Причём это происходит и в Калькуляторе, и в браузере, и на панели задач и т.д.

Кликните по нескольким кнопкам Калькулятора левой кнопкой мыши. В верхней части экрана появляется подсказка о том, что в любой момент мы можем завершить запись нажатием на клавишу Escape.



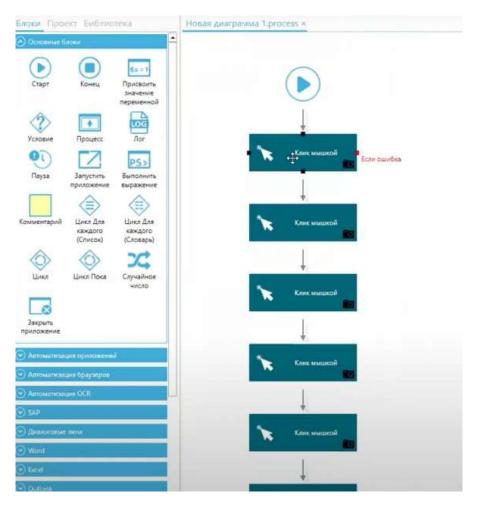
Обратите внимание, сейчас вы кликали мышью. Т.е. можно нажать на Ctrl и записать клик мышью на заданном элементе, но не производить самого клика. Это иногда бывает полезно – если вы хотите просто записать клик, но не выполнять никаких операций и перейти на другой экран. Если это не критично, записывайте клик просто нажатием на левую кнопку мыши.



Теперь у кнопки «Сохранить» и «Закрыть» на панели записи в верхнем углу появилась цифра 8. Это значит, что мы записали 8 действий.

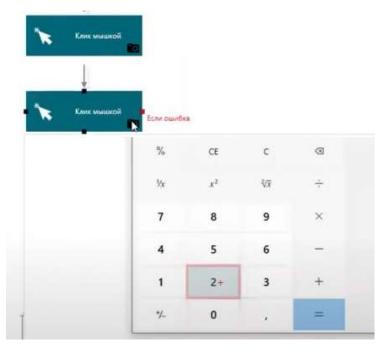


Теперь вы можете нажать «Escape» для выхода из режима записи, и затем нажмите на кнопку «Сохранить» и «Закрыть». В Дизайнере появились блоки с записанными действиями:



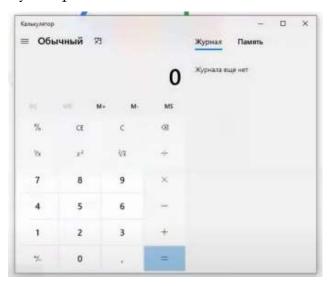
Можно подвести курсор к иконке с фотоаппаратом и просмотреть скриншот, на котором указано, что за действие записано и выполняется данным блоком. Точка на скриншоте, куда был произведён клик, помечается крестиком.



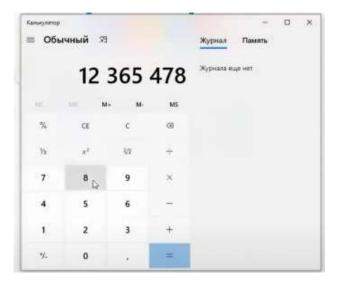


Сохраните диаграмму и запустите её.

Загружается Калькулятор.



Выполняются записанные клики на клавиши.



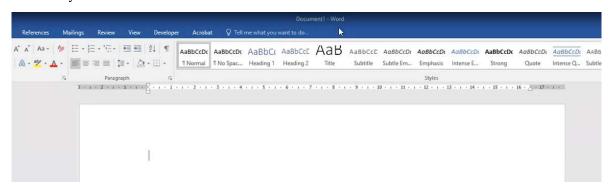


Робот завершил данный сценарий.

3.3.8.2 Точная настройка селектора в Microsoft Word

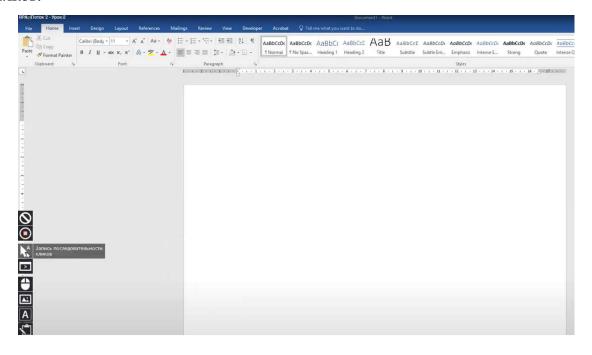
Предположим, вы работаете с текстовым редактором. В заголовке окна текстового редактора вы увидите не только его название (MS Word и т.д.), но и через дефис имя текущего документа. Допустим, вы не хотите, чтобы в данном сценарии робот работал только с этим документом и ни с каким другим.

Запустите Word.



В заголовке окна текстового редактора отображается имя текущего документа и через дефис название редактора. Допустим, робот должен найти окно редактора Word, но имя документа не важно.

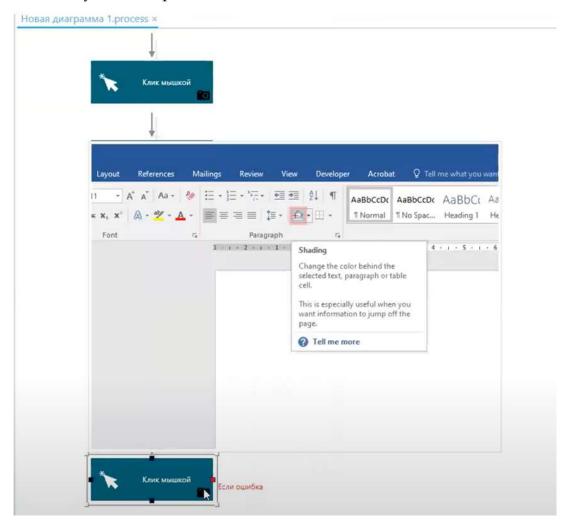
Попробуйте записать клик по какой-нибудь кнопке в окне Word, например, на кнопке заливки цветом. Нажмите на кнопку для записи последовательности кликов.



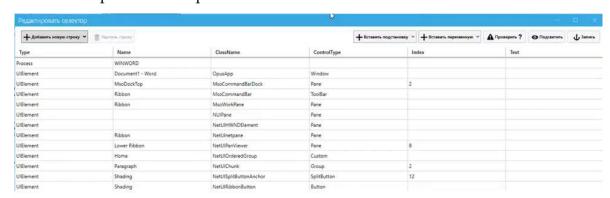


В качестве эксперимента нажмите клавишу Ctrl, а не кликните левой кнопкой мыши. Т.е. действие записалось без фактического клика по кнопке. Нажмите на «Еscape» для отмены и на кнопку (Сохранить и закрыть).

В конец текущей диаграммы добавился блок клика мышью:



Посмотрите селектор блока.



Первой строкой происходит поиск приложения WINWORD. Второй строкой происходит поиск окна с заголовком Document1 – Word, т.е. как раз поиск документа с конкретным названием Document1. Это не входит в условие задачи.



Необходимо сделать так, чтобы сценарий работал независимо от того, какой документ открыт. Самое простое, что можно сделать – удалить содержимое этой ячейки и сохранить таблицу селектора в таком виде.

Как правило, других условий, содержащихся в этой строке, достаточно, чтобы робот нашёл нужный элемент управления. В строке остаются такие свойства, как имя класса и тип элемента управления.

Ниже представлены способы проверить, сработает ли робот правильно в таком случае.

Первый вариант: с помощью кнопки Проверить вверху таблицы селектора.

Редистировать селектор — - П								П ×
+ Добавить новую строку 🕶 👚 Удалить строку				+ Вставить подстановку 🕶	+ Вставить переменную 🕶	▲ Поовериль ?		⊕ 3annes
Туре	Name	ClassName	ControlType	le le	ndex	Text		
Type Process	WINWORD							
UlElement		OpusApp	Window					
UlElement	MsoDockTop	MsoCommandBarDock	Pane	2				
UlElement	Ribbon	MsoCommandBar	ToolBar					

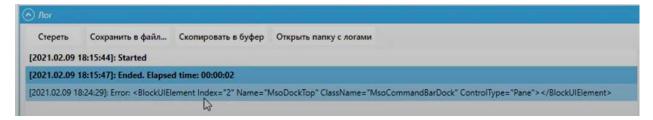
Кнопка выполняет формальную проверку на соответствие хоть какогонибудь элемента на экране всему списку команд, которые указаны в селекторе.

Второй вариант: Кнопка Подсветить позволяет более интерактивно и наглядно показать процесс поиска роботом того или иного элемента управления.

Нажмите на кнопку Проверить. Дизайнер сворачивается, но проверка закончилась неудачно.



При нажатии на кнопку Подсветить видно, что определилось окно приложения Word, но затем возникла некая ошибка. Ошибку можно найти в логе.



Очевидно, проблема возникла на второй строке, в ячейке ClassName со значением MsoDockTop. Этот элемент не найден. Теоретически проблемы быть не должно, однако это не так.

Запишите ещё раз селектор и не редактируйте его. Нажмите на кнопку для записи последовательности кликов. Нажмите на кнопку для заливки цветом.



Перейдите в дизайнер и откройте селектор нового блока. Нажмите на кнопку Проверить. Однако даже в исходном виде сценарий не работает. Судя по логу, робот так же останавливается на второй строке.



Иногда бывает такое, что при наводке курсора мыши на панель что-то в свойствах этой панели меняется. Самая популярная проблема при запуске селекторов – у какой-то кнопки или ссылки на сайте меняется имя класса. Когда курсор мыши наведен на элемент, имя класса одно. Когда курсор мыши уходит с элемента, имя класса меняется – так устроена сама страница. Скорее всего, так произошло и в данном случае, когда вы записывали клик на кнопку заливки цветом, наводя на нее курсор.

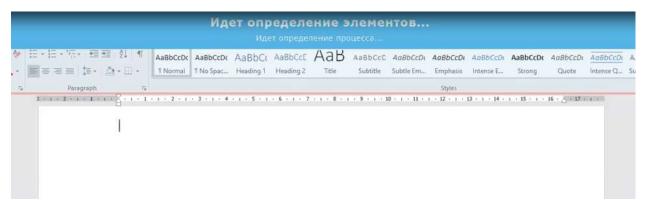
Рассмотрим, в этом ли проблема и какие действия необходимо предпринять для отладки селектора.

Для начала попробуем ослабьте одно из условий, находящихся в проблемной строке. Пробуйте начать с условия Index. Удалите значение этой ячейки (2), нажмите на кнопку «Проверить». Открывается окно с заголовком Document1 – Word, т.е. всё работает правильно. В таблице селектора на кнопке Проверить появляется зелёная галочка. Формальная проверка селектора пройдена.

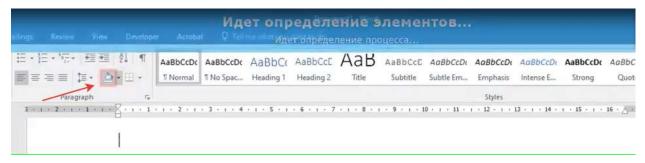
+ Добавить новую строку 🕶 📋 Удалить строку				+ Вставить подстановку 👻 + Вставить переменную		▲ Проверить у	 Подсветите
Туре	Name	ClassName	ControlType		Index	Text	
Process	WINWORD						
JElement	Document1 - Word	OpusApp	Window				
JiElement	MsoDockTop	MsoCommandBarDock	Pane				
JiElement	Ribban	MsoCommandBar	ToolBar				
JiElement	Ribbon	MsoWorkPane	Pane				
JElement		NUIPane	Pane				
DElement		NetUIHWNDElement	Pane				
liElement	Ribbon	NetUlnetpane	Pane				
JiElement	Lower Ribbon	NetUlPanViewer	Pane		8		
JIElement	Home	NetUlOrderedGroup	Custom				
JiElement	Paragraph	NetUlChunk	Group		2		

Нажмите кнопку «Подсветить». После нажатия на эту кнопку робот пройдет последовательно по всем шагам – строкам селектора и будет каждый раз подсвечивать рамочкой текущий шаг, т.е. тот элемент на экране, на котором он сейчас находится.





Сначала робот нашёл и выделил всё окно, затем в окне нашёл панель инструментов и выделил её, потом внутри панели инструментов нашёл и выделил секцию Paragraph и затем в ней нашёл кнопку заливки цветом.



Причём робот нашёл эту кнопку по внешнему контуру, т.к. кнопка состоит из двух частей: сама кнопка и стрелка, вызывающая палитру. На последнем шаге робот выделил уже саму кнопку с заливкой.

Таким образом, вы «отремонтировали» селектор.

Иногда можно просто удалить всю проблемную строку, а не угадывать, какое из условий здесь лишнее. Это будет работать в том случае, если на экране внутри контейнера верхнего уровня больше нет ещё одного экземпляра следующей строки. В данном селекторе под такое условие подходит строка после проблемной:



Попробуйте удалить проблемную строку.



Сохраните селектор, снова в него зайдите и нажмите кнопку Проверить. Проверка пройдена успешно.





Вернитесь к исходному примеру. Необходимо, чтобы селектор был универсальным и работал с любыми документами Word, а не только с Document1.

Удалите в селекторе содержимое этой ячейки и проверьте работу селектора.

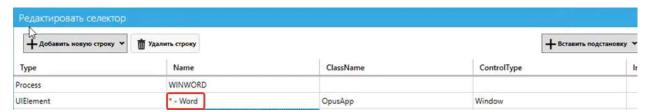


Селектор работает.

Существует другой способ – существует функционал вставки подстановочных символов. Можно вставить знак вопроса, который будет заменять один любой символ, или звёздочку, которая будет заменять несколько любых символов.

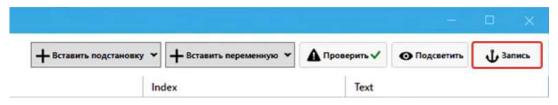


T.e. вы можете записать значение ячейки Name во второй строке как "* — Word"



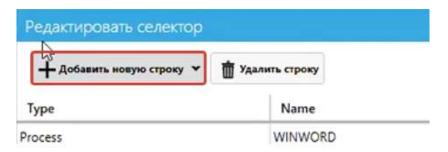
Такой приём бывает полезен, если нужно уточнить роботу, с каким из множества однотипных элементов мы имеем дело.

Если вы недовольны результатом, можно перезаписать селектор с помощью кнопки Запись в таблице селектора.



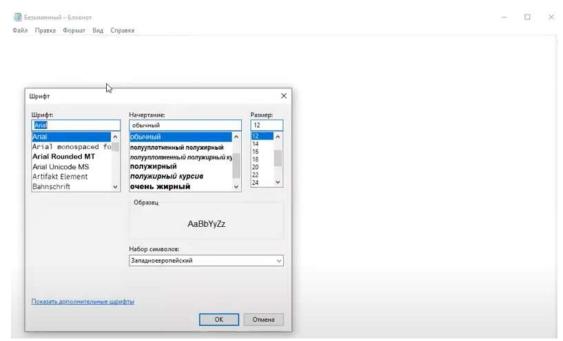


Также можно добавить вручную новую строку с помощью кнопки «Добавить новую строку» и прописать какие-то параметры в сроке селектора автоматически.



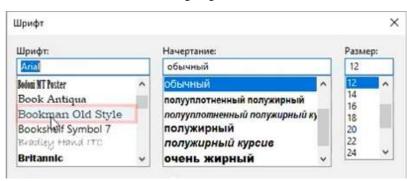
3.3.8.3 Автоматизация выбора шрифта в Блокноте

Рассмотрим, как можно с помощью робота автоматизировать выбор шрифта в блокноте.



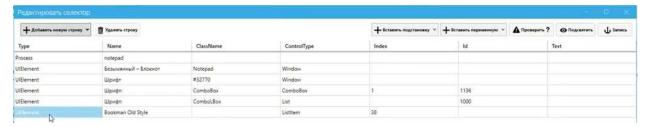
Предположим, требуется, чтобы робот выбрал в этом окне необходимый шрифт. Задача усложнена наличием полосы прокрутки.

Нажмите на кнопку Запись последовательности кликов и запишите клик на каком-либо пункте из длинного списка шрифтов. Кликните на BookmanOldStyle.





Блок добавился в текущий процесс. Рассмотрим селектор.

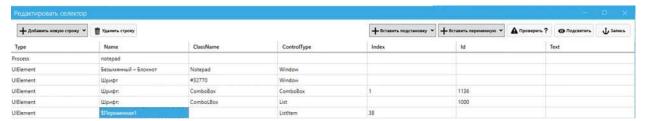


Обратите внимание на последнюю строку. В ней есть имя шрифта и его порядковый номер в списке — 38 (в колонке Index). Получается, что на компьютере, где мы собираемся выполнять сценарий, мы должны всегда иметь этот шрифт в списке шрифтов под номером 38. Если условие не выполнится, робот не выберет этот пункт. Можно сделать так, чтобы робот выбирал именно шрифт BookmanOldStyle безотносительно его номера в списке, т.к. в разных ОС номер может различаться. Или, наоборот, будет выполняться перебор позиций списка, так, чтобы был выбран шрифт под конкретным номером независимо от его имени. Мы можем либо удалить порядковый номер, и тогда мы будем искать конкретно BookmanOldStyle, либо удалить название шрифта и искать по индексу. И здесь будет полезно, как обычно при задачах перебора, подставлять переменные в ячейки селектора. Это можно сделать с помощью кнопки "Вставить переменную".

Создадим Переменную 1 текстового типа.



Откроем редактор селектора. Можно вставить переменную с помощью кнопки или просто вписать её имя вручную в ячейку.



Теперь робот будет искать шрифт, хранящийся в Переменной1. Не забывайте, что перед именем переменной необходимо ставить символ \$. Значение колонки Index нужно будет удалить, чтобы поиск производился независимо от позиции шрифта в списке.



Если Переменная1 является числом, то можно подставить её значение в колонку Index.



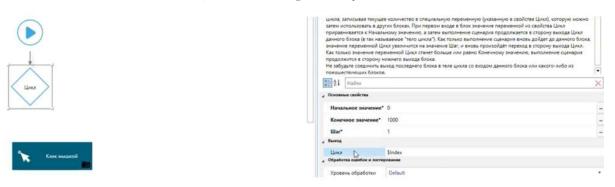
Тогда мы можем сделать цикл, в котором пробегаем по всем значениям этой переменной, допустим, от 1 до 100, и возьмём содержимое каждого элемента этого списка в текстовую строку, чтобы понять, какие вообще шрифты имеются в системе. Это можно сделать с помощью блока «Получить текст».

Как в таком случае понять, какое количество элементов в этом списке? Чтобы пробежать весь список, нужно знать количество элементов. Это не так важно, т.к. мы можем заранее задать какое-то большое количество элементов, например, 3000, которого точно никогда не будет. И далее можем воспользоваться коннектором «Если ошибка» для того, чтобы выйти из нашего цикла.

Как это выглядит на практике:

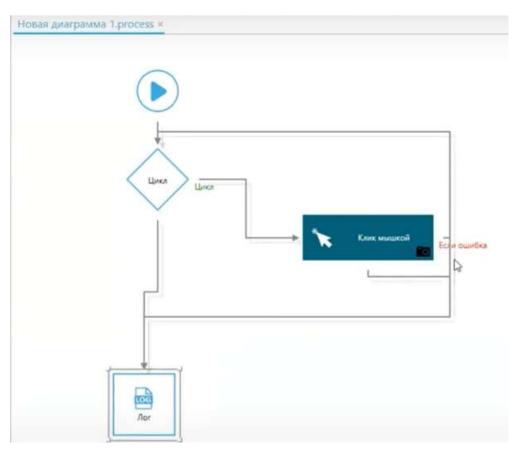
Имеется цикл. Его начальное значение 0 или 1. Важный момент: в разных приложениях и веб-страницах могут быть разные начальные значения элемента. Это может быть 0 или 1. Нужно проверять заранее.

Зададим для нашего цикла начальное значение 0, конечное значение 1000. В качестве итератора цикла (переменной, в которую каждый раз будет записываться новое значение) возьмем переменную Index.



Теперь подключим к выходу цикла блок «Клик мышкой», т.е. просто прокликаем по всем этим пунктам. Выход блока «Клик мышкой» соединим с началом цикла и поставим следующий блок – «Лог».





Если клик мышью будет неуспешным, т.е. мы не найдём индекс какого-то пункта этого списка, возникнет ошибка. Соединим коннектор «Если ошибка» с входом следующего после цикла блока (Лог). Пропишем нашу переменную Index в селекторе.



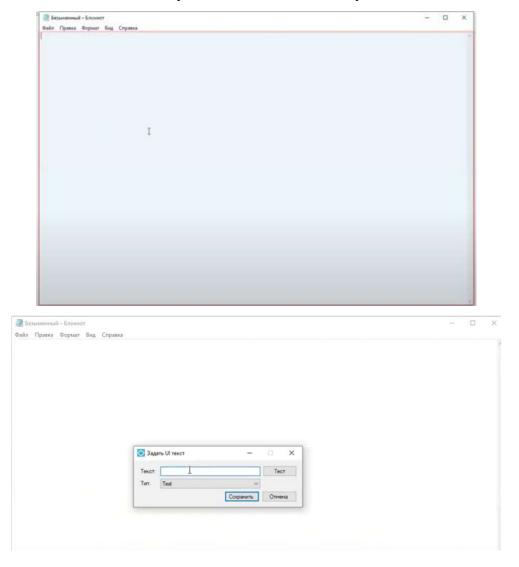
Таким образом, мы сможем быстро пройти и прокликать по всем пунктам этого списка, даже если мы не знаем, сколько там значений. Возвращаясь к примеру с калькулятором, мы так делали в одном проекте, где нужно было показывать клики мышкой по разным кнопкам, выбирать произвольные числа, складывать их и вычитать и т.д. Можно было, конечно, записать 10 кликов по 10 кнопкам, а потом выбирать с помощью 10 условий (если значение переменной = 1, то кликаем по такой-то кнопке и т.д.). Но мы обощлись одним блоком клика, т.к. внутри этого блока в селекторе, в колонке Index прописали переменную Index, содержащую порядковый номер.



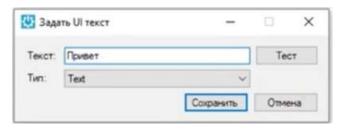


В зависимости от того, какое будет значение в этой ячейке, на эту кнопку робот будет кликать.

Кнопки для записи действий работают достаточно однотипно. Например, откроем Блокнот и нажмём кнопку «Задать текст элементу».

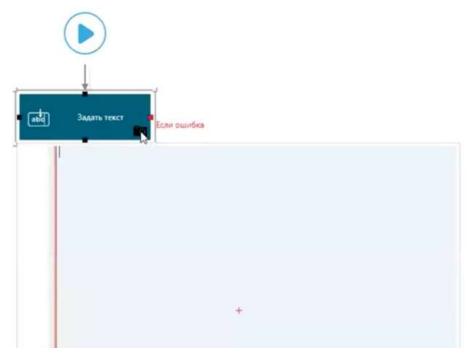


Зададим текст и сохраним.

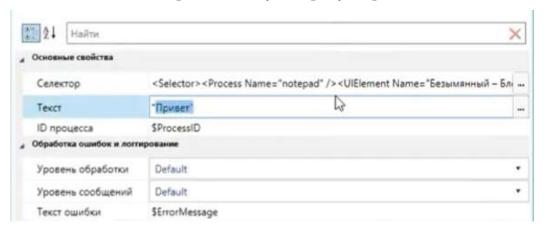




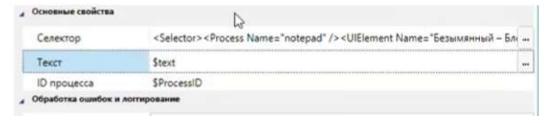
Блок добавляется в текущий проект.



У него также есть селектор, впишем туда строку "Привет".



Если нужно записать это не константой, а переменной, тогда необходимо вписать в это поле имя соответствующей переменной.



Остальные блоки записи кликов работают похожим образом.

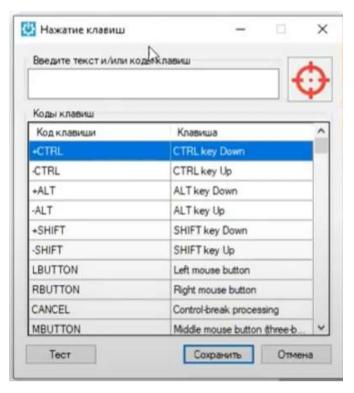
3.3.8.4 Автоматизация поиска в Яндексе

Работа с браузерами и десктопными приложениями в Sherpa RPA максимально унифицирована.



Рассмотрим задачу ввода текста и клика в строке Яндекса. Работая с этой задачей, можно увидеть сразу несколько потенциальных проблем и научиться эти проблемы решать. При записи клика на кнопку «Найти» обратите внимание: при наведённом на кнопку курсоре цвет кнопки становится чуть темнее. Это значит, что в момент наведения курсора на кнопку у неё меняется стиль, а значит, меняется селектор. Кроме того, при попытке вписать или вставить что-то в строку поиска Яндекса, в поле редактирования работает javascript, который выполняет мгновенный поиск и мешает стандартному способу вставки текстов в поля редактирования. Чтобы решить эту проблему, воспользуемся функцией эмуляции нажатия кнопок клавиатуры.

Запишем небольшой сценарий. Нажмите на кнопку Запись. Нажмите кнопку для эмуляции нажатия клавиш клавиатуры. Появилось окошко, с помощью которого можно настроить данную функцию.



Во-первых, необходимо с помощью красного прицела указать на тот элемент управления, с которым вы хотите работать.

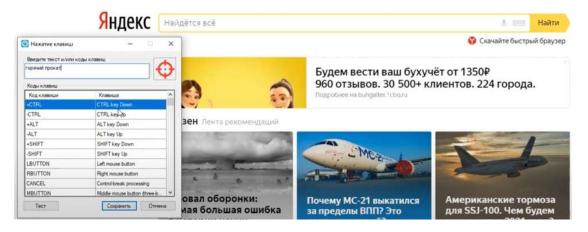
Обратите внимание на важный момент, который часто встречается при работе с веб-сайтами. Как только курсор переходит за границу поля редактирования, снаружи поля редактирования выделяется красной рамкой другой прямоугольник, который чуть-чуть больше поля редактирования.





Дело в том, что веб-страницы тоже состоят из множества вложенных друг в друга контейнеров, элементов и пр., и проблема с несколькими уровнями вложенности на веб — страницах выражена гораздо сильнее, чем в обычных приложениях. Из-за этого иногда вы можете кликнуть не туда и записать не то действие. Зачастую по внешнему виду сложно сказать, что из наложенных друг на друга элементов – контейнер, а что элемент управления. Нужно попробовать работать сначала с одним, и в случае ошибки с другим.

Наведите курсор на внутренний прямоугольник поля редактирования. Клик записан. Теперь введите текст для поиска.



В приведенном выше окне можно ввести любой текст, на английском или русском языке. Если для того, чтобы ввести очередной символ, нужно переключить раскладку клавиатуры, робот сделает это самостоятельно. Здесь же можно проэмулировать ввод клавиш или комбинаций клавиш.

Например, после введенного текста нужно нажать какую-то комбинацию клавиш, допустим, Ctrl+Shift+S. Для этого дважды кликните по строке, где указано +Ctrl («+» означает нажатие на кнопку).



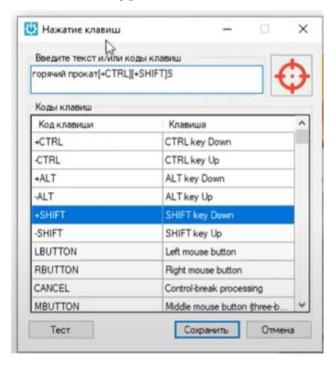
Потом дважды кликните на кнопку +Shift



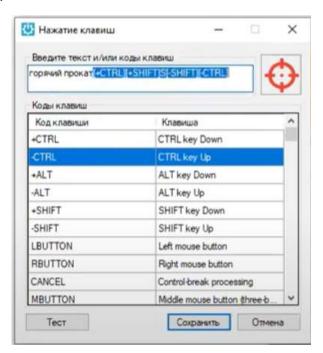


Это значит, что вы нажали на Shift, но Ctrl ещё не отпустили, т.е. обе клавиши нажаты вместе.

Потом просто введите с клавиатуры S.



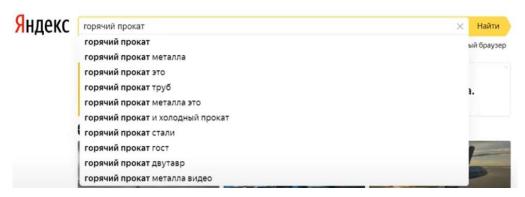
Потом нужно отпустить нажатые клавиши. Для этого выберите пункты «-Shift» и «-Ctrl».



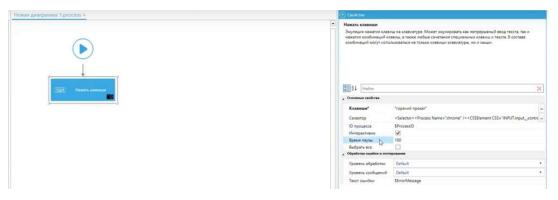


Такой код, записанный в текстовом поле, позволит проэмулировать нажатие комбинации клавиш Ctrl+Shift+S.

Нажмите на кнопку «Тест».



В поисковой строке Яндекса вводится ваш запрос. Нажмите на кнопку Сохранить, а потом на кнопку Сохранить и закрыть на панели записи. Результат сохраняется в виде блока.



Рассмотрим свойства блока «Нажать клавиши».

Свойство «Интерактивно» (галочка) указывает, будет ли использоваться способ ввода, при котором весь текст вставляется целиком, или после нажатия на каждую кнопку будет эмулироваться некая пауза. Часто бывает такое, что при снятой галочке скрипты, выполняющиеся на странице, мешают вводу текста. Поэтому галочку лучше проставлять, хотя без неё быстрее. Если галочка стоит, можно указать время паузы между нажатиями клавиш в миллисекундах.

«Выбрать всё» – довольно полезное свойство. Если проставить эту галочку, то робот перед вводом текста проэмулирует нажатие клавиш Ctrl+A, чтобы выделить весь текст, уже введенный в поле редактирования. Это полезно, если вы не уверены, что в поле, куда вы будете вводить текст, уже что-нибудь не напечатано.

Итак, после ввода нашего текста для поиска необходимо нажать кнопку Найти. Нажмём кнопку записи. Обратите внимание, при наведении мыши на 170



кнопку Найти происходит ситуация, аналогичная описанной ранее – есть внутренняя часть кнопки, и есть внешняя часть кнопки, и они друг в друга вложены.



Запишите клик на кнопку с помощью нажатия Ctrl.

Просмотрите появившийся в дизайнере блок и его селектор. Проверьте работу селектора. Селектор не работает. После ввода текста в строку поиска ничего не происходит, т.е. кнопка «Найти» не нажимается.

Как говорилось ранее, при наведении курсора на кнопку Найти прямоугольная область кнопки становится чуть темнее. Это значит, что на ней «висит» некий скрипт, что и меняет селектор кнопки. Это легко увидеть в окне редактирования селектора.



Хоть редактирование селекторов для браузеров осуществляется в том же окне, что и для приложений, внешний вид таких селекторов отличается. В них обычно всего две или три строки, причем верхняя строка всё так же обозначает процесс, в данном случае Chrome, а нижняя – тот селектор, с помощью которого мы будем искать нужный элемент управления страницей. Здесь используется нотация CSS-селекторов, она не такая сложная, как может показаться на первый взгляд.

Есть набор либо имён тегов (они пишутся большими буквами), либо имён классов (они пишутся маленькими буквами), и они соединены друг с другом точками. Точка здесь имеет значение «и». Найдите элемент управления, у которого тег BUTTON, а в свойствах перечислены все те свойства, что указаны в селекторе:





В данном селекторе последняя часть вызывает сомнения:



Свойство hovered_yes означает, что над кнопкой находится курсор мыши. Оно означает, что стиль кнопки меняется из-за того, что над кнопкой находится курсор. Для корректировки этого селектора нужно просто убрать данное свойство из записи в селекторе вместе с точкой. Остальное можно оставить.

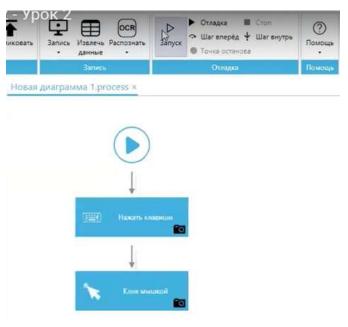


Проверьте работу селектора с помощью кнопки «Подсветить».

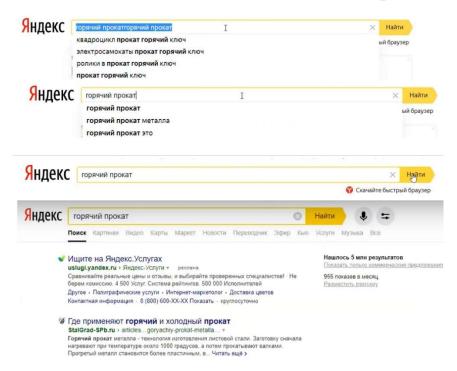


Робот фокусирует рамку вокруг нужного элемента.

Проверьте работу селектора на практике. В дизайнере нажмите кнопку «Запуск».







Робот выполнил свой сценарий корректно.

Благодаря тому, что в свойствах проставлена галочка «Выбрать всё», робот сначала удаляет текст, который уже был в элементе управления.

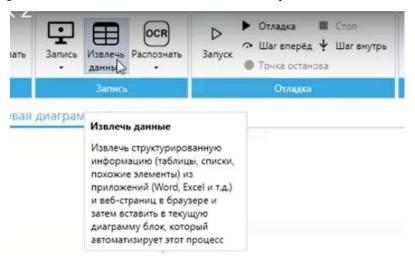
Итак, вы вышли на страницу результатов, и здесь следующее логичное действие – спарсить все результаты поиска, т.е. нужно получить все результаты в виде списка – прокликать все результаты и получить их описания. Это можно сделать как минимум пятью разными способами.

- 1. Существуют блоки невидимого браузера, которые позволяют скачать всю страницу с результатами и извлекать из html-кода нужные результаты, но существуют способы проще.
 - 1) Для работы с невидимыми браузерами есть блок Yandex поиск, который под ключ реализует задачу поиска ключевых слов и получения результатов поиска с заданного количества страниц. Также он поддерживает автоматическое распознавание Captcha Яндекса.
 - 2) При визуальной работе с браузером можно использовать блок «Посчитать элементы», с помощью которого можно посчитать, какое количество однотипных элементов находится на странице.
 - 3) Можно получить текст, воспользовавшись подстановочными переменными, чтобы получить определённое количество однотипных элементов со



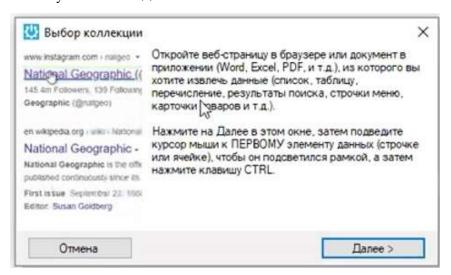
страницы. Но тогда нужно разбираться в том, как работает CSS—селектор, чтобы понять, куда правильно вставить подстановочные переменные.

4) Ещё один простой способ - использовать кнопку «Извлечь данные».



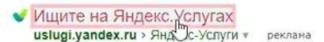
Эта кнопка позволяет извлекать структурированную информацию из веб — сайтов. Структурированная информация – списки, таблицы и просто набор похожих элементов на странице. Эта же кнопка может работать аналогичным образом не только в браузере, но и в некоторых приложениях. С её помощью можно извлечь таблицу из Excel, Word, и даже из интерфейса 1С.

Нажмите кнопку «Извлечь данные».



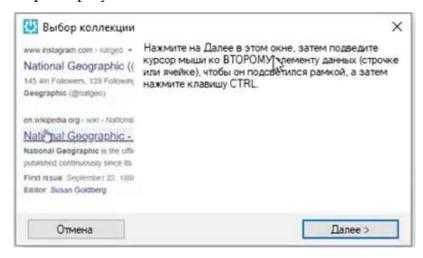
Окно подсказки сообщает, что после нажатия на кнопку Далее необходимо указать один или два образца тех элементов, которые требуется извлечь. Одного образца обычно достаточно для извлечения данных из таблиц, которые были отформатированы как таблицы, и два образца предоставляются для извлечения данных из списков результатов поиска. Нажимите на кнопку «Далее».





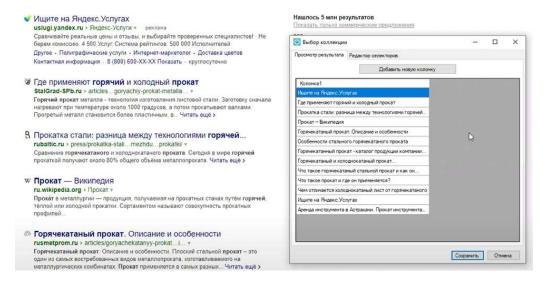
Сравнивайте реальные цены и отзывы, и выбирайте проверенных специалистов! - Не берем комиссию. 4 500 Услуг. Система рейтингов. 500 000 Исполнителей Другое - Полиграфические услуги - Интернет-маркетолог - Доставка цветов Контактная информация - 8 (800) 600-XX-XX Показать - круглосуточно

Подсветите первый результат и нажмите Ctrl.



Следующее окно-подсказка рекомендует нажать «Далее» в этом же окне, затем подвести курсор ко ВТОРОМУ элементу данных (строке или ячейке) для его подсветки, и нажать Ctrl. Последуйте данной рекомендации.

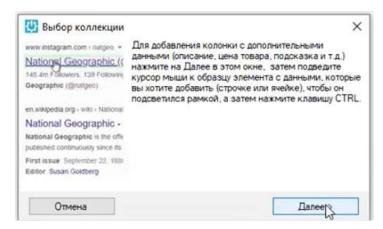
Выберите второй элемент на странице результатов. Обратите внимание, что здесь много вложенных контейнеров и при клике необходимо выбрать правильный элемент.



В окне представлены извлечённые со страницы результатов все 10 результатов поиска на основе двух отобранных нами образцов. Эти результаты можно автоматически сохранить в таблицу данных.

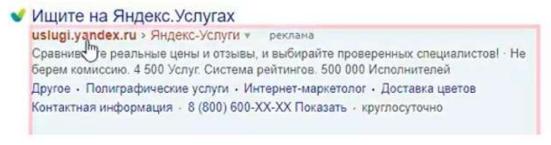


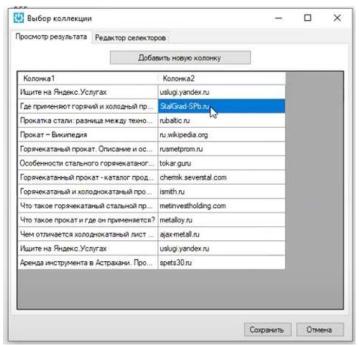
Если нужно извлечь из списка результатов не только заголовки, но и, допустим, описание, адреса и пр., можно использовать кнопку «Добавить новую колонку».



При нажатии кнопки система рекомендует нажать кнопку «Далее», подвести курсор мыши к образцу элемента с данными (строка или ячейка), которые нужно добавить, для подсветки рамкой данного элемента, и затем нажать Ctrl.

Попробуйте извлечь доменные имена.

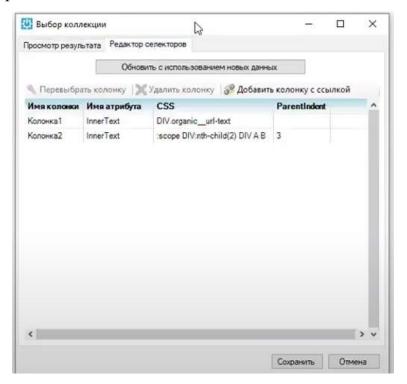




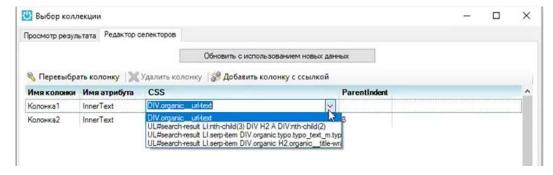


Появилась вторая колонка, где по нашему образцу робот собрал все результаты. Обратите внимание, что результаты из первой и второй колонки сопоставлены правильно. Т.е. робот понимает структурную связь между заголовком каждого результата и доменным именем.

Если требуется прокликать по результатам поиска и получить URL—адреса страниц, то для этого можно воспользоваться специальной функцией на вкладке редактора селектора.



Здесь представлены селекторы, по которым были извлечены первая и вторая колонки. Если что-то почему-то не срабатывает, то вы всегда можете изменить вручную данные селекторы. Кроме того, робот, как правило, предоставляет на выбор несколько разных вариантов селектора для выбора того или иного элемента:



Правильными могут быть несколько вариантов одновременно. Выбирая правильный вариант, необходимо учитывать, что в нём не должно быть

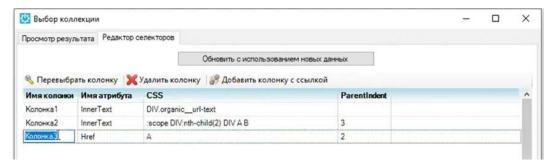


изменяющихся частей. В представленных вариантах такого нет. Но если вы увидите в селекторе странное сочетание букв, символов и цифр, — это скорее всего какой-то ID, меняющийся в реальном времени. Т.е. каждый раз, когда вы будете заходить на страницу и обновлять её, данное сочетание символов будет меняться. Это значит, что робот не сможет по данному селектору в любой момент времени найти нужный элемент управления. Также сомнительным является сочетание нескольких цифр (более трёх). Это, как правило, тоже ID, а не фиксированное значение.

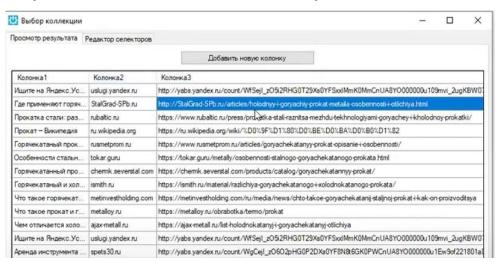
В данном случае селектор сработал сразу, поэтому оставляем исходное значение для обеих колонок.

Если бы вы изменили селектор, можно было бы нажать на кнопку «Обновить с использованием новых данных». Тогда на первой вкладке вы бы увидели обновлённые результаты. Кроме того, можно перевыбрать колонку из ранее записанных или поработать со списком селектора.

Для того, чтобы добавить ещё одну колонку со ссылками, необходимо нажать соответствующую кнопку.



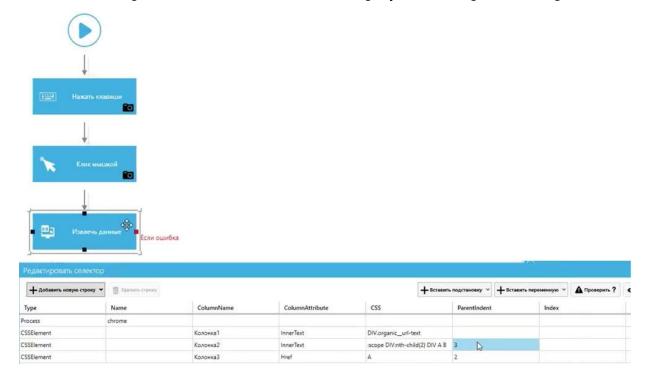
Робот посчитал, что селектора «А» будет достаточно, и мы сможем получить ссылку на этот элемент. Нажмите на кнопку «Обновить».





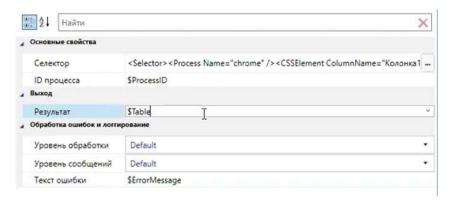
В результате получен новый столбец с URL-адресами страниц. Нажмите Сохранить.

В диаграмме появился предварительно настроенный блок «Извлечь данные», в котором всё готово для извлечения результатов в реальном времени.



Как проверить работу этого блока.

Данный блок отправляет результаты в переменную, указанную в выходном свойстве результата.



Можно указать здесь любую переменную, но принято сначала объявлять переменную, с которой мы далее будем работать. Особенно если это составная переменная.

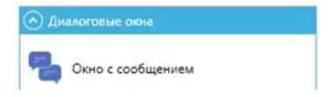
Объявите переменную Table и укажем её тип - таблицы данных.





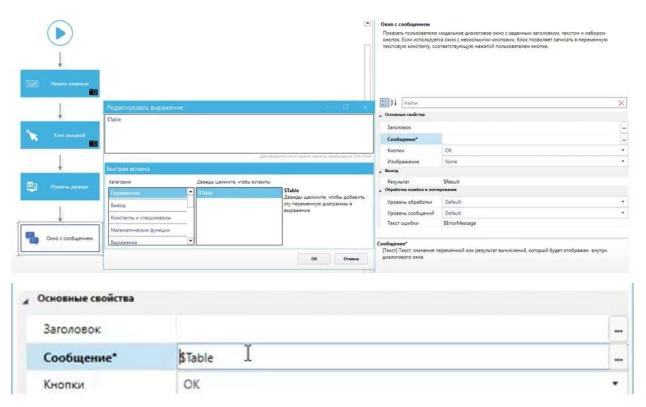
Этого можно не делать. Если вы не создаёте переменную, но она используется как выходной параметр в каком-то свойстве, она всё равно будет создана автоматически. Удобство явного объявления переменных на панели переменных в том, что вы в режиме отладки на любом шаге сможем посмотреть текущее содержимое этой таблицы.

Выведите на экран содержимое данной таблицы. Для этого используйте блок «Окно с сообщением» из группы «Диалоговые окна».



Этот блок поддерживает вывод и простых, и сложных типов данных.

В свойствах блока выберем нашу переменную Table. Из неё будет взят текст сообщения, выводимого на экран



Вернитесь на страницу Яндекса и запустим весь сценарий.



После поиска результатов по запросу появилось окно, в котором вы видите содержимое всех трёх колонок. Данные не разделены, но тем не менее здесь записана вся извлеченная информация.



Далее можно использовать содержимое этой таблицы в циклах, в переборе по строкам – используя любые другие инструменты, которые платформа предлагает для работы с таблицами данных.

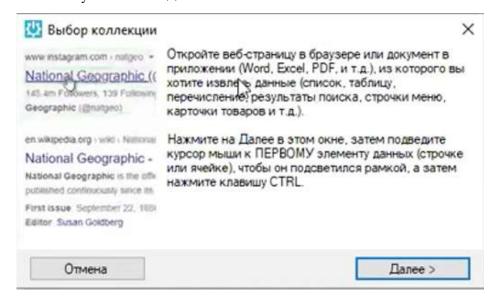
3.3.8.5 Извлечения данных из таблиц в Word и Excel

Pассмотрим работу кнопки «Извлечь данные» на примере приложений Word или Excel.

Откройте Word, создайте таблицу и введите какие-нибудь данные в ячейки.

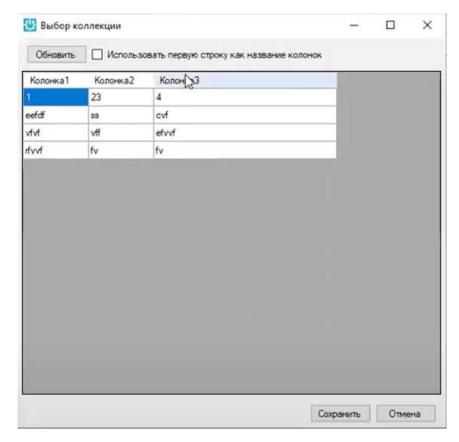
1	23	4
eefdf	ss	cvf
vfvf	vff	efvvf
rfvvf	fv	fv

Нажмите кнопку «Извлечь данные».

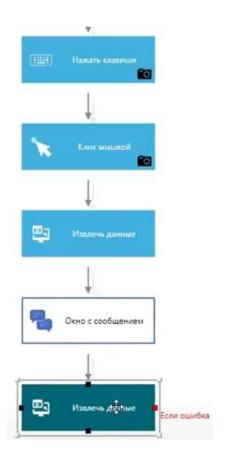


Предлагается задать образец извлекаемых данных. В данном случае достаточно указать на ячейку таблицы и нажать Ctrl.





Робот извлёк все данные из таблицы. Обратите внимание: при нажатии на кнопку «Извлечь данные» при работе с браузером, блок в диаграмму добавляется голубого цвета. А при работе с приложением блок темно-синего цвета.

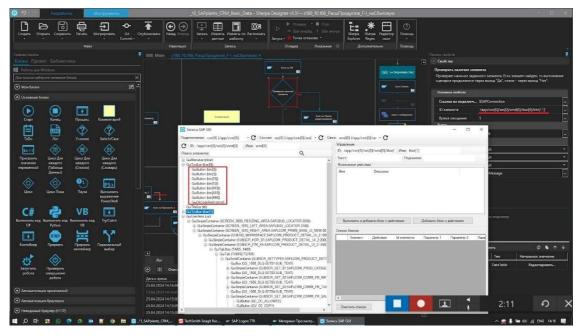




Робот понимает, что работа осуществлялась с разными типами приложений и разными методами извлечения данных, и берет блок из нужной палитры данных. Напомним, что такие блоки есть в палитре для работы с десктопными приложениями и для работы с браузерами.

3.4 Sherpa Designer: ответы на часто задаваемые вопросы

Вопрос: в проекте робот проверяет наличие кнопки «Сохранить» в SAP. Если не находит – то кликает по элементу «Изменить». Почему блок "Проверить наличие элемента" находит элемент «Сохранить» даже если его нет?



Ответ: скорее всего, этот элемент находится в кеше СОМ-компонента SAP. Попробуйте перед проверкой переподключиться к SAP. Например, с помощью блоков «Присоединиться» или «Выбор окна».

Вопрос: есть ли возможность автоматизации 1С с помощью горячих клавиш?

Ответ: да, есть. Вы можете воспользоваться шпаргалкой с официального сайта 1C: https://its.1c.ru/db/v8std/content/430/hdoc

Или подсказкой для назначения горячих клавиш по ссылке:

https://1s-portal.ru/index.php/stati/configuration/3-form/76-knopke-naznachit-gorjachuju-

klavishu.html#:~:text=%D0%9D%D0%B0%D0%B7%D0%BD%D0%B0%D1%87%D0%B5
%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BB%D0%B0%D0%B2%D0%B8%D1%88
%20%D0%BC%D0%BE%D0%B6%D0%BD%D0%BE%20%D0%B4%D0%B5%D0%BB%D



0%B0%D1%82%D1%8C%20%D0%BA%D0%B0%D0%BA,%D0%BA%D0%BB%D0%B0% D0%B2%D0%B8%D1%88%D0%B8%2C%20%D1%81%D1%80%D0%B0%D0%B7%D1%8 3%20%D0%B6%D0%B5%20%D1%83%D0%B2%D0%B8%D0%B4%D0%B5%D1%82%D0 %B5%20%D1%80%D0%B5%D0%B7%D1%83%D0%BB%D1%8C%D1%82%D0%B0%D1 %82

Bonpoc: почему логика блока выполнить PowerShell работает иначе в сравнении с встроенным PowerShell? И как его в таком случае использовать корректно. Пытаюсь использовать функцию – корректный вывод должен быть False. При вызове этой функции через PowerShell блок возвращается True (во втором блоке).

```
> $Parameters = @{"Kog EE" = "01111"}
$Row = @{"010XX"}

> function Check-Code {
    param {
        [string]$code,
        [string]$template
    }

# Исключения
    if ($code -eq "01004" -or $code -eq "01000") {
        return $true
    }

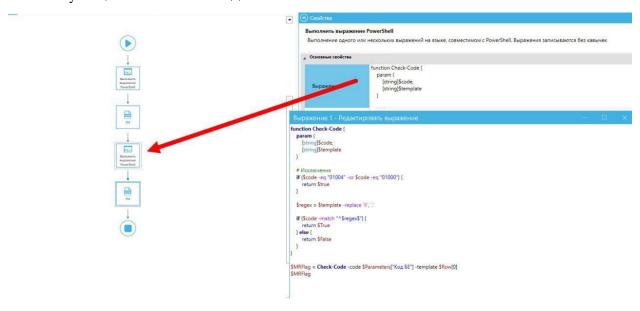
$regex = $template -replace 'X', '.'

if ($code -match "^$regex$") {
        return $True
    } else {
        return $False
    }
}

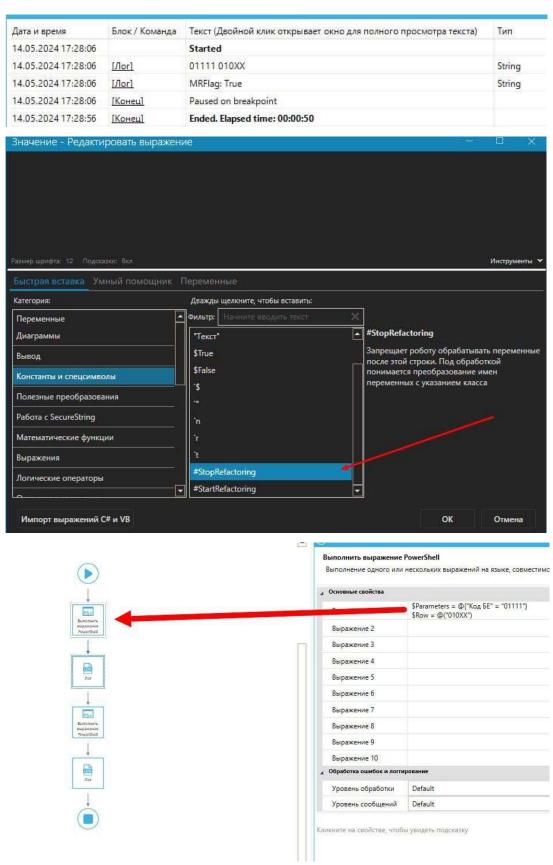
$MRFlag = Check-Code -code $Parameters["Kog EE"] -template $Row[0]
$MRFlag

False
```

Функция объявляется здесь:







Ответ: используйте для таких целей свои блоки. Важно понимать, что все, что Вы пишите выполняется внутри функции, так как каждый блок — это функция. Как итог все, что Вы присвоите в этой функции, будет ее локальными



переменными. Если Вы используете свой код, то должны отключить обработку переменных. Сейчас все переменные, что Вы написали преобразуются к локальным переменным скрипта, а именно этого класса диаграммы.

Вопрос: что делать, если возникла проблема при использовании блока «Выполнение кода Python»?

Ответ: порядок действий, следующий:

- 1) Закройте Дизайнер.
- 2) Удалите из папки проекта папку Library.
- 3) Скачайте и обновите Дизайнер.
- 4) Убедитесь, что у Вас стоит версия питона х64.
- 5) Проверьте, чтобы версия питона была не выше 3.12.

4 Sherpa Orchestrator

Sherpa Orchestrator (Оркестратор) — это инструмент для централизованного подключения и управления программными роботами, созданными на платформе Sherpa RPA. Оркестратор управляет роботами, определяет их состояние, назначает разным роботам разные задачи, а также, организует очереди выполнения задач и контролирует их исполнение.

Каждый программный робот является самостоятельным продуктом, а для управления парком роботов можно использовать единую панель управления — Sherpa Orchestrator. Для корректной работы программных роботов использование Sherpa Orchestrator не обязательно. Но пользоваться Оркестратором можно даже при наличии одного робота.

Sherpa Orchestrator предоставляет пользователю следующие возможности:

- 1) Управление роботами:
- определение состояния роботов;
- назначение задач роботам;
- версионирование сценариев роботов;
- управление очередями роботов;



- сводный отчет по исполнению задач;
- вывод наглядных графиков по исполнению задач;
- составление расписаний для роботов.

2) Администрирование:

- администрирование разрешений и доступов для роботов;
- управление лицензиями роботов;
- удаленная активация лицензий роботов;
- установка обновлений;
- управление правами доступа пользователей на основе ролевой модели;
- централизованное хранение паролей от учетных записей;
- централизованное хранение общих доступных роботу данных с ограничением уровня доступа;
- 3) Ведение журнала роботов, в том числе, полное логирование действий робота, включая скриншоты состояний экрана.
 - 4) Аудит безопасности Оркестратора.
- 5) Управление роботами, работами, задачами, расписаниями и другими объектами Оркестратора через API.

Возможности экспорта данных

На всех экранах Оркестратора, содержащих табличные представления, доступны функции экспорта в CSV и XLSX. Во всех таблицах при наведении курсора мыши на заголовок колонки таблицы появляется пиктограмма вызова операций с таблицей со следующими функциями: авторазмер колонок, выбор отображаемых колонок, фильтр по значениям колонок, полнотекстовый поиск по всем или по одной конкретной колонке.

Системные требования к серверу для установки Sherpa Orchestrator

Системные требования до 100	Системные требования от 100 до 1000
лицензий	лицензий
Процессор: 3GHz x 2 ядра	Процессор: 3GHz x 8 ядер



Оперативная память: 8 GB	Оперативная память: 16 GB
SSD: 100 GB свободного места	SSD: 100 GB свободного места
CentOS / Ubuntu / Debian / RedOS /	CentOS / Ubuntu / Debian / RedOS /
Astra Linux и т.д.*	Astra Linux и т.д.*
Docker 22+**	Docker 22+**
Порты: 80, 443 (с возможностью	Порты: 80, 443 (с возможностью
пробросить в контейнер при	пробросить в контейнер при
использовании установки с	использовании установки с помощью
помощью Docker-контейнера)	Docker-контейнера)
Доступ к серверу через протокол	Доступ к серверу через протокол SSH /
SSH / sFTP на время установки,	sFTP на время установки, настройки и
настройки и отладки	отладки

* Для уточнение минимальной версии конкретного дистрибутива обратитесь в техническую поддержку. Возможно использование на виртуальном сервере, VPS на базе средств виртуализации, поддерживающих Docker.

**Рекомендуется установка с помощью Docker-контейнера, однако возможна установка и без Docker-контейнера, непосредственно на операционную систему. Возможно использование на виртуальном сервере или VPS на базе средств виртуализации, поддерживающих Docker.

Лицензирование Оркестратора

Одна лицензия на Sherpa Orchestrator запускается на одном сервере в одном экземпляре и поддерживает управление неограниченным количеством роботов.

4.1 Развертывание платформы под управлением Оркестратора

Диаграмма развертывания платформы Sherpa RPA под управлением Оркестратора (см. раздел «1.3 Диаграмма развертывания Sherpa RPA») показывает варианты взаимодействия Оркестратора с другими компонентами платформы Sherpa RPA:



- RDP сервер с учетными записями Unattended-пользователей Sherpa Coordinator Service + Sherpa Unattended Robot
- Pабочее место разработчика (Windows) Sherpa Designer + Sherpa Robot
 - Клиентское рабочее место (Windows) Attended Robot
 - Рабочее место с браузером.

Последовательность процесса

- 1) Разработчик сценария роботизации бизнес-процесса с помощью десктопного ПО Sherpa Designer и локального Sherpa Robot создаёт и отлаживает сценарий роботизации. Готовые сценарии роботизации передаются на рабочие места пользователей или роботов вручную путём копирования конфигурационного файла робота либо с помощью функции удаленной публикации новой версии сценария из Sherpa Designer в Sherpa Orchestrator.
- 2) Серверный компонент Sherpa Orchestrator поддерживает связь с роботами, запущенными на клиентских машинах, хранит конфигурации роботов и версии сценариев, общие глобальные переменные и учетные данные, логи и скриншоты работы роботов, журналы аудита оркестратора, пользователей, роли и тенанты самого Оркестратора, лицензии всех компонентов платформы и статистику исполнения сценариев.
- 3) Пользователи и администраторы Sherpa Orchestrator получают доступ к ресурсам, настройкам и статистике с помощью веб-приложения оркестратора, доступного через веб-браузер. Sherpa Orchestrator включает в себя веб-сервер Nginx, интерпретатор PHP и реляционную базу данных (по умолчанию MariaDB, опция Postgre).
- 4) Локальные пользователи по необходимости запускают Sherpa Attended Robot на исполнение хранящихся локально сценариев. В процессе исполнения сценария Sherpa Attended Robot может передавать Оркестратору текущий статус, задачи, логи, значения глобальных переменных и учетных данных или получать от него задачи, значения глобальных переменных и учетных данных.



5) По расписанию, вызову API, команде робота либо по другому поддерживаемому триггеру Sherpa Orchestrator даёт задание Sherpa Coordinator Service соответствующего терминального сервера создать RDP-подключение к локальной или удаленной учетной записи, выделенной для Unattended-робота, при этом на одном терминальном сервере может быть размещено и одновременно активно несколько таких учетных записей.

Вход в учётную запись Unattended-робота производится с предоставленными Оркестратором логином и паролем.

После входа в учетной записи запускается соответствующий экземпляр Sherpa Unattended Robot, подключается к Оркестратору, получает задание на выполнение сценария и сам сценарий, хранящийся в оркестраторе.

В процессе исполнения сценария Sherpa Unattended Robot может передавать оркестратору текущий статус, задачи, логи, значения глобальных переменных и учетных данных или получать от него задачи, значения глобальных переменных и учетных данных, команды для «мягкого» или «жесткого» завершения сценария.

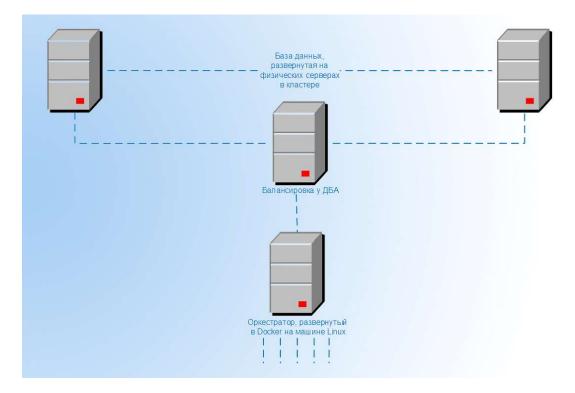
После завершения исполнения сценария Sherpa Unattended Robot выполняет logoff из своей учетной записи.

В простых бизнес-процессах использование Оркестратора не обязательно, так как Attended версии Роботов могут использоваться автономно. Потенциал Оркестратора раскрывается в бизнес-процессах, которые требуют использования нескольких роботов, а также организации централизованного управления и взаимодействия между ними.

4.1.1 Отказоустойчивое развертывание Оркестратора

Оркестратор поддерживает отказоустойчивое развертывание, что продемонстрировано на схеме ниже. Для этого он использует две синхронизирующиеся базы данных и балансировку.





Для реализации этой схемы развертывания понадобятся две дополнительные виртуальные машины с администрированием баз данных, и осуществить все системные настройки баз данных (БД), а также три IP из одной подсети (два – для серверов баз данных и один свободный). Затем необходимо в docker-compose.yml прописать следующие настройки:

- 1) Для контейнера orchestrator в environment исправить DB_HOST
 - "DB HOST= ір адрес хоста балансировки БД"
- 2) Для контейнера orchestrator закомментировать следующие строки:
- # depends on:
- # orchestrator-db

Прописать адрес балансировки БД в конфигах:

/opt/SherpaOrchestrator/backend/config/config.ini

И

/opt/SherpaOrchestrator/backend/config/phinx.php



- 4.2 Установка Sherpa Orchestrator
- 4.2.1 Рекомендации по размещению

Использование Оркестратора на терминальном сервере

В Attended-сценариях Работы, при использовании Робота с интерактивным входом в учетную запись Пользователя, если внутри сценария есть имитация действий Пользователя в графическом интерфейсе десктопа или браузеров, эти действия будут работать, только если существует активное подключение к рабочему столу этой виртуальной машины (на экран смотрит какой-либо Пользователь). Как только Пользователь отключился от рабочего стола (RDP), Windows уничтожает графический контекст. Получается, что учетная запись продолжает работать. Робот также продолжает работать, но никаких кнопок и окон на экране уже не существует, то есть программно они перестают быть доступны. Поэтому чтобы без подключенных к учетным записям Пользователей можно было автоматизировать работу с приложениями путём имитации действий пользователя, к данной учетной записи должно быть постоянно установлено RDP-соединение.

Для обеспечения Unattended-сценариев Работы служит компонент Sherpa Coordinator. По команде оркестратора Координатор поднимает RDP-подключения к нужным учётным записям и удерживает их, пока работает сценарий, чтобы графический контекст Windows продолжал существовать, и все функции Робота работали в полном объеме. После окончания Работы сценария Робот самостоятельно выходит из учетной записи пользователя.

Использование Оркестратора без терминальных серверов

Можно организовать работу Оркестратора без терминальных серверов. В этом случае нужно выделить одну из виртуальных машин под работу Координатора. Установленный Координатор с этой выделенной виртуальной машины будет устанавливать RDP-подключения к другим виртуальным машинам, на которых уже будут запущены конкретные Unattended-роботы.



Таким образом, нет ограничения в том, что Координатор может подключаться по RDP только к своим собственным учетным записям на том же сервере. Он может подключаться к любым учетным записям на любых серверах, в том числе к виртуальным машинам в рамках сетевой связности.

Известные ограничения

Оркестратор позволяет завести любое количество Координаторов и любое количество сессий в рамках одного Координатора. Ограничение в использовании большого количества Координаторов и сессий может возникнуть только из-за ограничений аппаратных ресурсов и ограничений со стороны операционной системы в отношение того, какое количество одновременных RDP сессий к учётным записям Unattended-роботов может быть запущено и удерживаться одновременно из одного терминального сервера или виртуальной машины.

4.2.2 Установка

Оркестратор представляет собой веб-приложение, предназначенное для работы в среде Linux. Он совместим с большинством популярных дистрибутивов Linux: Ubuntu, Debian, RedHat, Cent OS, AstraLinux и т.д.

Варианты установки:

- внутри контура компании (On Premises);
- развертывание в облаке.

Для установки в контуре компании потребуется выпустить SSL-сертификат на локальное доменное имя Оркестратора, который будет обеспечивать безопасное соединение между Роботами, Пользователями и Оркестратором.

Обратите внимание! SSL-сертификат должен быть создан на основе корневого (root authority) сертификата компании, а сам корневой сертификат должен быть установлен и доступен в локальных хранилищах сертификатов на конечных рабочих местах пользователей и роботов.



Примечание. Для получения подробной инструкции по выпуску SSL-сертификата для Оркестратора обратитесь к администратору SSL-сертификатов вашей компании.

Установка Оркестратора в контуре компании

• с помощью Docker-контейнера

Установку оркестратора рекомендуется производить посредством развертывания Docker-контейнера (требуется Docker 22+). Состав команд может меняться в зависимости от используемого дистрибутива Linux и системного окружения.

Для получения актуальной версии инструкции по установке необходимо обратиться в техническую поддержку (support@sherparpa.ru).

без Docker-контейнера

При необходимости оркестратор можно установить без использования Docker-контейнера. Состав команд может меняться в зависимости от используемого дистрибутива Linux и системного окружения. Инструкции по установке приведены на примере дистрибутива «Debian 11 Minimal».

Для получения актуальной версии инструкции по установке Оркестратора непосредственно на хост-систему или для получения рекомендаций по установке оркестратора на другие дистрибутивы необходимо обратиться в техническую поддержку (support@sherparpa.ru).

Использование облачной версии

При использовании облачной версии доступ к Оркестратору осуществляется по ссылке, предоставленной поставщиком решения. Вход производится по логину и паролю.



4.3 Начало работы

4.3.1 Использование Attended-роботов в Оркестраторе

Attended-роботы — это роботы, которые запускаются на компьютере пользователя вручную, из автозагрузки, по комбинации клавиш или по локальному расписанию. Лицензии Sherpa Attended Robot являются именными. Одна лицензия запускается на одной учетной записи пользователя, лицензируется на учетную запись конкретного пользователя.

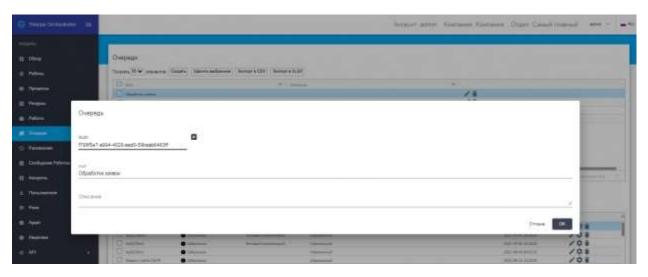
Attended-роботы могут быть добавлены в Оркестратор, но данный тип роботов не может запускаться из Оркестратора с помощью создания работ и расписания. Таким образом, сценарии Attended-роботов запускаются только локально.

Attended-роботов целесообразно заводить в Оркестратор в тех случаях, когда у вас есть Unattended-робот на потоковой обработке каких-либо документов или заявок и требуется использовать Attended-робота в режиме станции верификации.

Unattended-робот при этом обрабатывает все заявки, которые не требуют взаимодействия с человеком, а Attended-робот в режиме станции верификации может использоваться в процессе обработки заявки на каком-либо этапе, когда необходимо подключить человека для принятия определенного решения (то есть верифицировать вручную). Такой Attended-робот (верификатор) обращается к Очереди и берет из нее задачу, которая была размещена для него Unattended-роботом.

Attended-Роботы имеют доступ к очередям. Они могут получать задачи из очередей, брать эти задачи в обработку и менять статус задач. Их можно завести в Оркестратор, но у них ограниченный список возможностей в Оркестраторе.





С этой задачей может быть связан какой-либо набор данных, потому что любая задача содержит словарь пара «Ключ=Значение». Этих пар может быть сколько угодно, и можно передавать такие данные между роботами с помощью очередей внутри задач.

4.3.2 Использование Unattended-роботов

Unattended-роботы — это роботы, которые запускаются без участия человека и управляются только через Оркестратор, запускаются по расписанию, через очереди задач или через АРІ. Лицензии Sherpa Unattended-Robot являются конкурентными. Одна лицензия позволяет под управлением Оркестратора запустить выполнение сценария на учетной записи любого пользователя (в том числе виртуального, созданного специально для робота). Общее число используемых в один момент времени лицензий не может превышать число зарегистрированных в Оркестраторе Лицензий.

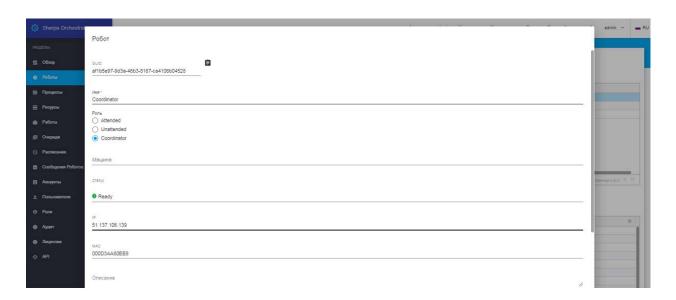
Unattended-робота можно запускать вручную или через автозагрузку или планировщик Windows, и работать с ним так же, как и с Attended-роботом.

Варианты использования Unattended-робота:

• Использование Unattended-робота с ручным запуском сессии пользователя. Если Оркестратор увидит, что Робот находится в сети (зеленый индикатор), Оркестратор сможет выдавать ему работу. В этом случае запуск учетной записи Робота должен быть произведен вручную.



• Использования Unattended-робота с автоматическим запуском сессии пользователя. При классической схеме, когда есть RDP-сервер. На сервере есть одна или несколько учетных записей, зарегистрированных за роботами. Сам по себе сервер работает постоянно, но учетные записи по умолчанию не работают. Соответственно, если на комплектного робота Оркестратор выделил задание, он этого робота должен включить, а для этого он должен активировать соответствующую учетную запись на сервере. Для этого предназначен компонент, называемый «Координатор». Это отдельная программа, которая не требует лицензии, устанавливается под учетной записью администратора RDP-сервера и запускается в режиме сервиса.



4.3.3 Координатор Unattended-роботов

Координатор — это сервис, который должен быть установлен под учетной записью администратора RDP-сервера (терминального сервера), на котором планируется в пользовательских учетных записях запускать Unattended-роботов, при условии, что требуется автоматический вход в учетные записи роботов. Координатор служит для запуска сессий локальных пользователей вместе с Unattended-роботами на терминальных серверах под управлением операционных систем Windows 2008 — 2022.



Задача Координатора: по команде оркестратора поднять и удерживать RDP-подключение к локальной или удаленной пользовательской учётной записи, в которой затем запускается Unattended-робот.

Для Attended-роботов указывать Координатор не требуется.

Координатор устанавливается из соответствующего дистрибутива Sherpa Coordinator, предоставляемого вендором. Sherpa Coordinator не требует отдельной лицензии.

Робот с функцией Координатор (для Unattended-роботов) отображается в таблице «Роботы». Координатор всегда находится во включенном состоянии, о чем сигнализирует зеленый индикатор. Данный сервис обеспечивает создание и удержание сессий пользовательских учетных записей для Unattended-роботов в рамках используемого терминального сервера.

Все остальные роботы – это технические учетные записи, запущенные на том же терминальном сервере.

В случае если у вас конфигурация предусматривает терминальный сервер (Windows Server с ролями и лицензиями RDS), координатор надо установить изпод админа, а SherpaRPARuntime в режиме «Unattended» в каждую из учётных записей пользователей на этом терминальном сервере, которые будут работать как Unattended-роботы.

В случае если у вас не предусмотрены терминальные сервера, а есть только группировка виртуальных машин, одну из них нужно выделить под роль координатора. Эта виртуальная машина должна быть постоянно включена, т.к. она будет обеспечивать работу Unattended-роботов на других машинах. При этом на самой виртуальной машине согласно стандартной RDP-лицензии Windows вы сможете установить одного Unattended-робота в пользовательской учётной записи.

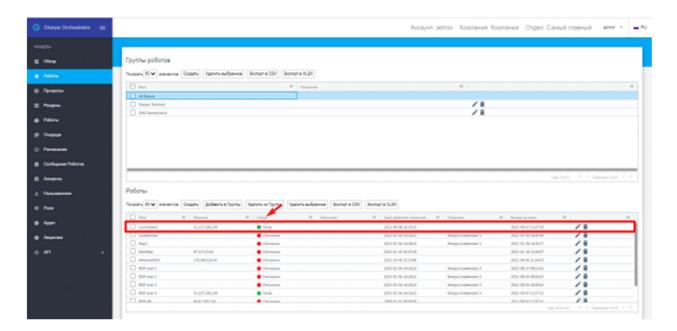


4.3.3.1 Принцип работы Координатора

По расписанию, вызову API, команде робота либо по другому поддерживаемому тригтеру Оркестратор даёт задание координатору соответствующего терминального сервера создать RDP-подключение к локальной или удаленной учетной записи, выделенной для Unattended-робота, при этом на одном терминальном сервере может быть размещено и одновременно активно несколько таких учетных записей. Вход в учётную запись Unattended-робота производится с предоставленными Оркестратором логином и паролем.

На стороне Оркестратора этот процесс выглядит следующим образом:

• После создания работы Оркестратор видит, что Unattended-робот недоступен, но ему назначен Координатор, и этот Координатор доступен, о чем сигнализирует зеленый индикатор.



- Оркестратор направляет Координатору команду на включение учетной записи этого Unattended-робота.
- Координатор принимает команду и в рамках сервера поднимает RDP-подключение своей собственной учетной записи, в которой находится этот робот. Т.е. он запускает RDP-подключение, учетная запись включается, внутри учетной записи запускается Unattended-робот, в списке его индикатор становится зеленым.





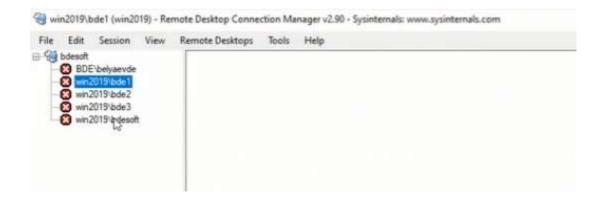
- Unattended-Робот обращается к Оркестратору, видит, что в работах для него есть задача, принимает ее и отрабатывает.
- Если запуск производится через такую цепочку действий, после выполнения задачи, робот автоматически выйдет из своей учетной записи, и учетная запись полностью выключится.

4.3.3.2 Установка Координатора

Робот в любом из способов установки (с дизайнером или без, в версии ехе или в версии msi) может иметь роль как Attended, так и Unattended, это определяется его лицензией (локальной или полученной из Оркестратора) и способом запуска (локально или через Оркестратор).

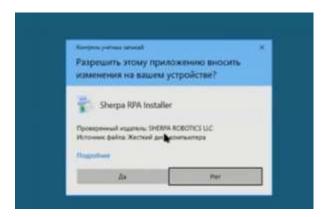
При установке Координатора необходимо выбрать тип устанавливаемого робота, на первом шаге установки.

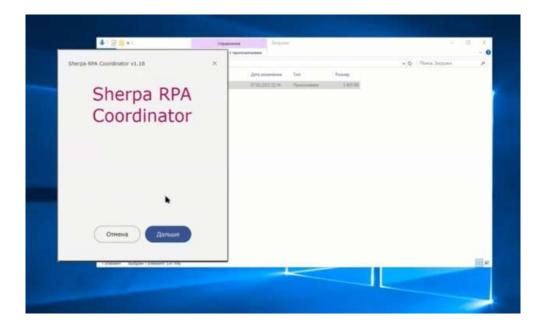
В примере на скриншоте ниже на виртуальной машине установлена ОС Windows Server 2019. На этой машине поднят виртуальный сервер, также созданы несколько учетных записей для входа под терминальным сервером.





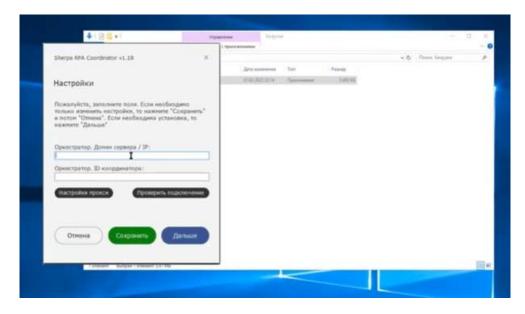
При входе на сервер под учетной записью администратора можно установить Координатор. Это служба, которая имеет два файла – саму службу и программу, которая управляет сессией.



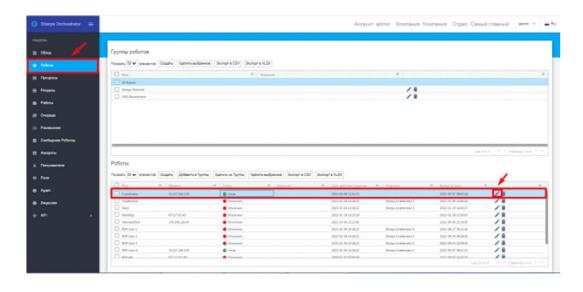


При установке координатора необходимо ввести домен сервера Оркестратора и идентификатор (ID) Координатора.





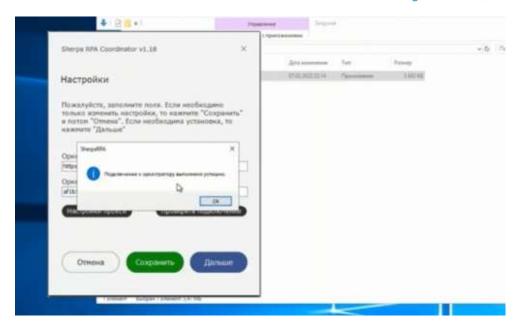
ID Координатора копируется в карточке «Робот».



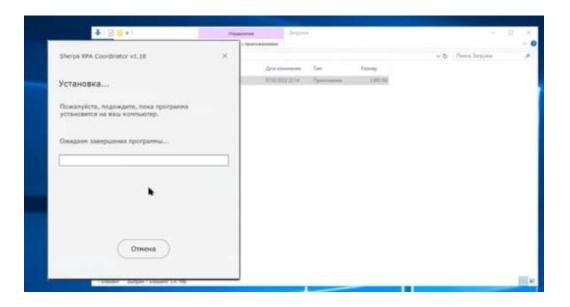


Далее необходимо проверить подключение.



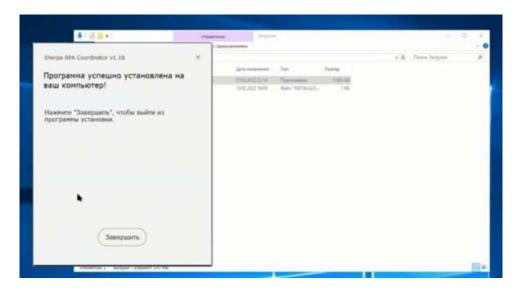


При успешном подключении производится запуск установки. Также при необходимости можно настроить прокси.

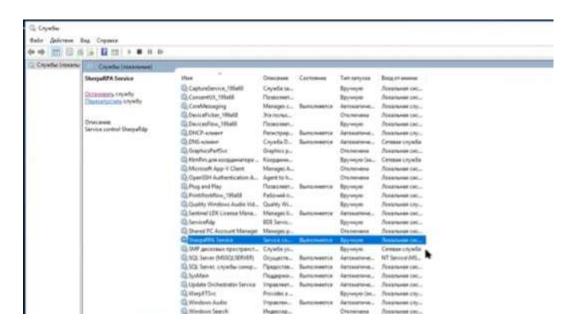


После завершения установки Координатора, запускается служба Sherpa RPA Service.



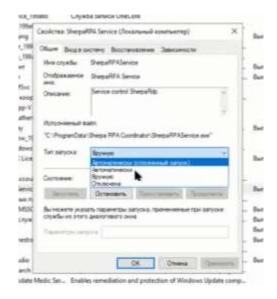


При проверке наличия службы Sherpa RPA Service в реестре служб, видно, что она установлена и находится в состоянии «Выполняется».



В настройках свойств Sherpa RPA Service предусмотрена возможность выбрать Тип запуска (вкладка «Общие»). Для удобства тип запуска можно установить «Автоматически».



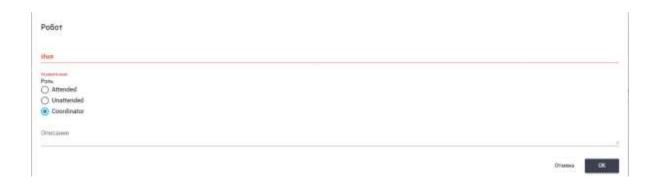


После установки Координатора дополнительная настройка не потребуется.

Далее нужно войти в учетную запись пользователя, в которой будет установлен робот, при этом используется установщик Runtime.

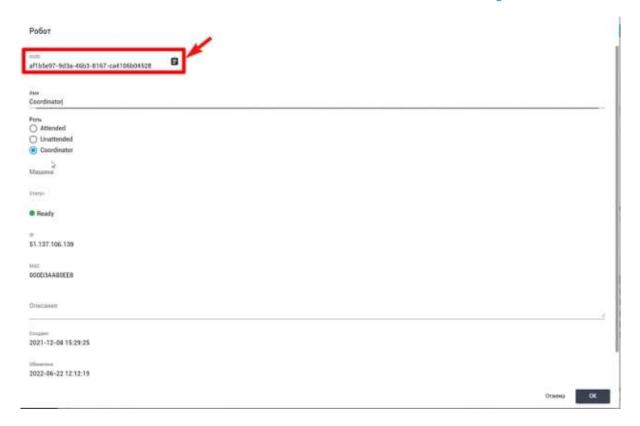
При создании Координатора необходимо указать имя (задается произвольно) и выбрать роль Coordinator.

Оркестратор, потребуется только его GUID (скопировать из Оркестратора).



После скопировать в карточке Координатора его GUID.





GUID нужно вставить в настройки Координатора, который был установлен под учетной записью Администратора на терминальный сервер или виртуальную машину. На одном из этапов установки программа запросит этот GUID.

Далее, необходимо завести Unattended робота:





Здесь нужно выбрать ранее созданного Координатора, ввести IP машины с Unattended-роботом, логин (с доменом, если есть) и пароль от учетной записи Пользователя, под которой должен заходить робот. Если есть active directory, то логин учетной записи должен указываться через доменное имя (через слэш).

Torow petrock samps
SHERPA\rdpuser1

В выпадающем списке необходимо выбрать тот Координатор, через которого будет осуществляться запуск Unattended-робота.



Также необходимо указать Лицензионные ключи, которые ранее были добавлены на экране Лицензии.





Далее выбрать Unattended-лицензию. Лицензия должна быть заранее заведена в Оркестратор и активирована (дата окончания лицензии должна быть больше текущего дня).

Скопированный GUID данного робота нужно вставить в настройки Unattended-робота который будет установлен уже под учетной записью Пользователя на терминальный сервер или на виртуальную машину.

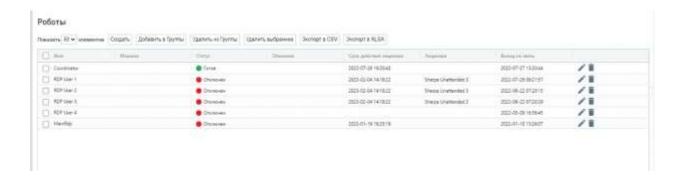
После установки и настройки Unattended-робота, нужно выйти из его учётной записи (logoff).

В карточке Робота необходимо обязательно заполнить IP-адрес или доменное имя машины, а также логин и пароль, с которыми Робот зайдет в свою учетную запись и сохранить изменения.

Далее, если сервер включен, и Координатор также включен, в списке возле него будет зеленый индикатор

Учетная запись Unattended-Робота выключена и в списке будет отмечена красным индикатором, как на изображении на скриншоте ниже.

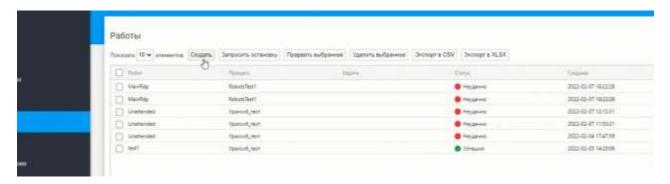
В результате экран «Роботы» должно выглядеть следующим образом:

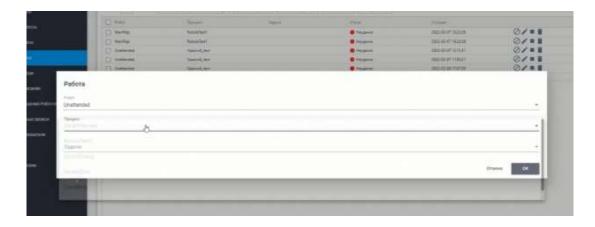


Координатор – зелёный индикатор (на связи), робот в учетной записи – красный индикатор (не на связи), т.к. его учетная запись разлогинена.

Тем не менее, можно создать работу, выбрать этого Unattended-Робота, а также выбрать какой-либо процесс или задачу, который он будет выполнять. Задачу можно указать, а можно и не указывать, это опциональная функция.







4.3.3.3 Создание работы на экране «Работы»

Работа может создаваться не только вручную, но и автоматически Оркестратором по мере отработки определенного расписания или работу можно завести через API.

После создания работы Оркестратор видит, что Unattended-Робот недоступен, но ему назначен Координатор, и этот Координатор доступен. Оркестратор направляет Координатору команду на включение учетной записи этого Unattended-робота. Координатор принимает команду и в рамках сервера поднимает RDP-подключение своей собственной учетной записи, в которой находится этот робот. Т.е. он запускает RDP-подключение, учетная запись включается, внутри учетной записи запускается Unattended-робот, в списке его индикатор становится зеленым. Он обращается к Оркестратору, видит, что в работах для него есть задача, принимает ее и отрабатывает.

Если запуск производится через такую цепочку действий, после выполнения задачи робот автоматически выйдет из своей учетной записи и учетная запись полностью выключится.



Автоматический выход из учетной записи — это стандартная практика, часть подхода enterprise-фреймворков, в которой каждый запуск какой-либо задачи изолируется от других запусков через перезагрузку учетной записи.

При возникновении нестандартных ситуаций в учетной записи, например, если одна из программ, с которой работает робот, зависает или появляется модальное диалоговое окно, которое блокирует все и робот не может продвигаться дальше, потому что данный вариант не был предусмотрен в сценарии. Сценарий Робота в этом случае завершается с ошибками, но программа, с которой работал Робот (1С, к примеру), остается «висеть» с модальным окном и получается, что интерфейс заблокирован. Когда в следующий раз по расписанию на этой же учетной записи запустится робот и начнет отрабатывать сценарий, он не сможет преодолеть заблокированную программу. Поэтому существует стандартная практика – когда Работа с Unattended-сценарием завершена (успешно или неуспешно), происходит его «Log off». Операционная система убирает все лишнее и к следующему запуску Робота все начинается с чистого листа.

Эта функция безусловна, то есть при запуске через Координатор, Logoff произойдет в любом случае.

Примечание: подобная организация взаимодействия с работами позволяет обрабатывать множество записей из очереди внутри одного запуска работы. В сценарии робота может быть множество разных блоков: получить сценарии из очереди, изменить статус задачи и другие. При запуске сценария из очереди берется задача, обрабатывается, затем берется следующая и так далее.

4.3.4 Sherpa RPA Runtime

Sherpa RPA Runtime — это установщик робота в учетную запись пользователя для использования без Дизайнера.

Это Runtime установщик робота, который используется во всех случаях, когда нужно установить робота без Дизайнера — т.е. для развертывания на рабочих местах конечных пользователей Attended-роботов, Unattended-роботов, а 210



также агентов RDP и Citrix. Когда он устанавливается в режиме Attended (галочка в инсталляторе), он может работать как для локального запуска Attended-роботов (но тогда нужна активированная лицензия на этом рабочем месте), так и для удаленного запуска и работы через RDP-подключение из другого Attended или Unattended-робота (тогда лицензия на нем не нужна).

Когда вы с клиентской машины (робота) устанавливаете подключение к удаленной машине, то робот на удаленной машине получает команды от робота на клиентской машине, например для работы с селекторами, и исполняет их. Коммуникация между локальным и удаленным экземпляром робота идёт через само RDP-подключение, то есть никакие порты не открываются.



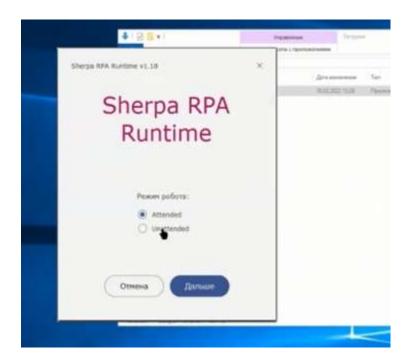
4.3.4.1 Установка Sherpa RPA Runtime

Отличие между Runtime и Sherpa RPA только в том, что в связке с Runtime нет Дизайнера.

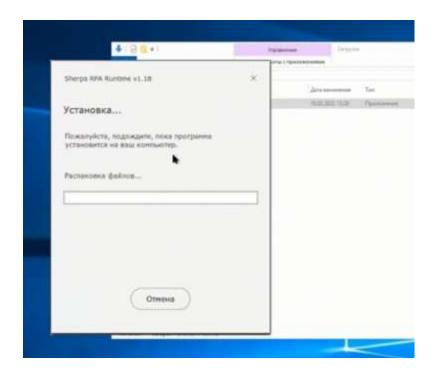
При установке Runtime создаются некоторые задания для Windows, а также некоторые установки в реестр для работы RDP.



После запуска необходимо выбрать режим робота: Attended или Unattended.



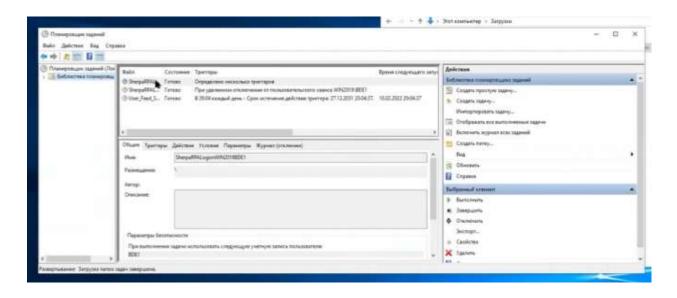
После нажатия кнопки «Дальше» происходит запуск установки.

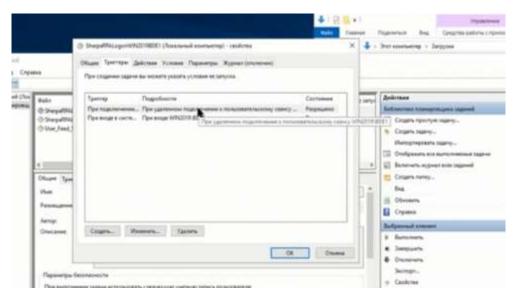


Отличие Attended- и Unattended -роботов – в создании задания для Windows.



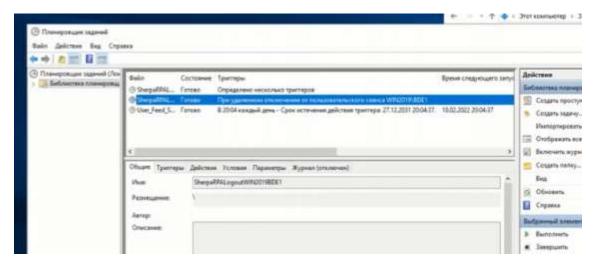
При установке Unattended-робота в планировщике заданий создаются два задания. Одно – тригтер на подключение к удаленной машине и вход в систему. Второе задание – на выход из системы.





Второе задание необходимо в случае, если вход на машину, а также отключение от нее происходит вручную, сработает отключение от сеанса.



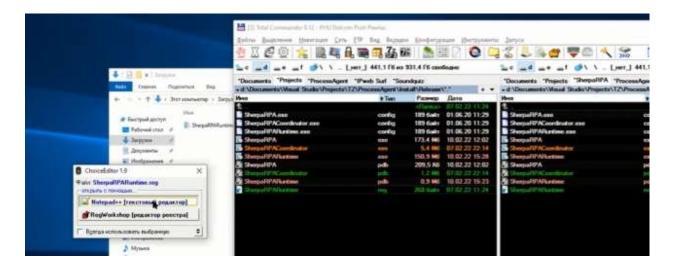


Когда Робот осуществляет процесс сам, он выполняет задания и осуществляет Log off.

При установке пакета .msi и файла .exe на рабочем столе появляется ярлык.

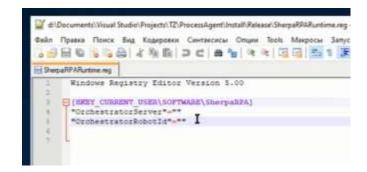


При установке msi их можно рассылать по соответствующим машинам, при этом, чтобы указать у них сервер Оркестратора и идентификатор робота, необходимо разослать или записать в реестр определенные записи групповыми политиками.



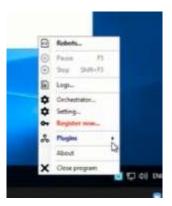


Произвести запись: текущий сервер, папка SOFTWARE.



Этот способ используется при рассылке msi.

В случае, простой установки запустить робота можно с помощью ярлыка на рабочем столе. При появлении ярлыка в системном трее, необходимо правой кнопкой мыши выбрать Оркестратор и войти в него.

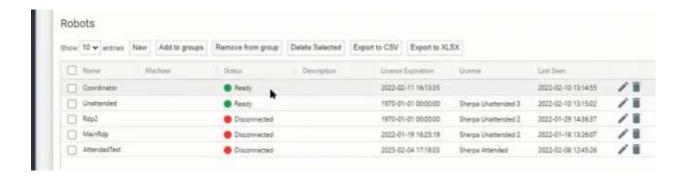


Далее ввести имя сервера и ID робота, которого будет использоваться. Далее осуществляем подключение.



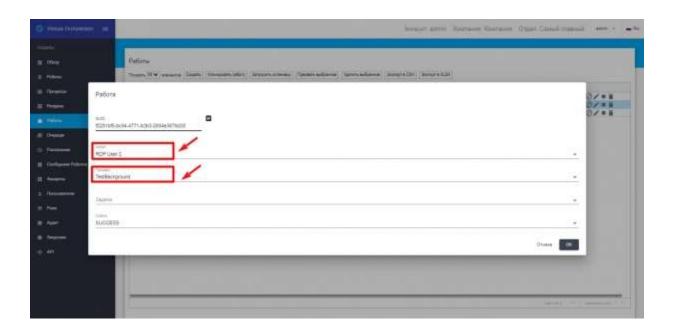


На экране «Роботы» отображаются зеленые индикаторы напротив Координатора и робота. Индикаторы сигнализируют о том, что Координатор запущен, а выбранный робот подключен.



В качестве примера создадим работу на экране «Работы». Выберем простую работу, которая создает ошибку деления на ноль. В результате действий робот должен произвести запуск (запуститься), а в «Сообщения Робота» должно прийти сообщение об ошибке.

В карточке «Работы» выбираем нужного робота и текущий процесс. Задачу в нашем примере можно выбрать любую, т.к. робот ее не будет использовать.

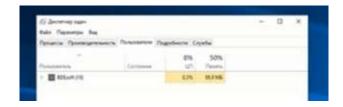


В Диспетчере задач среди процессов мы увидим, что робот запустил сессию.

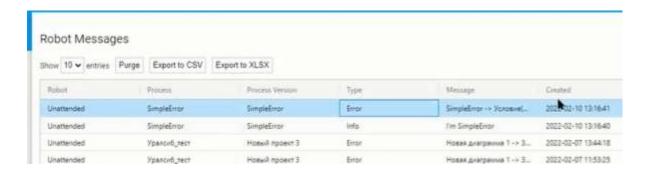




Далее закрыл ее.



Работа была выполнена роботом. На экране «Сообщения Роботов» Оркестратора мы видим, что пришло сообщение о том, что робот запустился и выдал ошибку.



Статус ошибка будет «Failed».



Таким образом происходит установка Координатора и самого Робота.

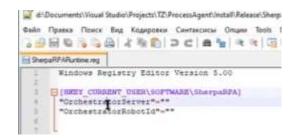
Примечание: можно разными способами записывать в реестр текущего Пользователя два параметра – сервер Оркестратор и ID Робота. Через групповые политики, программу, Power Shell. Это остается на усмотрение системного администратора.



MSI-пакет рассылается также через групповую политику. Машина в любом случае должна авторизоваться, чтобы этот пакет установился. Если пользователь не совершит вход, установка не произойдет. При входе пользователя можно инициировать запуск какого-либо скрипта PowerShell, который из какой-либо таблицы возьмет эти данные и запишет в реестр.

Если впоследствии нужно будет поменять ID, это можно будет сделать вручную: залогиниться через RDP в данного пользователя, запустить робота, зайти в его настройки (иконка появится в трее) и изменить ID. Также можно сменить ID удаленно.

Примечание: когда робот запускается и его настройки пусты, он считывает информацию о сервере и ID из реестра.



Единожды считав информацию из реестра, Робот записывает ее себе в настройки. Таким образом, если в дальнейшем появится необходимость изменить эти данные, нужно будет зайти на машину, запустить Робота с помощью ярлыка и внести необходимые изменения.

4.3 Интерфейс Sherpa Orchestrator

Пользовательский интерфейс Sherpa Orchestrator, открываемый по умолчанию, состоит из основных областей:

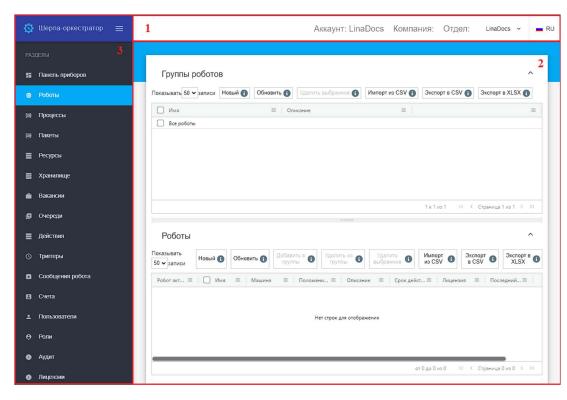
- Панель «Параметры пользователя».
- Рабочая область.
- Панель «Экраны».



Элементы пользовательского интерфейса Sherpa Orchestrator.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	панель «Параметры пользователя»	Содержит сведения о параметрах пользователя и позволяет совершать действия с текущим аккаунтом пользователя.
2.	Рабочая область	Область для работы с выбранным экраном.
3.	Панель «Экраны»	Содержит экраны: Обзор; Роботы; Процессы; Пакеты; Ресурсы; Хранилище; Работы; Очереди; Действия; Тригтеры; Сообщения робота; Аккаунты; Пользователи; Роли; Аудит; Лицензии; АРІ.

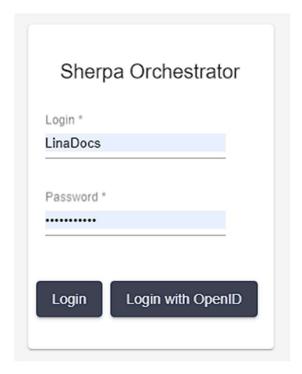




4.3.1 Панель «Параметры пользователя»

Для осуществления авторизации пользователя необходимо ввести логин и пароль учетной записи на отдельной странице авторизации Оркестратора. В случае, если авторизация осуществляется пользователем аккаунта (подробнее об аккаунтах описано в разделе 4.3.3.11 Экран «Аккаунты»), то необходимо ввести логин и пароль от учетной записи аккаунта, также поддерживается авторизация через OpenID.





Время жизни сессии без активных действий пользователя — 15 минут. Кнопка Logout очищает текущую сессию пользователя и возвращает пользователя на экран авторизации. В рамках Аккаунта можно создать любое количество Пользователей, при этом каждый из Пользователей может быть участником только одного Аккаунта.

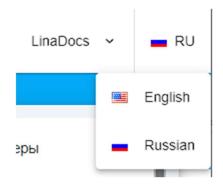
В верхней панели Оркестратора находится следующая информация:

- Аккаунт имя текущего аккаунта (если авторизация через аккаунт) или Аккаунта, к которому относится авторизованный Пользователь;
- Компания компания, которая указана при создании аккаунта (учетной записи), к которому относится авторизованный пользователь;
- Отдел отдел компании, который указан при создании текущего аккаунта (учетной записи), к которому относится авторизованный пользователь;
- Имя текущего Пользователя активная кнопка, по нажатию которой появляется кнопка «Выход».





Также в верхнем поле доступна кнопка, по которой можно поменять язык интерфейса Оркестратора, если это необходимо.



4.3.2 Рабочая область Оркестратора

На скриншоте ниже приведены общие для всех экранов действия на примере экрана «Роботы». Отличия в некоторых экранах будут приводится в соответствующих разделах.

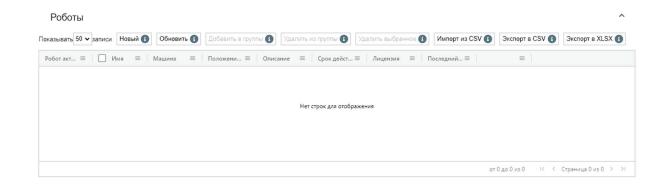


Элементы интерфейса Рабочей области Оркестратора. Таблица «Группы элементов».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	Показать ‹› элементов	Позволяет выбрать количество отображаемых групп элементов на одной странице (10, 25, 50).
2.	кнопка «Создать»	Позволяет создать новую группу элементов.
3.	кнопка «Обновить»	Позволяет обновить список групп элементов.
4.	кнопка «Добавить в Группы»	Позволяет добавить элемент в группу.



5.	кнопка «Удалить из Группы»	Позволяет удалить выбранный элемент из группы.
6.	кнопка «Удалить выбранное»	Позволяет удалить выбранный элемент.
7.	Импорт из CSV	Позволяет импортировать группы элементов из CSV-файла.
8.	Экспорт в CSV	Позволяет экспортировать выбранные группы элементов в CSV-файл.
9.	Экспорт в XLSX	Позволяет экспортировать выбранные группы элементов в XLSX-файл.
10.	колонка «Имя»	Наименование группы элементов.
11.	колонка «Описание»	Краткое описание группы элементов.
12.	кнопка	Позволяет изменить имя и описание выбранной группы элементов.
13.	кнопка	Позволяет удалить выбранную группу элементов.
14.	I< Страница 1 из 1 > >I	Навигация по страницам.





Элементы интерфейса Рабочей области Оркестратора. Таблица «Элементы».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	Показать ‹› элементов	Позволяет выбрать количество отображаемых элементов на одной странице (10, 25, 50).
2.	кнопка «Создать»	Позволяет создать новый элемент (в зависимости от экрана – робота, работу, процесс и т.д.).
3.	кнопка «Обновить»	Позволяет обновить список элементов.
4.	кнопка «Удалить выбранное»	Позволяет удалить выбранные элементы.
5.	Импорт из CSV	Позволяет импортировать элементы из CSV-файла.
6.	Экспорт в CSV	Позволяет экспортировать выбранные элементы в CSV-файл.
7.	Экспорт в XLSX	Позволяет экспортировать выбранные элементы в XLSX-файл.
8.	колонка «Имя»	Наименование элемента.
9.	колонка «Описание»	Краткое описание элемента.
10.	кнопка 🖍	Позволяет изменить имя и описание выбранного элемента.
11.	кнопка	Позволяет удалить выбранный элемент.
12.	I< Страница 1 из 1 > >I	Навигация по страницам.

4.3.2.1 Действия с данными таблиц экранов Оркестратора

Действия с данными таблиц одинаковые для всех экранов:

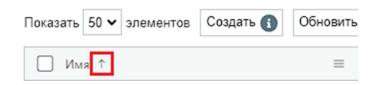
• изменение порядка колонок;



- сортировка;
- настройка отображения колонок;
- фильтр;
- настройка видимости колонок.

Для изменения порядка колонок необходимо зажать левую кнопку мыши на выбранной колонке и перенести ее.

Для выполнения сортировки необходимо нажать левой кнопкой мыши по названию нужной колонки, после чего рядом с названием появится стрелка , которая покажет сортировку данных. Для текстовых колонок сортировка происходит в алфавитном порядке. Для колонок с типом даты - сортировка происходит по дате.

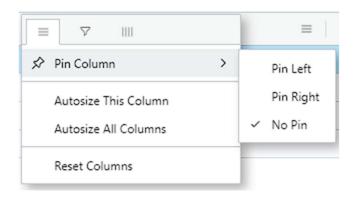


Чтобы получить доступ к настройкам отображения колонок, необходимо нажать на кнопку . Данная опция позволяет:

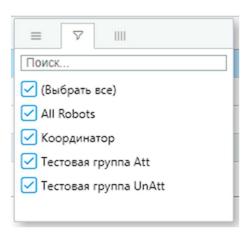
- 1) Pin column закрепить выбранную колонку:
- Pin Left закрепить выбранную колонку слева;
- Pin Right закрепить выбранную колонку справа;
- No Pin не закреплять выбранную колонку.
- 2) Autosize This Column установить автоматический размер выбранной колонки;
- 3) Autosize All Column установить автоматический размер всех колонок таблицы;



4) Reset Columns – очистить настройки отображения выбранной колонки.

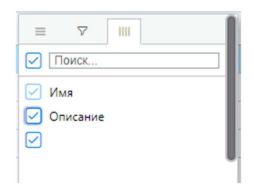


Чтобы отфильтровать содержимое таблицы элементов или таблицы групп элементов, необходимо нажать на кнопку и в открывшемся окне выбрать вкладку $\overline{}$. Фильтр позволяет оставить в выдаче результатов только определенные строки в таблице по их названию, временно отключив остальные.



Чтобы настроить видимость колонок, необходимо нажать на кнопку и в открывшемся окне выбрать вкладку . В этой вкладке можно выбрать, какие колонки таблицы будут отображены в выдаче результатов.





4.3.3 Панель «Экраны»

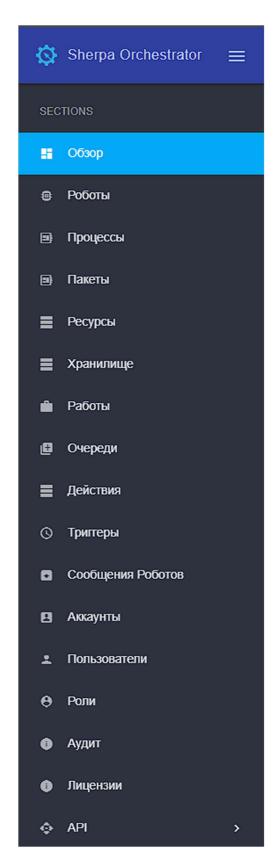
Sherpa Orchestrator. Экраны.

№ п/п	Экран	Описание экрана
1.	Обзор	Дашборд с набором виджетов, отображающих основные параметры и срезы данных по работе программных роботов и Оркестратора. Здесь отображается базовая статистика дашборда (количество роботов, процессов и т.д.).
2.	Роботы	Отображает таблицы «Роботы» и «Группы Роботов». Позволяет совершать различные операции (создание, редактирование, группирование и т.д.) со всеми доступными Роботами в Оркестраторе.
3.	Процессы	Отображает сценарии роботизации, которые выполняются роботами, и их имеющиеся версии.
4.	Пакеты	Отображает пакеты и их имеющиеся версии.
5.	Ресурсы	Отображает список объектов, доступный всем роботам, группе роботов или конкретному роботу. Поддерживаемые типы ресурсов: текст (текстовая строка), реквизиты (пара логин-пароль), календарь.
6.	Хранилище	Отображает список файлов и папок.
7.	Работы	Отображает совокупность процесса и робота, которому данный процесс назначен на выполнение.



8.	Очереди	Позволяет группировать однотипные задачи, создавать потоки задач и добавлять задачи в очереди.
9.	Действия	Позволяет создавать действия для задач в очереди, а также создать форму для выполнения задачи роботом.
10.	Тригтеры	Описывает условия, наступление которых позволяет автоматически создавать Работы, запускающие далее роботов.
11.	Сообщения роботов	Содержит информацию о сообщениях, которые роботы отправляют в Оркестратор по мере выполнения работ.
12.	Аккаунты	Позволяет создавать новые аккаунты, а также редактировать существующие.
13.	Пользователи	Позволяет создавать новых пользователей, а также редактировать существующих.
14.	Роли	Определяет набор прав доступа к тем или иным экранам Оркестратора и функциями в рамках этих экранов.
15.	Аудит	Показывает собственные события Оркестратора.
16.	Лицензии	Отображает все лицензии Оркестратора, а также набор инструментов по созданию, активации и управлению ими.
17.	API	Содержит набор инструментов для обмена данными через API (справочник по существующим APIметодам, инструменты получения токена для работы с API).

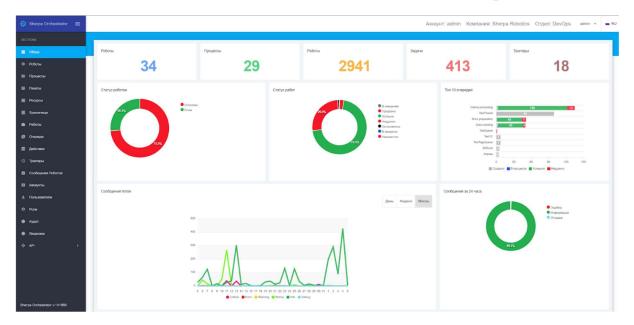




4.3.3.1 Экран «Обзор»

Экран «Обзор» представляет собой дашборд с набором виджетов, отображающих основные параметры и срезы данных по работе программных роботов и Оркестратора.





Доступны следующие параметры для отображения на дашборде экрана «Обзор»:

Роботы - общее количество роботов, управляемых Оркестратором. Под роботами понимается программа-исполнитель, экземпляры которой запущены на компьютере пользователя или на отдельно стоящих виртуальных машинах.

Процессы – общее количество процессов, хранящихся в Оркестраторе. Процессы – это сценарии роботов.

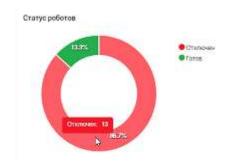
Работы — общее количество работ, управляемых Оркестратором. Работы — это взаимодействие между конкретными роботами и процессами, которые они выполняют.

Задачи – общее количество задач, управляемых Оркестратором. Задачи позволяют автоматизировать процесс создания работ, они могут быть поставлены в очередь с определенной последовательностью или по расписанию.

Тригтеры - количество настроенных тригтеров для запуска роботов.

Статус роботов – круговая диаграмма для отображения готовых к работе, занятых работами и недоступных Роботов.





Статус роботов показывает какое количество роботов находится в статусе «Готов» или в статусе «Отключен» в числовом и процентном соотношении (%).

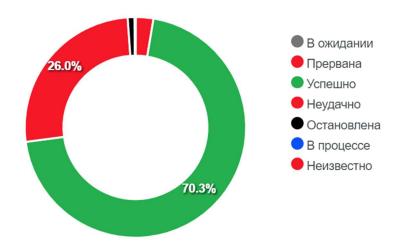
Количество отключенных роботов входит в число всех роботов, которые установлены в учетной записи на терминальных серверах и находятся под управлением Координатора. Они отражается в статусе «Отключен», но фактически находятся в режиме ожидания, и могут быть запущены в любой момент по команде Оркестратора при необходимости.

Статус работ – круговая диаграмма для отображения количества работ, находящихся в одном из состояний:

- Создано;
- В ожидании;
- Прерывается;
- Прервана;
- Успешно;
- Неудачно;
- Останавливается;
- Остановлена;
- В процессе;
- Неизвестно.



Статус работ



При наведении курсора мыши, показывает статусы работ в числовом и процентном соотношении.

Топ 10 очередей – гистограмма для отображения количества и соотношения выполненных задач в очередях в разрезе статусов их выполнения (успешно, неудачно, в процессе).



При наведении курсора мыши, показывает, какие процессы находятся в очереди, также отображается статус, в котором находятся заявки в этих очередях задач. 10 очередей — максимальное количество, которое отображается на дашборде.



Сообщения логов – график для отображения поступления сообщений в логи Оркестратора.



Отображаются различные типы ошибок:

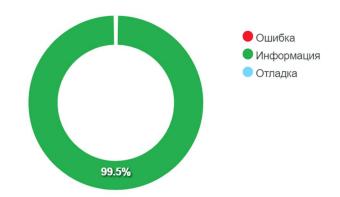
- Критично
- Ошибка
- Предупреждение
- Уведомление
- Информация
- Отладка

Есть возможность указать период для анализа возникающих ошибок (день, неделя, месяц).

Сообщения за 24 часа – поле для оперативного информирования о поступивших логах за последние 24 часа.



Сообщения за 24 часа



4.3.3.2 Экран «Роботы»

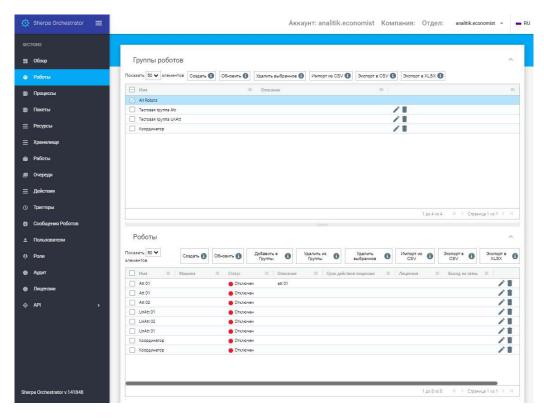
Роботы — экземпляры программного обеспечения платформы Sherpa RPA, развернутые на конкретных рабочих местах пользователей, виртуальных машинах и терминальных серверах и подключенные для управления к Оркестратору. Роботы могут быть объединены в группы роботов, при этом один робот может быть включен в несколько групп.

Для более гибкого управления доступом робота к тем или иным ресурсам один и тот же робот может быть назначен для разбора разных очередей задач по разным процессам.

Интерфейс экрана Роботы

Экран Роботы содержит две таблицы: «Группы Роботов» и «Роботы».



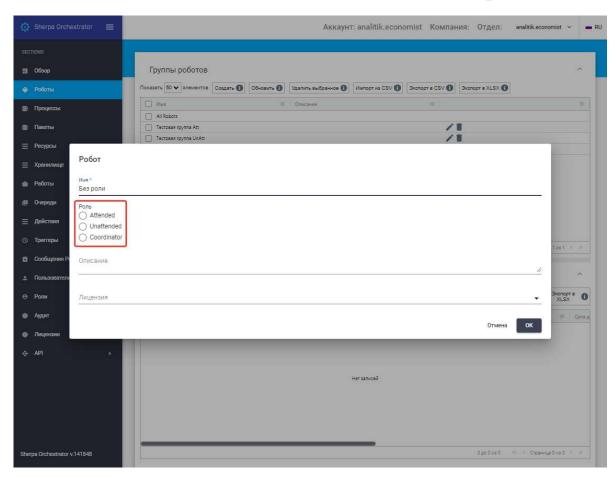


Создание нового «Робота»

Для создания нового робота нажать на кнопку «Создать» в таблице «Роботы» и заполнить поля:

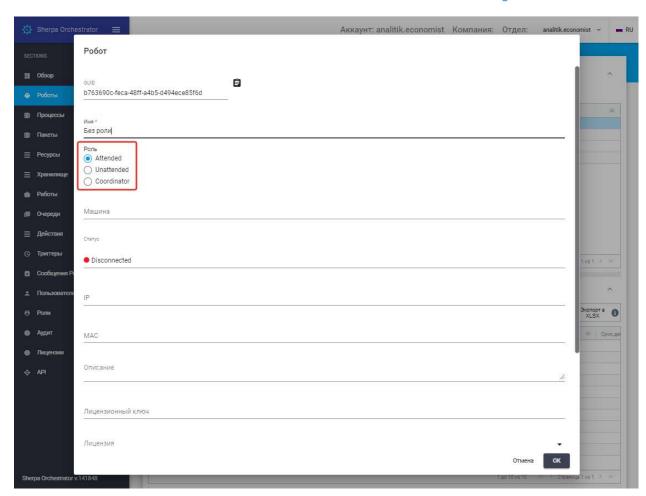
- Имя (обязательное);
- Роль;
- Описание;
- Лицензия.





Обратите внимание! Если не заполнить поле «Роль» и сохранить, то по умолчанию, флаг будет установлен на роли «Attended».





Для Attended-роботов достаточно задать отображаемое название и выбрать используемую роботом лицензию.

Для Unattended-роботов, кроме наименования и лицензии, необходимо указать IP адрес или доменное имя терминального сервера или виртуальной машины, на которой развернут данный робот, логин (при необходимости включающий домен) и пароль от пользовательской учетной записи, в которой должен запускаться робот, а также выбрать экземпляр Координатора, который будет использован для запуска данного робота. Подробнее об Координаторе и его установке в разделе «4.3.3 Координатор Unattended-роботов».

После заполнения карточки по кнопке «ОК» необходимо сохранить внесенные правки. Робот будет создан.



Для просмотра и редактирования свойств определенного робота в списке выбрать интересующего робота и нажать на кнопку редактирования. После этого откроется окно с его настройками.

Таблица «Группы Роботов»

Данная таблица позволяет группировать роботов между собой, объединять их в логические группы.

Например, при необходимости можно создать группу для работы в рамках определенного терминального сервера, присвоить ей имя и добавить в нее соответствующих роботов.

Для того, чтобы в таблице «Роботы» отобразился список роботов, предварительно нужно выбрать необходимую «Группу Роботов».

Группа «All Robots «перечисляет всех доступных роботов, имеющихся в оркестраторе. Она создается автоматически и недоступна для удаления.

При выборе интересующей группы в нижней таблице «Роботы» отобразится список всех роботов, относящихся к выбранной группе, в том числе и робот с функцией Координатор (для Unattended-роботов).

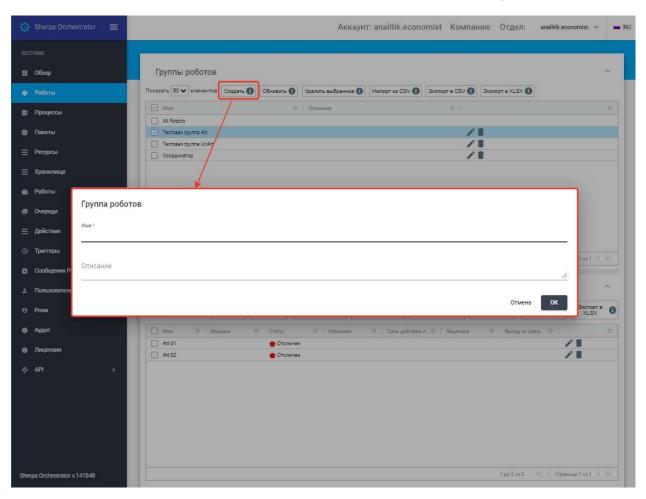
Обратите внимание! Координатор всегда находится во включенном состоянии, о чем сигнализирует зеленый индикатор. Данный сервис обеспечивает создание и удержание сессий пользовательских учетных записей для Unattended-роботов в рамках используемого терминального сервера.

Все остальные роботы – это технические учетные записи, запущенные на том же терминальном сервере.

Создание новой «Группы Роботов»

Для создания новой Группы Роботов нажать на кнопку «Создать» в таблице «Группы Роботов» и заполните все поля карточки.





4.3.3.3 Экран «Процессы»

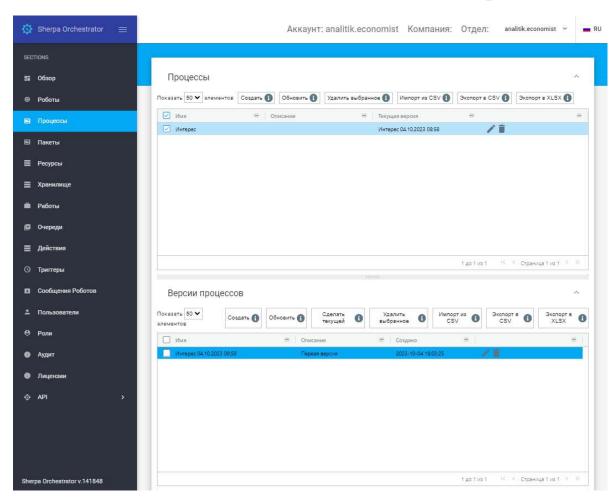
Процессы — это сценарии роботизации, которые выполняются роботами. Для каждого процесса может быть сохранено несколько его версий. При этом только одна версия процесса является на данный момент текущей, то есть именно эта версия процесса используется при назначении роботу новой работы, вручную, с помощью расписаний или с помощью API.

Интерфейс экрана «Процессы»

Экран Процессы содержит две таблицы: «Процессы» и «Версии процессов».

Для того, чтобы в таблице Версии процессов появился список версий, кликните на нужном процессе в списке Процессы.

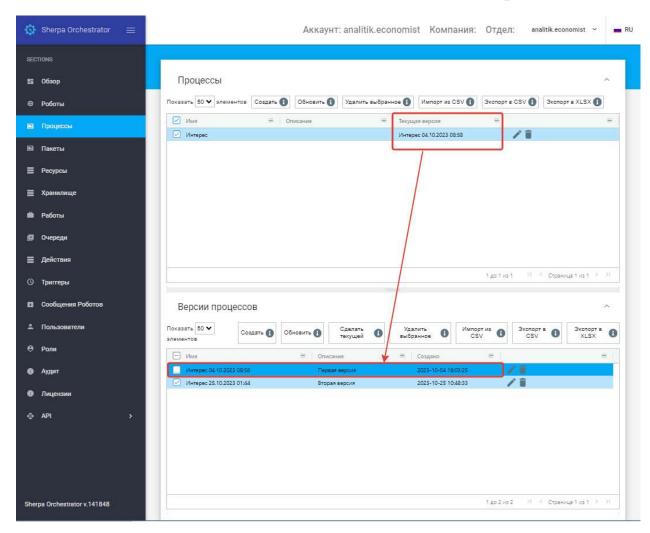




По умолчанию одна из версий является текущей, она отображается в колонке «Текущая версия» таблицы «Процессы».

В таблице «Версии процессов» текущая версия подсвечивается синим цветом при выборе процесса в таблице «Процессы».





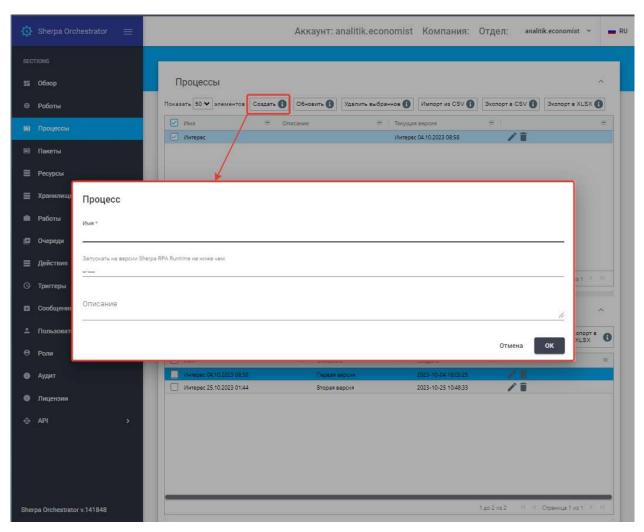
Создание нового процесса

Для создания нового процесса необходимо в таблице «Процессы» в меню нажать на кнопку «Создать» и в открывшейся форме заполнить поля карточки: Имя и Описание.

Поле «Описание» не является обязательным. При необходимости в нём можно оставить комментарий – поле текстовое.

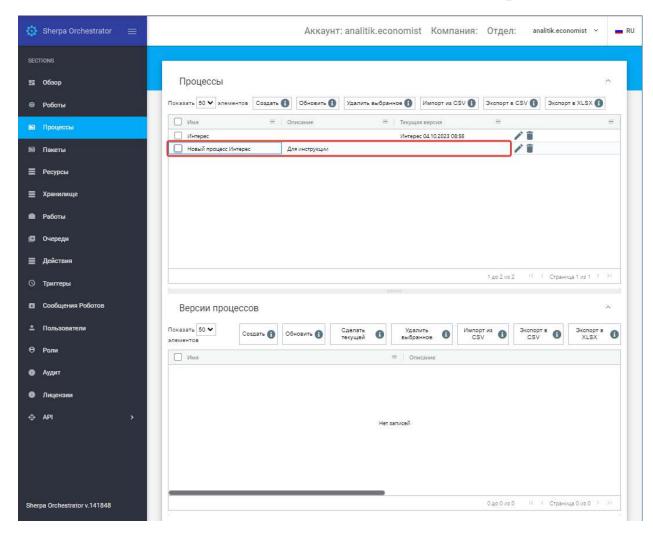
Поле «Запускать на версии Sherpa RPA Runtime не ниже чем» — позволит явно указать номер версии Sherpa RPA Runtime, ниже которой процесс не будет запускаться.





После заполнения полей необходимо сохранить процесс, нажав «ОК». В списке появится вновь созданный элемент – процесс. У вновь созданного процесса нет версий.





Создание новой Версии процессов из Оркестратора

Для создания новой версии процесса необходимо в таблице Процессы выбрать тот процесс, для которого будет создана новая версия (1). Если процесс не отмечен, то Оркестратор выдаст сообщение с предупреждением.

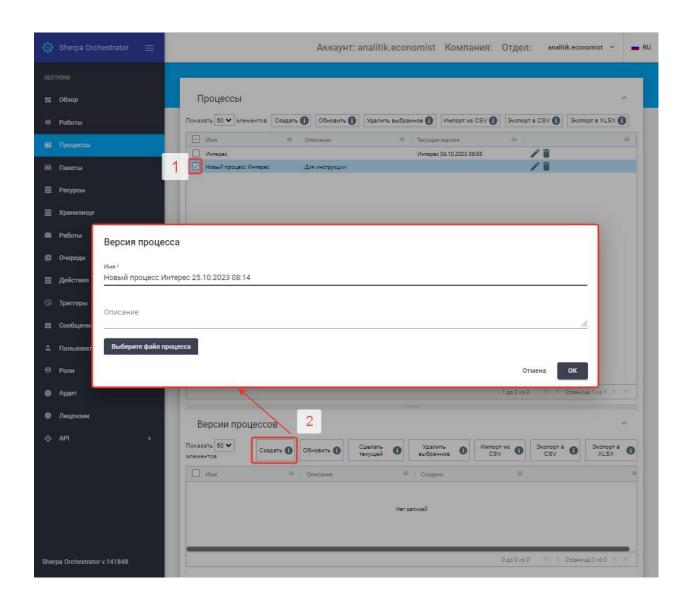
Уведомление от сайта orchestrator2.sherparpa.ru

Сначала выберите процесс

Закрыть



Затем в таблице «Версии процессов» нажать на кнопку «Создать» (2) и в открывшейся форме заполнить поля (Имя, Описание) и выбрать файл процесса. Затем нажать «ОК».

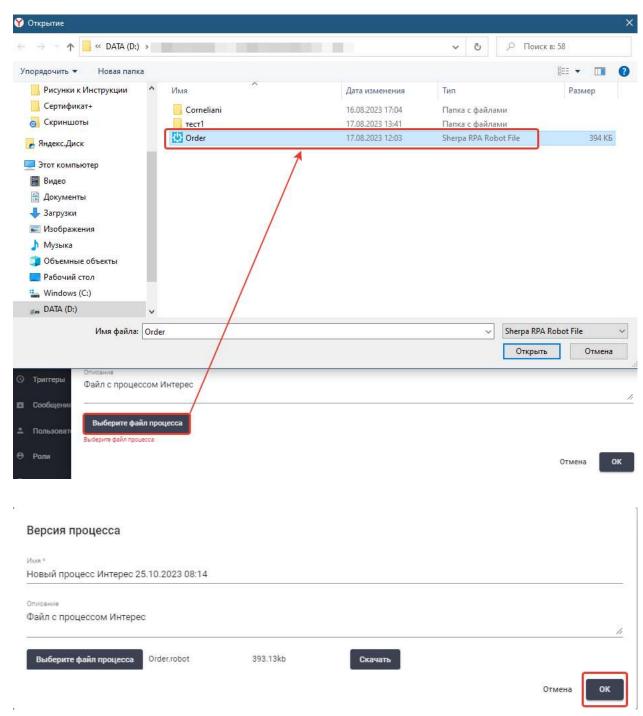


Обратите внимание! По умолчанию, в поле Имя проставляется наименование Процесса, текущая дата и время создания версии. Их можно изменить на актуальное название.

Поле «Описание» не является обязательным – в нем можно указать необходимый текст (например, изменения в новой версии).

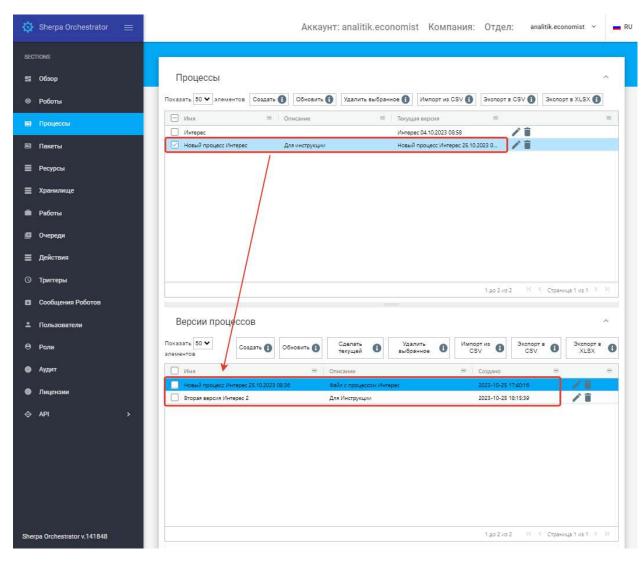
Требуется обязательно приложить файл процесса (файл с расширением .robot). Без файла версия не будет сохранена.





Для каждого процесса можно создавать несколько его версий.



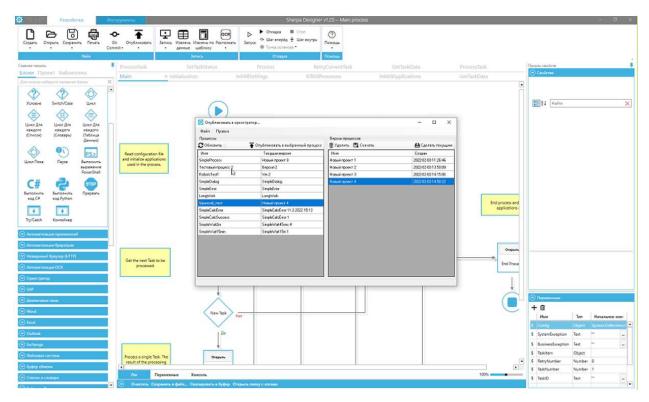


Создание новой Версии процессов из Дизайнера

Создание новой версии процесса можно проводить из Дизайнера. Для этого в Sherpa Designer в меню раздела «Опубликовать» предусмотрена кнопка «Опубликовать в Оркестратор», которая добавляет готовый проект в Оркестратор.

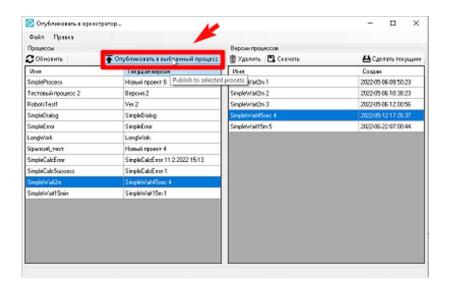






После нажатия кнопки «Опубликовать» или «Опубликовать в Оркестратор» откроется окно со списком добавляемых процессов и список версий этих процессов.

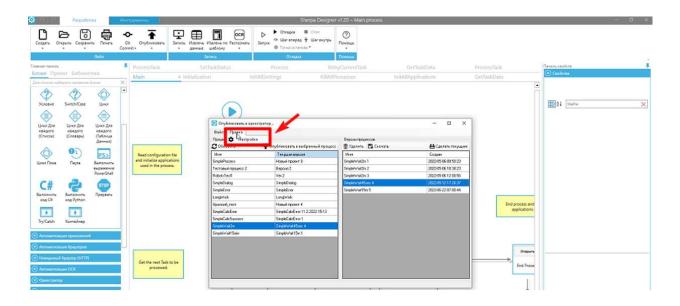
Выбрав нужный процесс и версию, необходимо нажать кнопку «Опубликовать выбранный процесс». После этого выбранный процесс появляется в Оркестраторе.



При публикации в списке версий появляется новая версия процесса, причем она не является текущей по умолчанию.

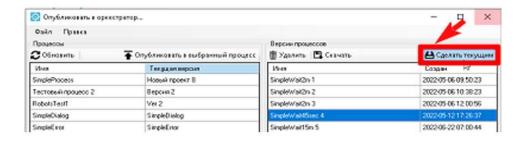


При необходимости подключиться к другому Оркестратору из Дизайнера, а также для смены параметров подключения, воспользуйтесь функцией «Правка» – далее «Настройки».



Определение текущей Версии процессов

Сделать версию процесса текущей можно нажав кнопку «Сделать текущей» при публикации из Дизайнера.



Для определения Текущей версии на стороне самого Оркестратора необходимо выделить текущую версию процесса явным образом, то есть для нужной строки с названием версии процесса требуется нажать кнопку Сделать текущим.





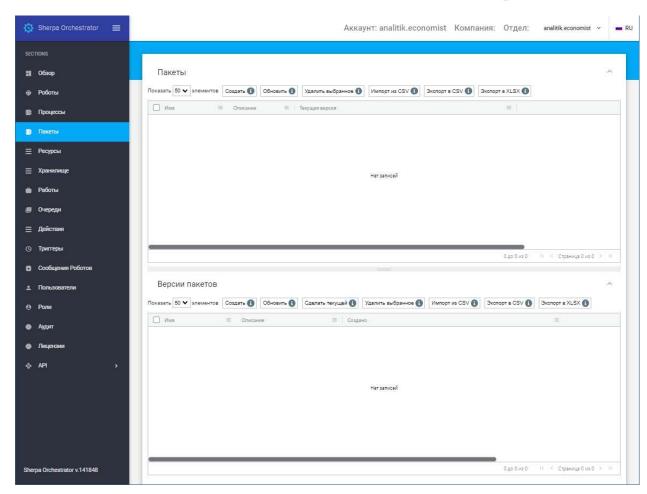
Тогда в колонке Текущая версия в таблице Процессы появится название текущей версии сценария. Именно эта версия будет использоваться при создании новых Работ с применением данного Процесса.

Примечание: помимо встроенного функционала версионирования процессов, Sherpa Designer также содержит интеграцию с системой контроля версий Git, которая может быть использована в процессе разработки. После завершения разработки процесса, он может быть опубликован из Sherpa Designer в Оркестратор для использования в составе встроенной в Оркестратор системы версионирования. Сам Оркестратор поддержки прямой работы с Git не имеет.

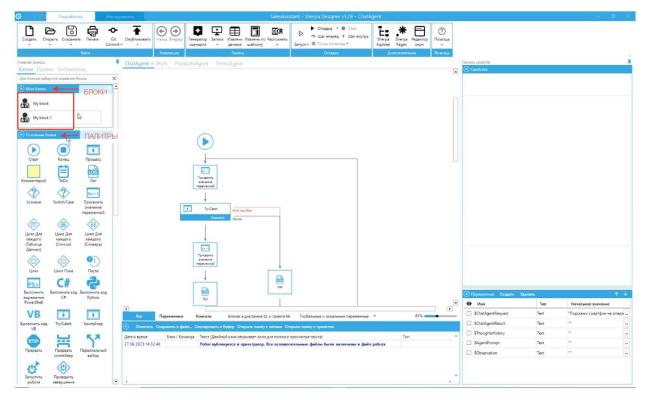
4.3.3.4 Экран «Пакеты»

Экран «Пакеты» имеет структуру аналогичную Процессам: «Пакеты» и «Версии пакетов».



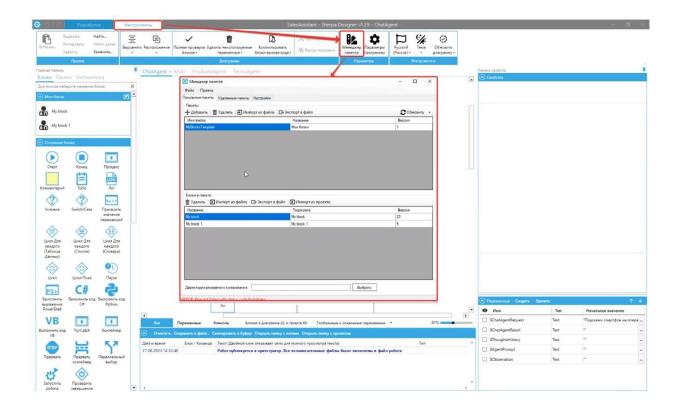


B Sherpa Designer можно создавать собственные блоки через редактор и объединять их в палитры.

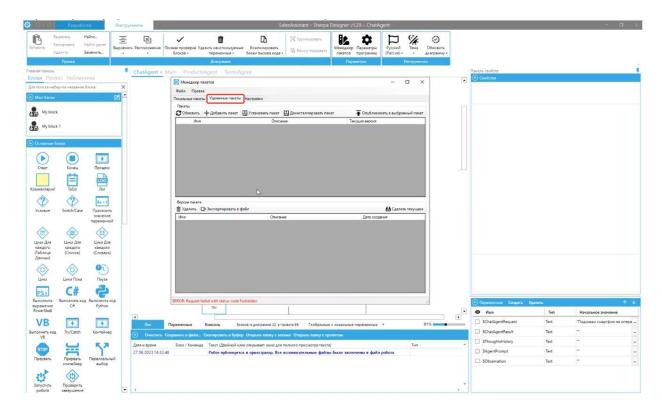




Блоки и палитры можно собрать в пакеты: для этого есть специальный инструмент – «Менеджер пакетов».



Пакетами можно обмениваться через файлы (Вкладка «Локальные пакеты») или через Оркестратор (Вкладка «Удаленные пакеты»).





Разработчик может создавать свои палитры блоков и использовать аналогичные палитры других разработчиков по согласованию. Возможность обмена реализована через обмен с помощью «Менеджера пакетов». Этот же инструмент позволяет получать обновления по палитрам и проводить обновление версий блоков.

4.3.3.5 Экран «Ресурсы»

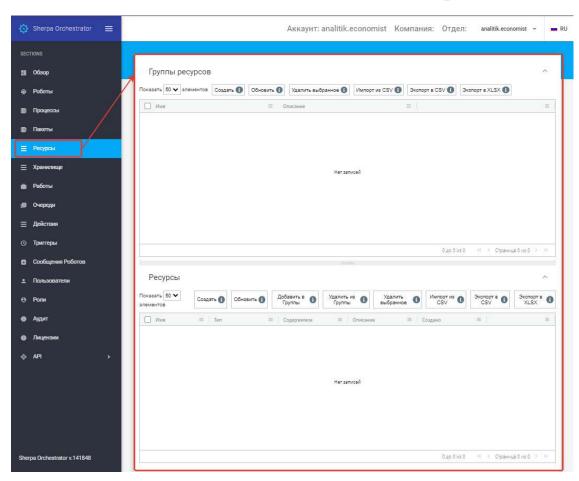
Экран «Ресурсы» представляет собой таблицы «Группа ресурсов» и «Ресурсы» со списком объектов, доступным всем роботам, группе роботов или конкретному роботу. Здесь можно создавать ресурсы, которые будут использоваться роботами.

Робот или роботы, имеющие доступ к ресурсу, могут запросить или установить значение соответствующего ресурса через соответствующие блоки или через АРІ. Поддерживаемые типы ресурсов:

- текст (текстовая строка),
- реквизиты (пара логин-пароль),
- календарь.

Пароли хранятся в базе в зашифрованном виде (используется алгоритм шифрования AES256).





К ресурсам Оркестратора можно обращаться из робота, получать и устанавливать значения глобальных переменных, а также получать или устанавливать реквизиты (пара Логин/Пароль).

Со стороны робота можно воспользоваться ресурсами с помощью блоков из палитры Оркестратор Sherpa Designer.

Создание нового ресурса

Для создания нового ресурса кликните на кнопку «Создать» в таблице «Ресурсы» и заполните все поля карточки.



Ресурс	
MMR *	
Укрыпте имя	
Описание	
Тип и содержимое ресурса © Рексизиты	
Логин аккаунта	
analitik.economist	
Пароль	
tingstrie	
Срок действия пароля	
дд.мм.гггг	
() Текст	
○ Календарь	
Ресурс доступен:	
Все роботы	
О Только определенный робот	
О Только определенная группа роботов	
	Отмена ОК

Поле карточки	Правила заполнения				
Имя	Название Ресурса. Обязательное поле заполняется в произвольной форме				
Тип и содержимое Ресурса	 В зависимости от типа ресурса заполняется его содержимое: Реквизиты – логин и пароль. Обязательное поле. Указываются логин и пароль Аккаунта. Текст – текстовое поле. Заполняется в произвольной форме Календарь – табло производственного календаря. Позволяет указать рабочие/ нерабочие дни 				
Ресурс доступен	Позволяет назначить робота или Группу Роботов, для которых будет доступен создаваемый Ресурс: • Все Роботы • Только определенный Робот • Только определенная Группа Роботов				

Типы Ресурсов

Ресурсы бывают трёх типов:

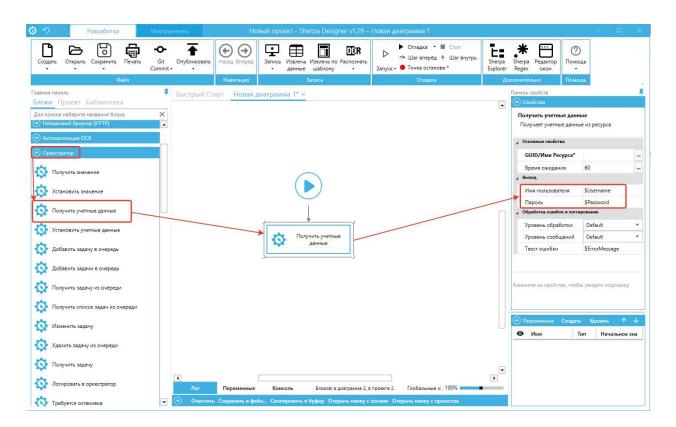
- Реквизиты
- Текст
- Календарь



Реквизиты

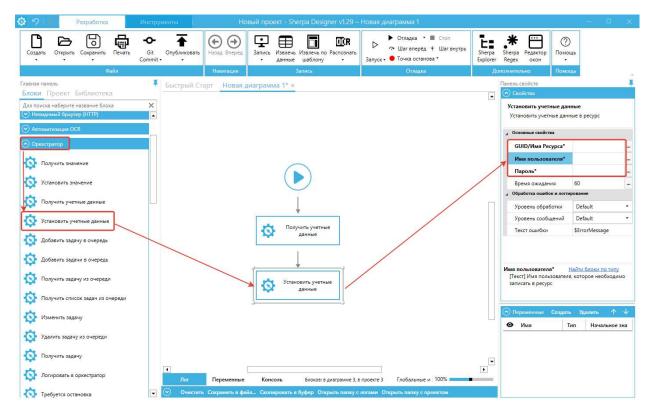
Реквизиты включают в себя Логин и Пароль. Пароль будет храниться в базе данных Оркестратора в зашифрованном виде, передаваться Пароль будет также в зашифрованном виде (тип шифрования AES 256). Здесь можно настроить срок действия пароля, указав дату окончания действия (после этого пароль необходимо заменить). Записи с истекшим сроком действия пароля будут выделены цветом.

Для доступа к ресурсам типа «Реквизиты» (пара Логин/Пароль) на стороне Дизайнера в палитре Оркестратора используется блок «Получить учетные данные».



Для редактирования учетных данных используется блок «Установить учетные данные».





Данный функционал доступен для Роботов любого типа: Attended и Unattended.

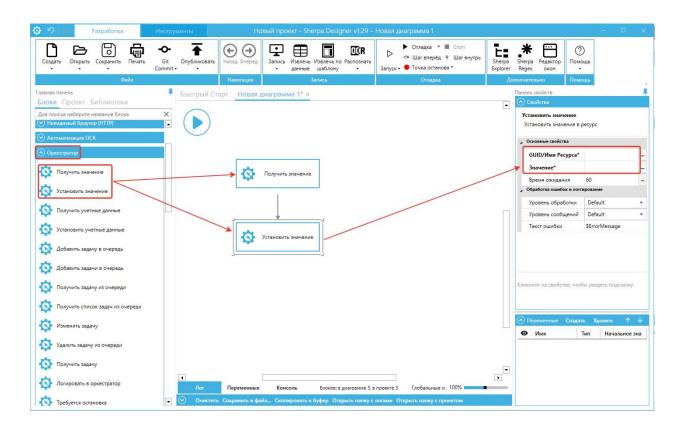
Текст

Ресурс «Текст» представляет собой любую произвольную строку. Позволяет хранить текст, числа и другие типы ресурсов. Ограничение на количество символов при выборе данного типа Ресурсов составляет 10 000 символов.



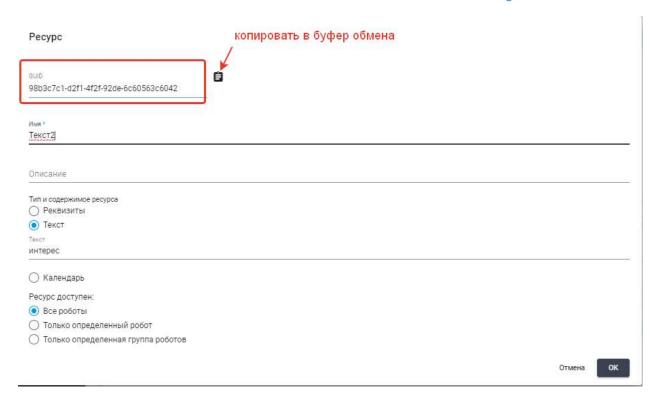


Для доступа к ресурсам типа «Текст» на стороне Дизайнера в палитре Оркестратора используются блоки «Получить значение» и «Установить значение».

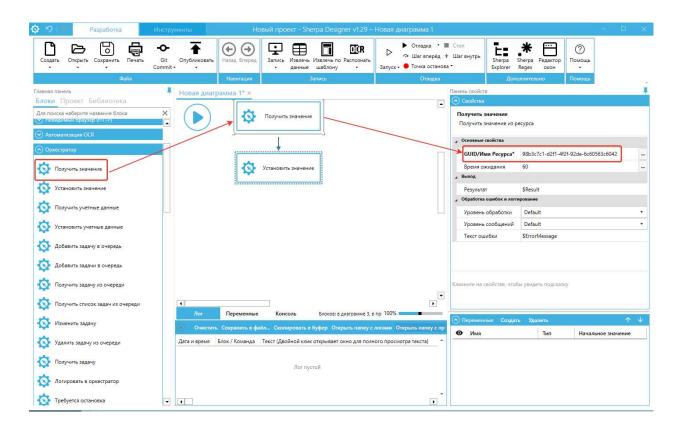


Из Оркестратора нужно скопировать GUID,





который указывается далее в Дизайнере в поле GUID/Имя Ресурса в свойствах блока.





После выполнения выбранного блока в переменную «Результат» на стороне Дизайнера будет передаваться значение, указанное в Оркестраторе в поле «Текст».

Календарь

Ресурс содержит информацию о количестве рабочих, выходных и праздничных дней. Они используются далее в нескольких местах:

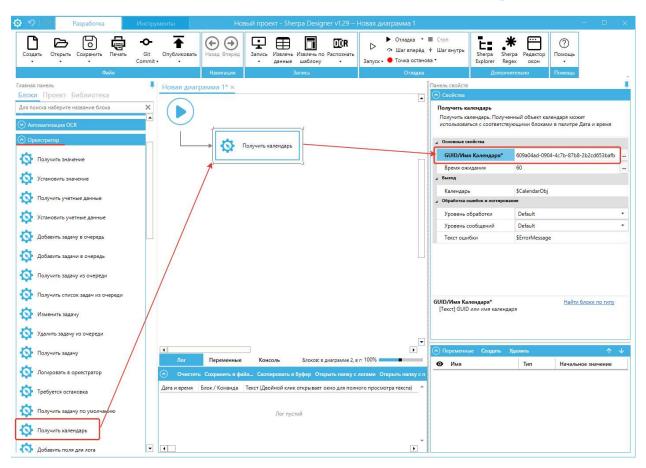
- в настройке триггеров (можно указать, например, чтобы робот не запускался в выходные дни);
- в палитре даты и время в блоках (например, подсчитать рабочие дни и т.п.).

Pecypc								
GUID	â							
98b3c7c1-d2f1-4f2f-92de-6c60563c6	5042							
Vion *								
Текст2								
Описание								
Гип и содержимое ресурса								
○ Реквизиты								
Текст€ Календарь								
Редактировать календарь	Экспортировать	календарь	Импортиро	вать календар	Ь			
Ресурс доступен:								
Все роботы								
○ Только определенный робот								
 Только определенная группа ро 	оботов							
							Отмена	OK

Ресурс «Календарь» позволяет настраивать запуск робота, например, только в рабочие дни, исключая выходные и праздники.

Для доступа к ресурсам типа «Календарь» на стороне Дизайнера в палитре Оркестратора используется блок «Получить календарь». Обязательно требуется ввести GUID (Имя) Календаря, который требуется получить.





Также в палитре «Дата и время» есть блоки, которые могут воспользоваться полученным из Оркестратора календарем для проведения различных проверок и вычислений.

- Подсчитать рабочие дни считает количество рабочих дней между двумя заданными датами.
- Следующий рабочий день возвращает следующий рабочий день, начиная с указанной даты.
 - Рабочий день проверяет указанную дату на рабочий день.
 - Выходной день проверяет указанную дату на выходной день.
- Получить рабочие дни получает рабочие дни между двумя заданными датами.
- Получить выходные дни получает рабочие дни между двумя заданными датами.

Область видимости Ресурса



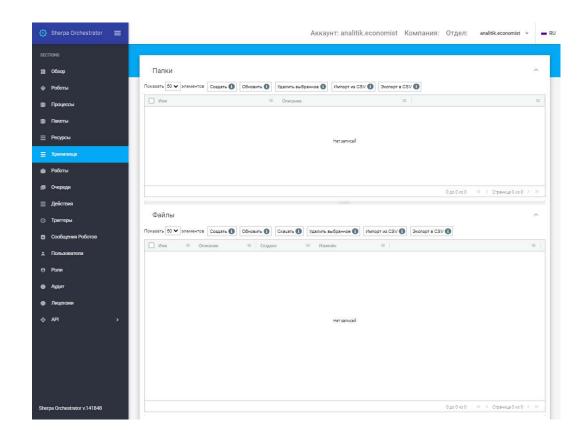
Для ресурсов можно задать область видимости: она может быть настроена в поле **Ресурс доступен:**

- Все Роботы ресурс доступен для всех роботов в Оркестраторе;
- Только определенные Роботы робота необходимо выбрать из списка доступных (выпадающий список);
- Только определенные Группы Роботов роботы, которые входят в определенные группы роботов (выпадающий список).

4.3.3.6 Экран «Хранилище»

Хранилище можно рассматривать как аналог «File Bucket» или Google Drive, встроенный в Оркестратор, который позволяет хранить произвольные файлы разного размера. Файлы могут быть сохранены как пользователем (человеком), так и роботом.

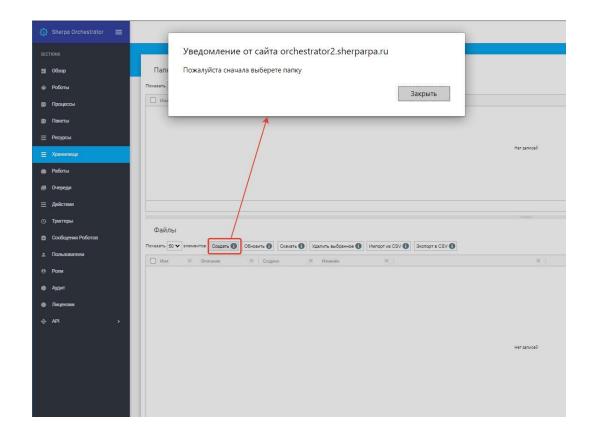
Экран Хранилище состоит из двух таблиц: «Папки» и «Файлы».

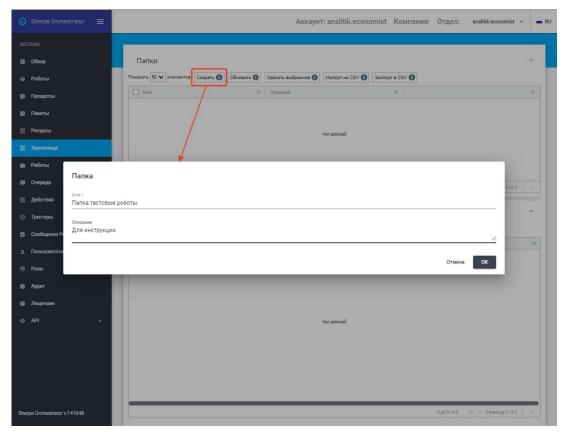


В Хранилище не предусматривается многоуровневая структура папок – двухуровневая (папка или группа файлов, и файлы).

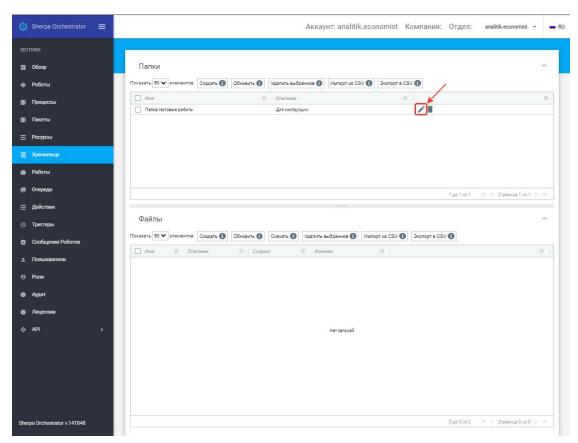


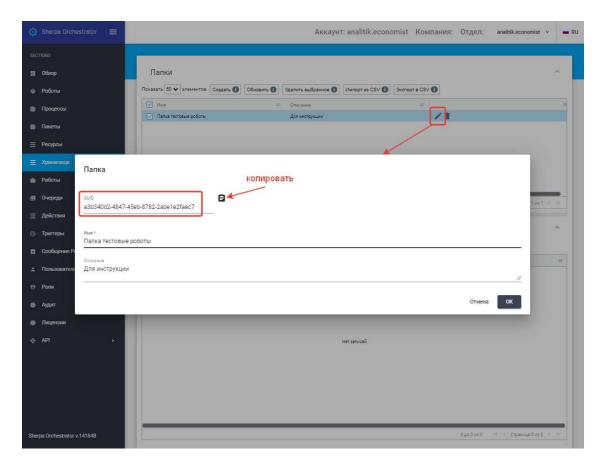
Перед созданием файла необходимо создать или выбрать из имеющихся в списке папку.



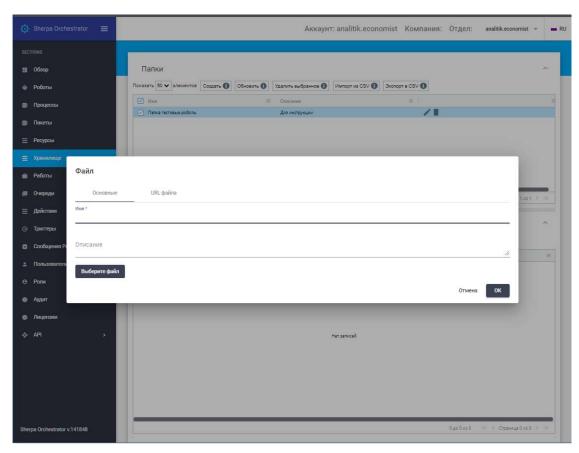


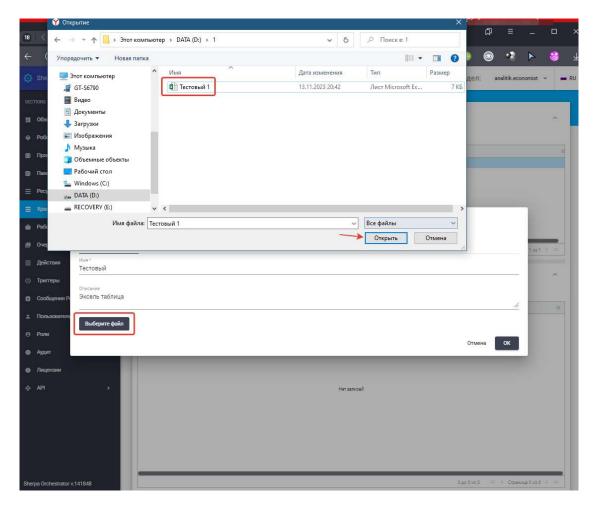




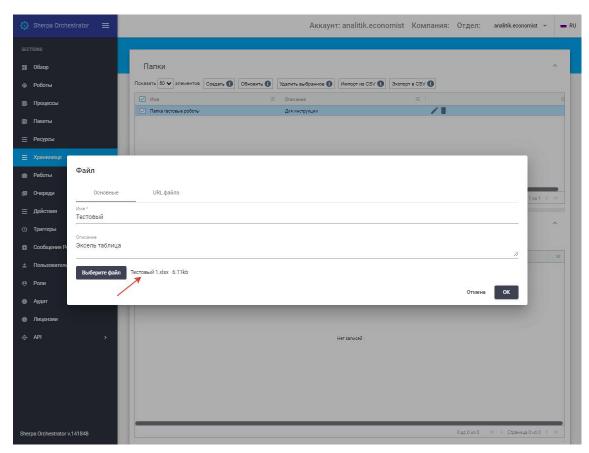


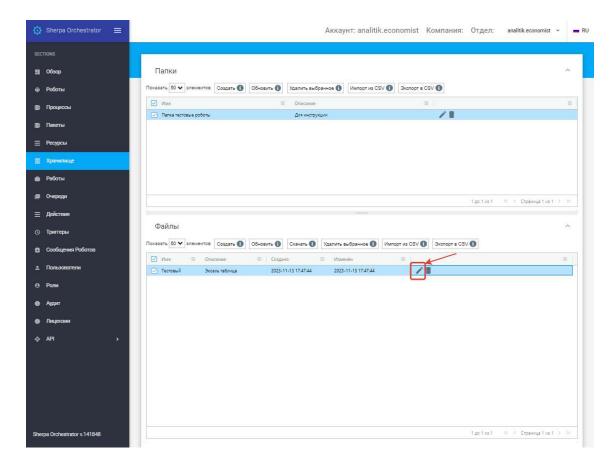




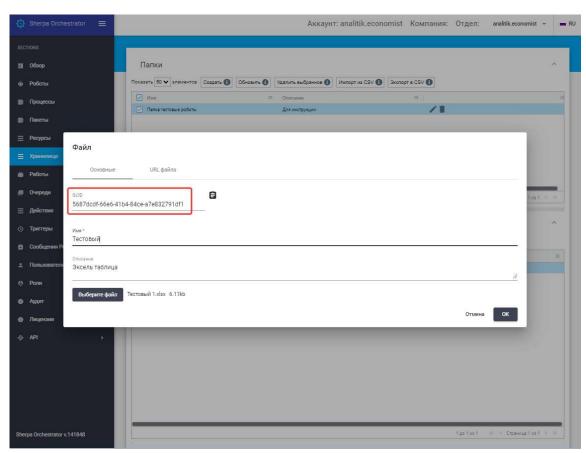


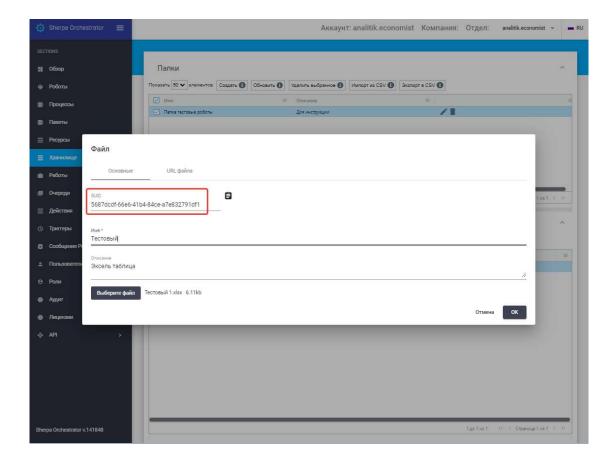




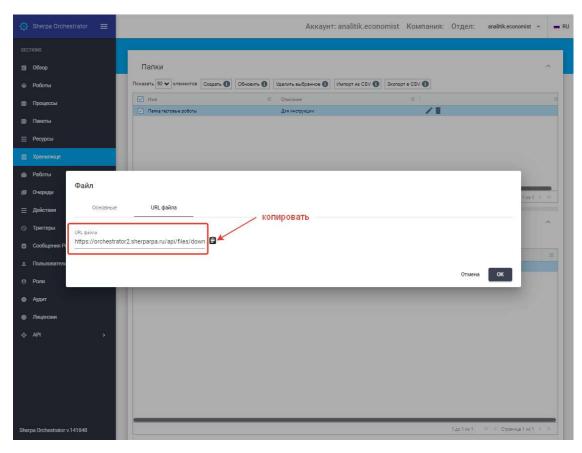


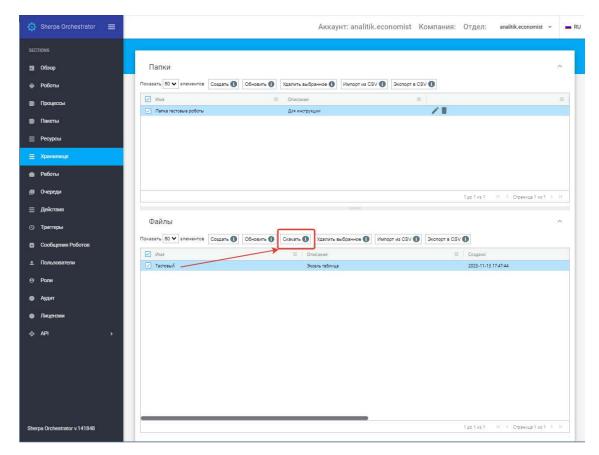




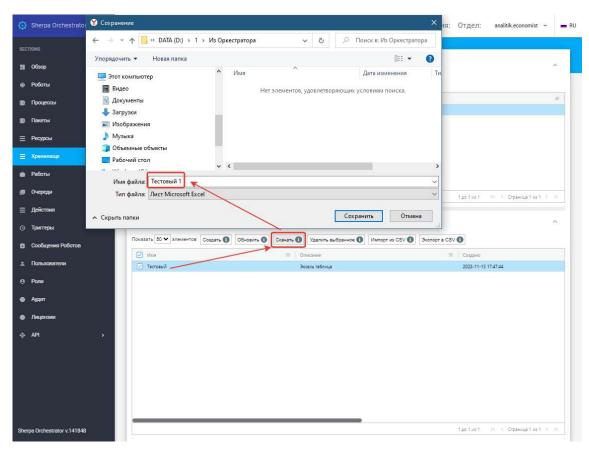




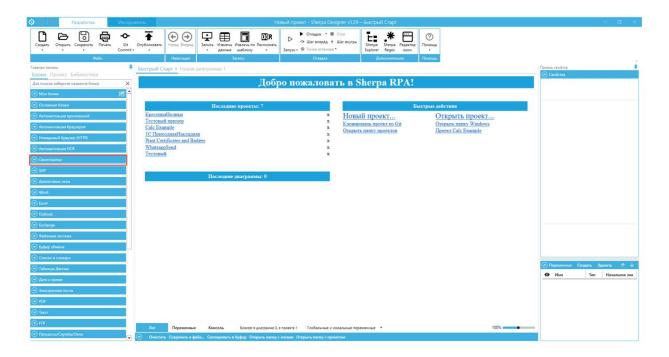




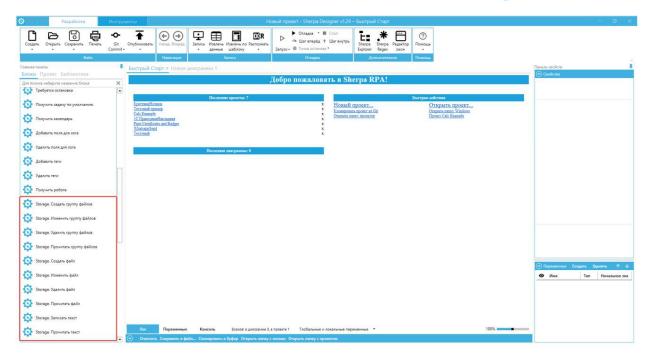




B Sherpa Designer есть отдельные блоки, которые работают с файлами. Они находятся в палитре Оркестратор.







Действия блоков представлены ниже:

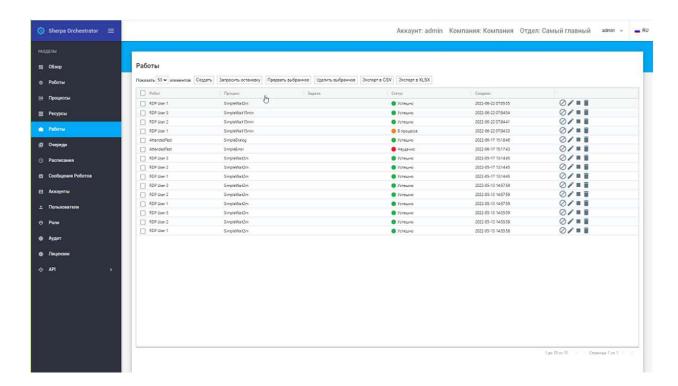




4.3.3.7 Экран «Работы»

Работа – это связка из робота и процесса либо робота, процесса и задачи. Работу можно создать вручную и с помощью очереди.

Экран «Работы» содержит таблицу «Работы».



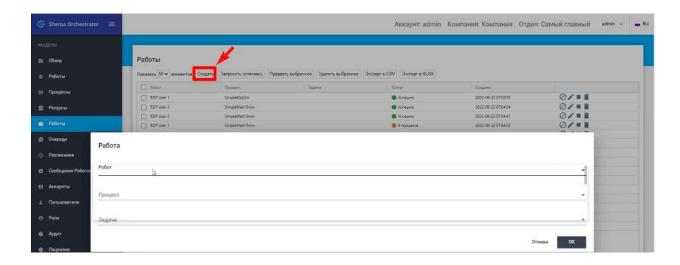
Таблица, кроме обычных действий, позволяет также выполнять следующие действия:

- Клонировать работу (копирование уже имеющейся работы);
- Запросить остановку («мягкая» остановка, позволяет завершить работу корректно);
- Прервать выбранное («жесткая» остановка остановка без дополнительных действий);
- Обслуживание (настройка времени и даты для остановки или прерывания выбранных работ).

Создание новой работы

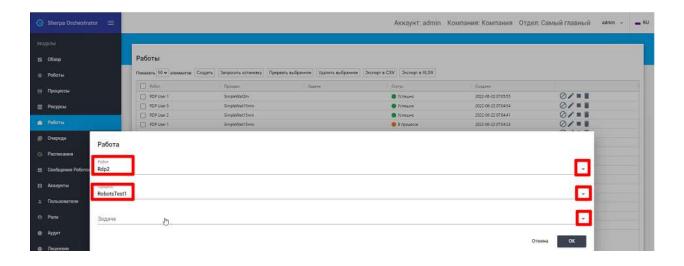


Для создания новой работы кликните на кнопку «Создать» в таблице «Работы» и заполните все поля карточки путем выбора из выпадающих списков: робот, процесс, задача.



Перед созданием Работы, должен быть создан Робот, Процесс и Задача. При заполнении полей доступен выбор только из уже созданных ранее Роботов, Процессов и Задач.

Задача – это опциональный параметр, т.е. Задачу можно не выбирать и оставить строку незаполненной.

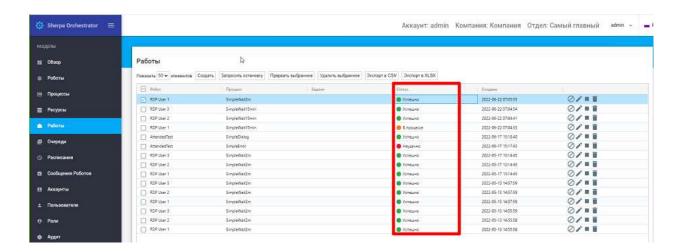


В случае, если вы выбираете задачу при создании работы (из ранее созданных задач в выпадающем списке), будут выбраны параметры, которые были для нее прописаны.



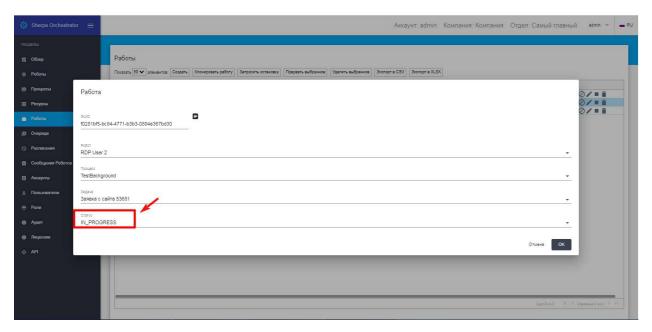


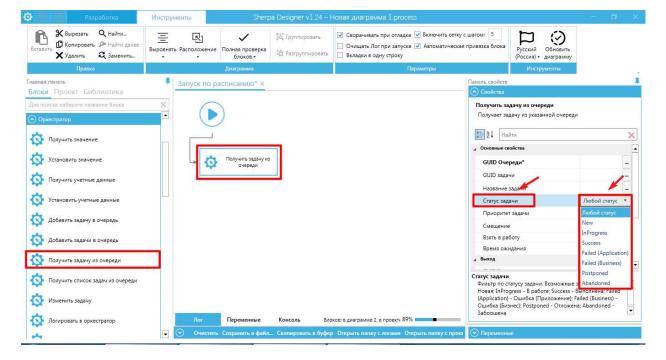
После нажатия на кнопку «ОК» в таблице «Работы» отобразится созданная вами работа в статусе «Ожидает». Как только освободится робот, для которого была назначена данная задача, робот возьмет задачу в работу, запустит сценарий соответствующего процесса, выполнит его и, в зависимости от результата работы, может меняться статус работы в колонке «Статус».



Статус задачи по умолчанию не меняется автоматически, эта опция настраивается на уровне созданного сценария в Дизайнере, т.е. вы сами определяете, будет ли меняться статус задачи и в какой момент.



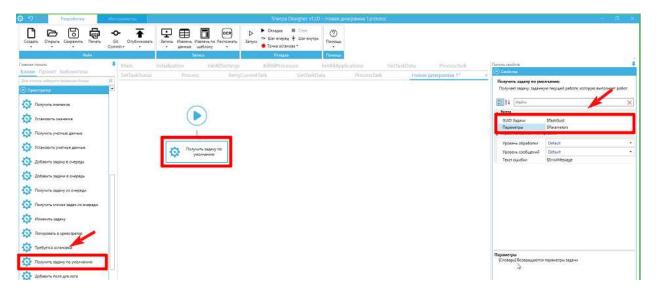




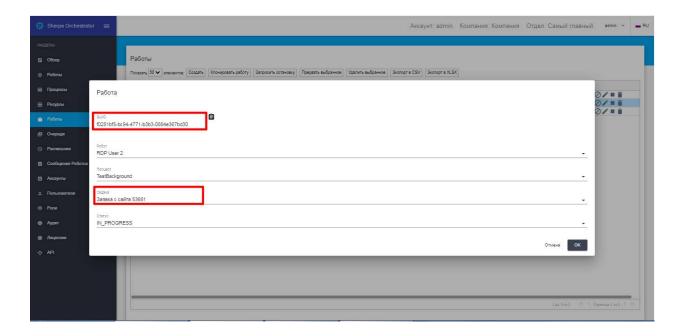
Существует возможность получить задачу по умолчанию, заданную в работе, которая привела к запуску данного Робота. Для этого используйте блок «Получить задачу по умолчанию» по умолчанию на стороне Дизайнера.

Данный блок позволяет получить параметры задачи по умолчанию и использовать их в сценарии робота в качестве входных параметров сценария.





Каждый раз, при запуске работы робота на вход робота поступают параметры и QUID задачи.



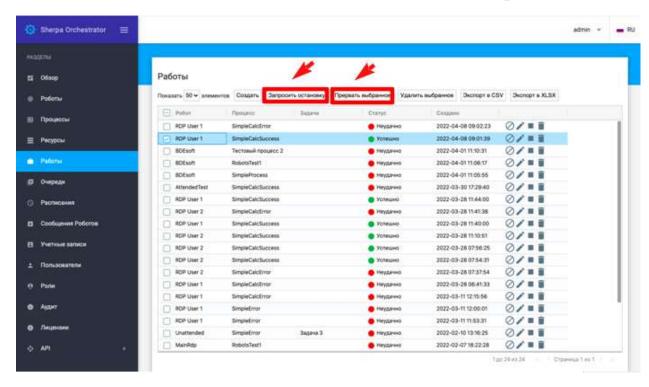
Остановить работу

При необходимости вы можете произвести остановку работающего в данный момент робота или работы из Оркестратора. Существует два варианта:

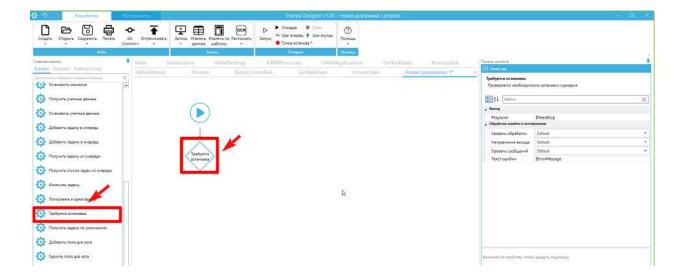
- Запросить остановку;
- Прервать выбранное.

Фактически это "мягкая остановка" и "жесткая остановка", соответственно.





При нажатии на кнопку «Запросить остановку», Оркестратор запрашивает выполняющийся сценарий робота о возможности остановки. Данный запрос может быть обработан внутри сценария с помощью блока Дизайнера «Требуется остановка».



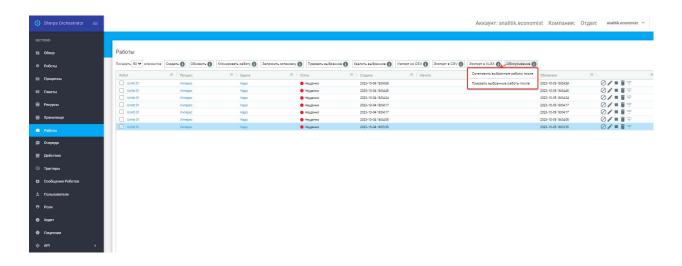
При нажатии на кнопку «Мягкая остановка», есть возможность решить – действительно останавливать процесс или нет. До фактической остановки можно сделать какие-либо действия, например, корректно сохранить и закрыть все файлы, чтобы ничего не повредить в сценарии.



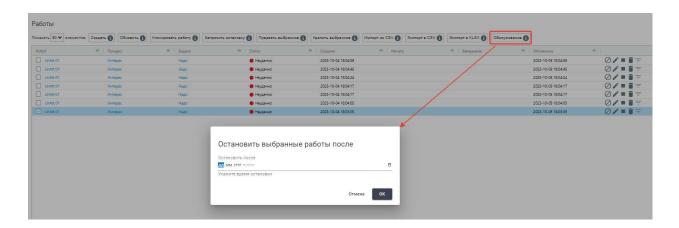
Жесткая остановка - безусловно прерывает сценарий на следующем действии.

Обслуживание работы

Пункт «Обслуживание» позволяет настроить параметры у выбранной Работы:

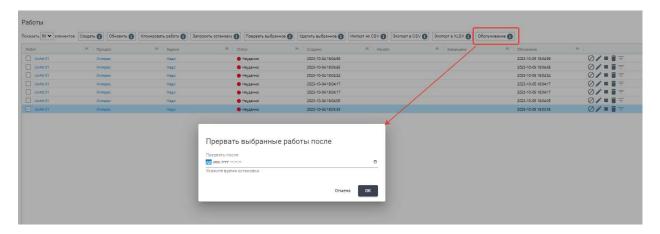


Остановить выбранные работы после указанной даты («мягкая остановка» после выбранной даты)

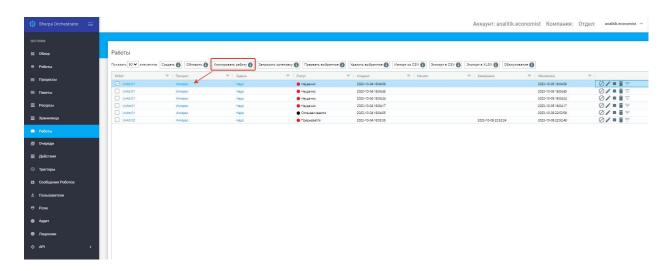


Прервать выбранные работы после указанной даты (безусловное прерывание выбранных работ)





Клонирование работы



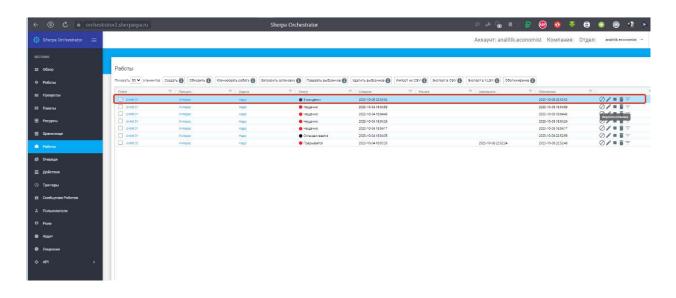
При необходимости создать работу, аналогичную имеющейся, т.е. с параметрами (робот, процесс, задача)





можно воспользоваться пунктом меню «Клонировать работу».

После нажатия на кнопку в списке создаться клон или дубль выбранной Работы с идентичными параметрами (робот, процесс, задача), но с другим GUID.





GUID 6699af28-0831-4909-9426-b449fdd88a36 Робот UnAtt 01 Процесс Интерес Задача Надо Статус В ожидании Остановить после дд.мм.гггг --:--:- Прервать после дд.мм.гггг --:--:--

Работа

4.3.3.8 Экран «Очереди»

Очереди – это способ группировать однотипные задачи, создавать потоки задач.

Можно рассматривать очереди как способ «хранения списка данных», которые требуется обработать.

Подобные группировки используются для того, чтобы внешняя система (или роботы) добавляли задачи в очередь, а другие роботы выполняли пункты очереди задач, то есть извлекали задачи из определённой очереди и делали работу.

Таким образом, длительный и сложный бизнес-процесс может быть разбит на набор подпроцессов, каждый из которых будет исполняться определенным роботом или группой роботов.



Очереди содержат набор Задач, при этом каждая задача — это набор параметров в виде пар «Параметр и Значение», которые могут указывать роботу, что именно ему нужно сделать в рамках выполнения процесса.

Задачи добавляются в очередь тремя способами:

- вручную пользователем с помощью Оркестратора;
- Роботом с помощью блока «Добавить задачу в очередь»;
- с помощью вызовов АРІ.

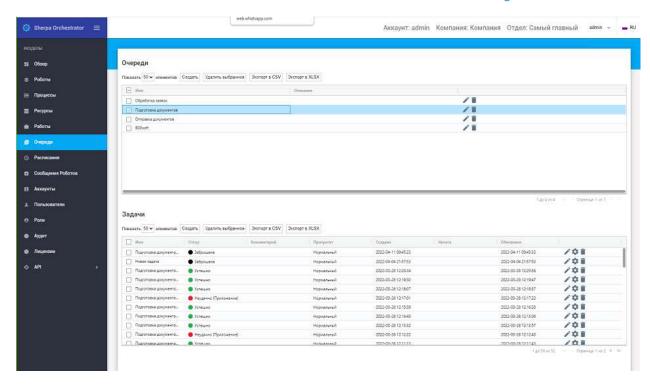
Добавлять новые задачи в очередь может как робот в процессе выполнения другой работы, так и какая-либо внешняя система.

Можно создавать неограниченное количество очередей, в каждой очереди находится поток задач и к каждой Задаче можно привязать словарь (hashtable) в виде набора пар «Ключ-Значение». При получении задачи из очереди вы получаете данный словарь и параметры и можете их обрабатывать. После обработки можно изменить статус задачи.

Интерфейс экрана «Очереди»

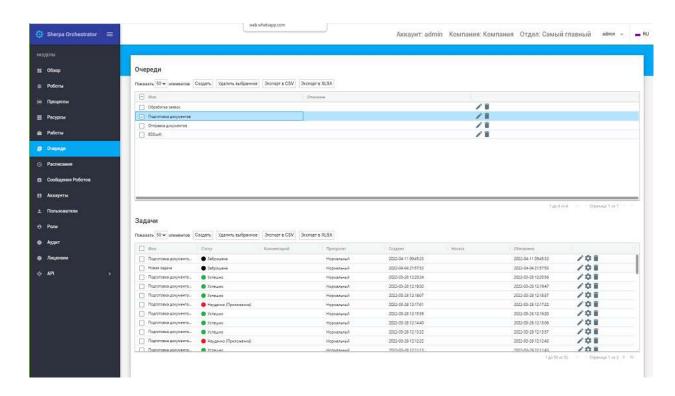
Экран «Очереди» содержит две таблицы: «Очереди» и «Задачи». Для того, чтобы в таблице задачи появился список задач, необходимо установить галочку у одной или нескольких задач.





Создание новой очереди

Для создания новой очереди кликните на кнопку «Создать» в таблице «Очереди».



и заполните все поля карточки:

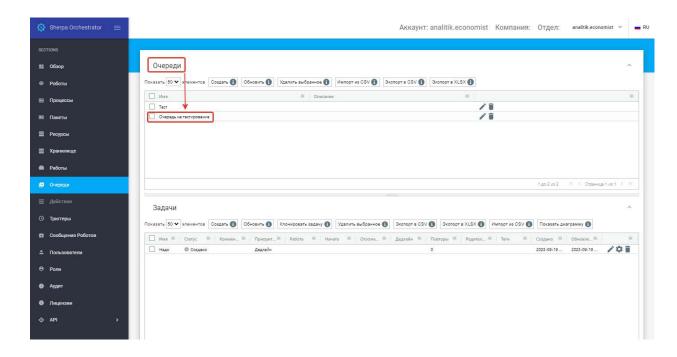


- Имя;
- Описание (не является обязательным).



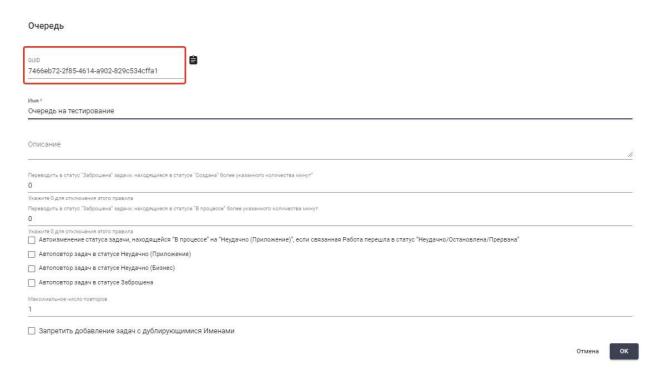
После заполнения наименования очереди и прочих параметров следует сохранить данные, нажав «ОК».

Новая очередь отобразится на экране в списке.

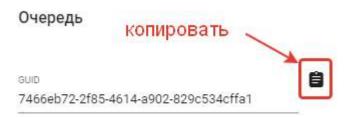


Обратите внимание! После создания любого элемента ему присваивается GUID – уникальный номер, который можно использовать в дальнейшей работе.





Для удобства рядом с GUID расположена иконка, при нажатии на которую копируется в буфер обмена текущее значение из поля GUID.



Настройки окна «Очередь»



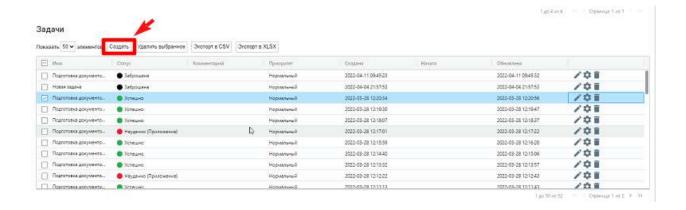
Очередь	
Maan *	
Описание	
Переводить в статус "Заброшена" задачи, находящиеся в статусе "Создана" более указанного количества минут". О	
Укажите 0 для отключения этого правила	
Переводиты в статус "Заброшена" задачи, находящиеся в статусе "В процессе" болев указанного количествя минут. О	
Укажите 0 для отключения этого правила — Автоизменение статуса задачи, находящейся "В процессе" на "Неудачно (Приложение)", если связанная Работа перешла в статус "Неудачно/Остановлена/Прерв	зана"
Автоповтор задач в статусе Неудачно (Приложение)	
Автоповтор задач в статусе Неудачно (Бизнес)	
🔲 Автоповтор задач в статусе Заброшена	
Максимальное число повторов	
1	

Заполнить необходимые параметры задачи.

Создание новой задачи

Для создания новой задачи кликните на кнопку «Создать» в таблице «Задачи» и заполните все поля карточки:

- Имя;
- Описание;
- Комментарий;
- Приоритет.



Поля «Описание» и «Комментарий» не являются обязательными. Также вы можете указать Приоритет задачи (низкий, нормальный, высокий). 284



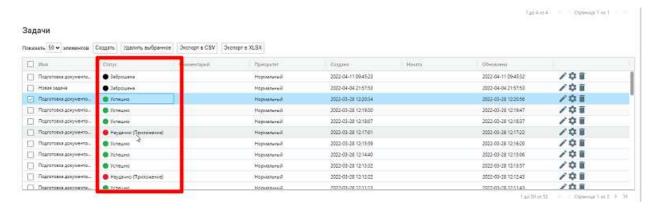
Задача	
Mua "	
Описание	in the second se
Комментарий	
Tiphicolities PRIORITY_NORMAL	*
	Отмена ОК

Статусы

Задачи могут иметь различные статусы:

- Создана (New) новая задача;
- В процессе (InProgress) в работе. Находится в процессе работы робота;
 - Успешно (Success) выполнена;
- Неудачно (Приложение) Failed (Application) ошибка (Приложение). Статус присваивается задачам, если возникшая проблема может быть решена при повторном выполнении задачи (например, при зависании системы, с которой работает робот);
- Неудачно (Бизнес) Failed (Business) ошибка (Бизнес). Используется, когда автоматическая попытка сигнализирует о проблемах с входными данными или со статусами внешних информационных систем, с которыми работает робот, и повторное выполнение задачи ожидаемо будет безрезультатным;
- Отложена до Postponed отложена. Статус автоматически устанавливается из сценария Дизайнера в случае, если мы хотим отложить задачу до определенного времени, чтобы Оркестратор в течение этого времени вернул эту задачу обратно в очередь;
- Заброшена (Abandoned) статус проставляется, если задача была взята в работу, но в течение более 1 суток она не была выполнена.





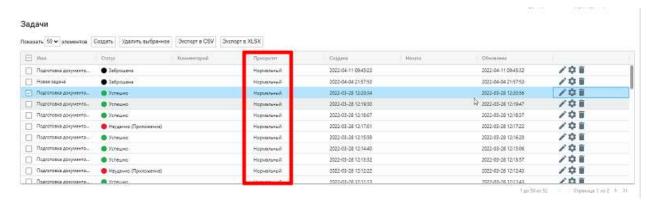
Статус задачи может быть установлен в момент создания новой задачи или изменен для уже существующей задачи.



При постановке задачи из очереди статус задачи должен быть изменен на «InProgress». Статус задачи не назначается автоматически. Иногда не требуется автоматически проставлять статус «InProgress», например, если вы хотите только посмотреть на задачу и определить, должен ли конкретный робот заниматься той или иной задачей.

Приоритеты





Каждая Задача имеет Приоритет:

Задача

- Низкий
- Нормальный
- Высокий
- Дедлайн

Укажите имя Описание Комментарий Приоритет Нормальный Низкий Нормальный Базовые приоритеты Высокий Дедлайн

Приоритет может быть установлен как в момент создания новой задачи, так и для существующей задачи.



Приоритет влияет на то, в каком порядке задачи будут получены роботом. Первыми робот получит задачи с высоким приоритетом, затем с нормальным, далее с низким.

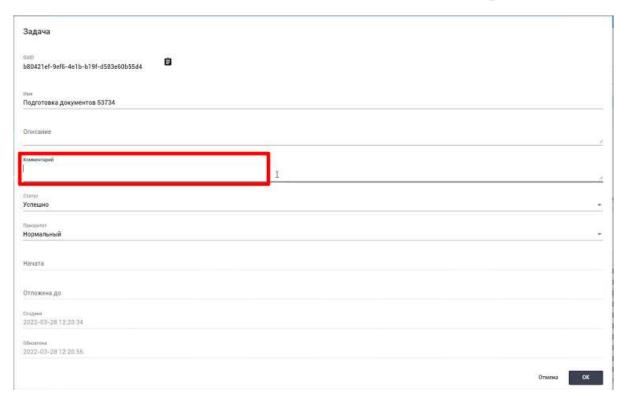
Приоритет «Дедлайн» устанавливается у задачи, у которой установлен параметр дедлайн (дата крайнего срока), и дедлайн-дата уже подошла. Такая задача будет выполняться с наивысшим приоритетом.

адача	
ino É 448b135-8751-4818-8dc6-2f0ee2242820	
MR * BAO	
писание	11
омментарий	h
уатус оздано	•
рикритет едлайн	•
еги	
ачата	
A.MM.FFFF -:-	
ALAMATETT — (:	0
Отмена	OK

Комментарии

Для каждой задачи можно создать комментарий. Комментарий может быть указан на любом этапе, например, в момент создания задачи. Также мы можем сделать так, чтобы робот изменил комментарий в процессе выполнения задачи.





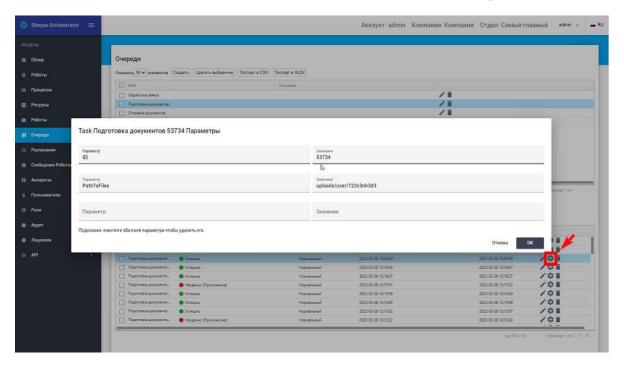
Эту опцию удобно использовать в процессе логирования, чтобы объяснить причину невыполнения той или иной задачи.

Параметры

Для каждой задачи может быть установлен набор параметров (пара Параметр/Значение). Благодаря этой опции мы можем передавать суть задачи, например, мы можем указать ID документа, с которым должен работать робот, а также указать путь к папке, в которой расположен документ.

Для этого в таблице задачи нажимаем на кнопку «Настройки». В появившемся окне заполняем необходимые параметры.





К задачам из сценария обращаются по GUID, а не по названию, которое указано в Оркестраторе.

В конце выполнения задачи ее статус нужно снова изменить, в зависимости от того, была она завершена успешно или неуспешно.

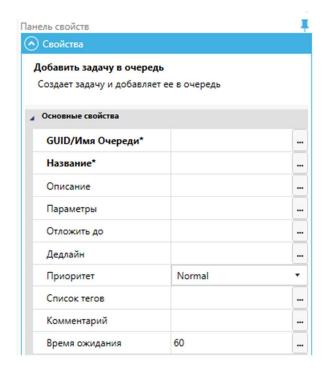
Пока статус задачи не изменен на «InProgress», ее может взять любой робот, но со статусом «InProgress» задача становится «забронирована» за конкретным роботом.

4.3.3.8.1 Работа с Задачами в Sherpa Designer

На стороне Дизайнера работа с задачами осуществляется через следующие блоки палитры Оркестратора:

1) Блок «Добавить задачу в очередь».





При использовании блока необходимо добавить GUID, который нужно скопировать из карточки соответствующей очереди.

Необходимо указать:

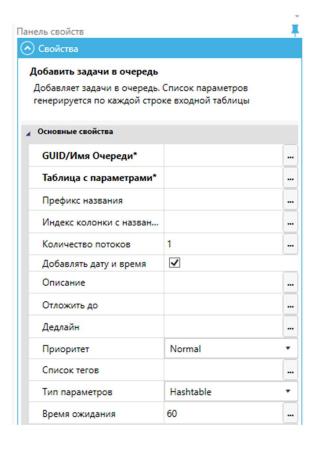
- Название (обязательное поле);
- Описание (опционально);
- Параметры (тип данных параметров словарь);
- Приоритет.

После заполнения данных полей на выходе будет сформирован QUID Задачи и Объект Задачи типа \$TaskObject. С данным объектом вы сможете работать в любом блоке, выражении, для этого воспользуйтесь подсказками в окне «Редактировать выражение» – «Примеры работы с объектами Задачи \$TaskObject».



2) Блок «Добавить задачи в очередь».



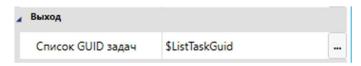


Необходимо указать:

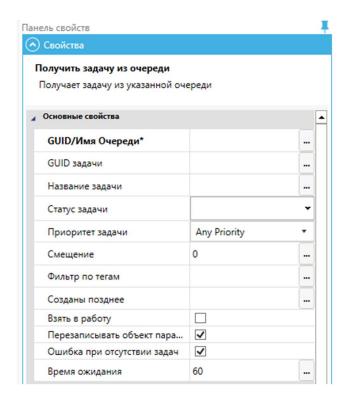
- QUID Очереди (обязательное поле);
- Префикс названия позволяет приписывать создаваемым задачам одинаковые префиксы в названии задачи. В случае указания непустого префикса, каждая созданная данным блоком задача будет иметь название, начинающееся на этот префикс, и к нему будет приписаны текущие дата-время.
 - Описание описание задачи (опционально)
- Таблица с параметрами [Таблица Данных] Таблица, по каждой строке которой будет создана новая Задача
 - Приоритет приоритет задачи. Возможные значения:
 - Low низкий;
 - Normal нормальный;
 - High высокий

На выходе: Список GUID созданных задач





3) Блок «Получить задачу из очереди».



Необходимо указать:

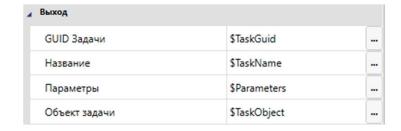
- GUID Очереди
- GUID Задачи (необязательно, если вы хотите получить не конкретную, а какую-либо очередную Задачу, то поле можно оставить пустым)
 - Название Задачи
- Статус Задачи (например, если необходимо получать только новые задачи New или в каком-либо другом статусе)
- Приоритет Задачи по желанию можно установить фильтр (Low, Normal, Hight). При выборе любого приоритета задачи будут поступать начиная с высокого приоритета, заканчивая низким.
- Смещение используется в случае, если вы хотите получить из Очереди не первую Задачу, а n-ую по счету (5, 10 и т.д.)
- Взять в работу позволяет получить задачу из очереди и сразу взять ее в Работу по принципу атомарной операции. Опция полезна в случае, если 293



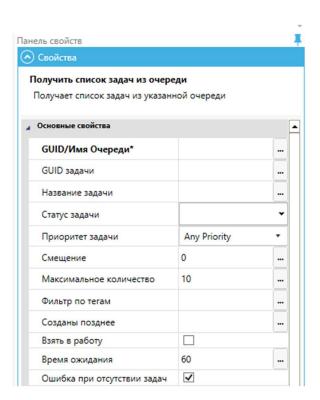
одновременно несколько роботов заканчивают работу с предыдущей Задачей и пытаются взять в работу следующую задачу из очереди. Наличие галочки «Взять в работу» гарантирует, что определенная Задача будет взята в работу конкретным Роботом. Если галочка не была проставлена и сложилась описанная выше ситуация, то в Дизайнере появится уведомление об ошибке.

На выходе будут получены:

- GUID Задачи;
- Название;
- Параметры;
- Объект задачи.



4) Блок «Получить список задач из очереди».



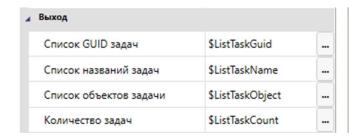


- GUID Очереди (обязательно);
- GUID Задачи (необязательно, если вы хотите получить не конкретную, а какую-либо очередную задачу, то поле можно оставить пустым);
 - Название Задачи фильтр для названия задачи;
- Статус Задачи фильтр по статусу Задачи (если необходимо получать только новые задачи New или в каком-либо другом статусе). Возможные значения:
- New Новая;
- InProgress В работе;
- Success Выполнена;
- Failed (Application) Ошибка (Приложение);
- Failed (Business) Ошибка (Бизнес);
- Postponed Отложена;
- Abandoned Заброшена.
- Приоритет задачи по желанию можно установить фильтр (Low, Normal, Hight). При выборе любого приоритета задачи будут поступать, начиная с высокого приоритета, заканчивая низким;
- Смещение смещение относительно начала очереди. Используется в случае, если вы хотите получить из очереди не первую задачу, а n-ую по счету (5, 10 и т.д.);
- Максимальное количество возвращается не больше указанного количества задач. Если параметр не указан, возвращается не больше 100 задач;
- Взять в работу если выбрано, то статус задач будет сразу изменен на статус «В работе». Позволяет получить задачу из очереди и сразу взять ее в работу по принципу атомарной операции. При использовании блока «Получить список задач из очереди» не рекомендуется использовать данную функцию.

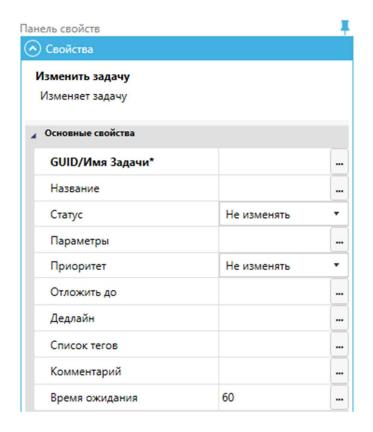
На выходе будут получены:

- Список GUID Задач;
- Список Названий Задач;
- Список Объектов Задач.





5) Блок «Изменить задачу».



Необходимо указать:

- GUID Задачи;
- Название Задачи фильтр для названия задачи;
- Статус Задачи фильтр по статусу задачи. Возможные значения:
- New Новая;
- InProgress В работе;
- Success Выполнена;
- Failed (Application) Ошибка (Приложение);
- Failed (Business) Ошибка (Бизнес);
- Postponed Отложена;

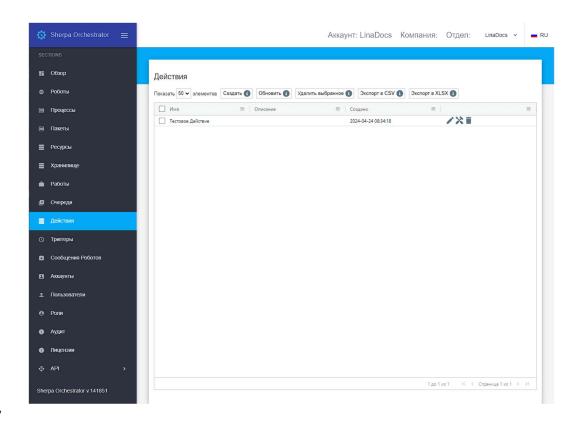


- Abandoned Заброшена;
- Параметры (тип данных Параметров словарь).
- Приоритет задачи по желанию можно установить фильтр (Low, Normal, Hight). При выборе любого приоритета задачи будут поступать начиная с высокого приоритета, заканчивая низким;
- Отложить до [Дата и время/Текст] обработка задачи будет отложена до указанного момента, при этом статус задачи будет установлен в Postponed (отложена). После наступления этого момента оркестратор автоматически переведет статус Задачи в New (Новая);
 - Комментарий комментарий задачи.
- Время ожидания укажите время ожидания выполнения задачи в секундах.

4.3.3.9 Экран «Действия»

Экран «Действия» позволяет создавать отдельные рабочие места внутри Оркестратора. С помощью экрана «Действия» можно создавать действия для задач в очереди, а также форму для выполнения задачи.

Экран «Действия» содержит таблицу «Действия».





Создание нового действия

Для создания нового действия необходимо нажать на кнопку «Создать» в таблице «Действия» и заполнить открывшуюся форму.

Форма создания нового действия содержит две вкладки:

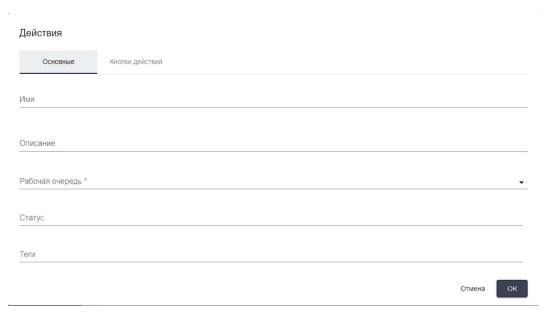
- Основные;
- Кнопки действий.

Вкладка «Основные»

Элементы интерфейса Вкладки «Основные».

№ п/п	Элемент интерфейса	Описание
1.	текстовое поле «Имя»	Имя действия.
2.	текстовое поле «Описание»	Описание действия.
3.	выпадающий список «Рабочая очередь»	Позволяет выбрать из выпадающего списка ранее созданную очередь. Обязательное для заполнения поле.
4.	поле «Статус»	Статус задач из очереди, которые будут доступны для изменения в рамках создаваемого действия. . Возможные варианты: . Создано; . В процессе; . Успешно; . Неудачно (Приложение); . Неудачно (Бизнес); . Отложена до; . Заброшена.
5.	текстовое поле «Теги»	Теги для выбора тех задач из очереди, которые будут доступны для изменения в рамках создаваемого действия.





Вкладка «Кнопки действий»

Во вкладке «Кнопки действий» можно создать кнопки для задачи в указанной во вкладке «Основные» очереди. После перехода на вкладку «Кнопки действий» форма создания будет свернута. Чтобы ее развернуть, необходимо нажать .



Форма создания кнопки действия содержит следующие флаги и поля:

Элементы интерфейса Вкладки «Кнопки действий».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	кнопка	Позволяет удалить кнопку действия.

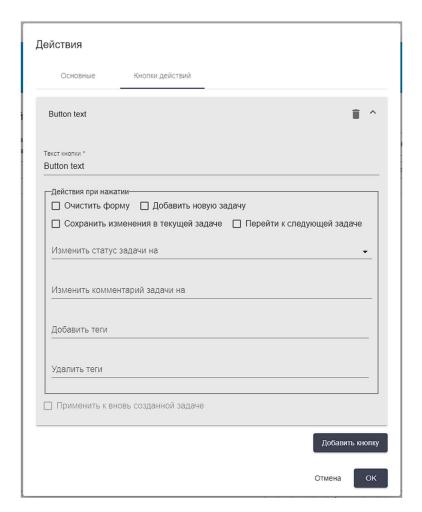


2.	текстовое поле «Текст кнопки»	Текст кнопки. Обязательное для заполнения поле.
3.	поле «Действия при нажатии»	Позволяет настроить действия при нажатии кнопки.
3.1.	флаг «Очистить форму»	Позволяет очищать форму при нажатии кнопки.
3.2.	флаг «Добавить новую задачу»	Позволяет добавить новую задачу при нажатии кнопки.
3.3.	флаг «Сохранить изменения в текущей задаче»	Позволяет сохранить изменения в текущей задаче при нажатии кнопки.
3.4.	флаг «Перейти к следующей задаче»	Позволяет перейти к следующей задаче при нажатии кнопки.
3.5.	выпадающий список «Изменить статус задачи на»	Позволяет изменить статус задачи при нажатии кнопки. Возможные значения:
3.6.	текстовое поле «Изменить комментарий задачи на»	Позволяет изменить комментарий задачи при нажатии кнопки на указанный в данном текстовом поле.
3.7.	текстовое поле «Добавить теги»	Позволяет указать теги, которые будут добавлены к выбранной Задаче при нажатии кнопки.
3.8.	текстовое поле «Удалить теги»	Позволяет указать теги, которые будут удалены из выбранной Задачи при нажатии кнопки.



флаг «Применить к вновь созданной задаче»

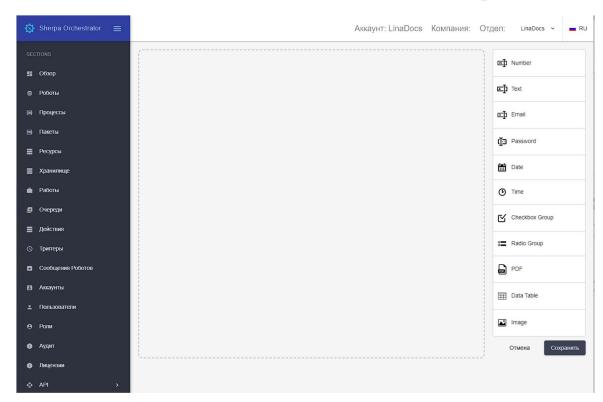
Позволяет применить указанные в поле «Действия при нажатии» к вновь созданной задаче.



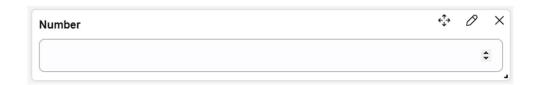
Дизайнер формы

Для каждого созданного действия по кнопке хароступен «Дизайнер формы». После нажатия кнопки «Дизайнер формы» Оркестратор открывает окно редактора формы.





Кнопка позволяет добавить в форму действия новый элемент - поле для ввода номера.



Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку ${\cal O}$, после чего откроется окно настройки элемента.

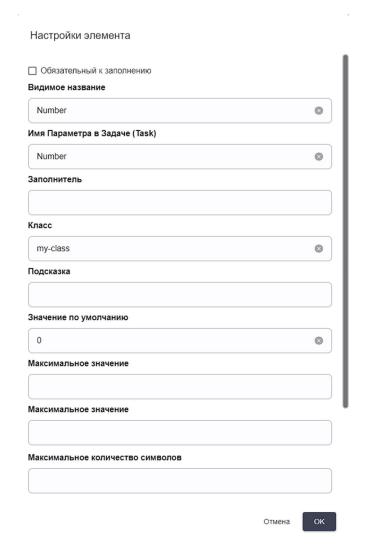
Элементы интерфейса «Дизайнер формы» Настройка элемента «Number».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.



3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в Задаче, откуда будет считываться и куда будет сохранятся информация данного элемента.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля. Подсказка будет отображаться внизу под полем.
7.	текстовое поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для настраиваемого поля.
8.	поле «Минимальное значение»	Позволяет указать минимальное значение, которое можно ввести в поле.
9.	поле «Максимальное значение»	Позволяет указать максимальное значение, которое можно ввести в поле.
10.	поле «Максимальное количество символов»	Позволяет указать максимальное количество символов, которое можно ввести в поле.





Кнопка позволяет добавить в форму действия новый элемент - поле для ввода текста.



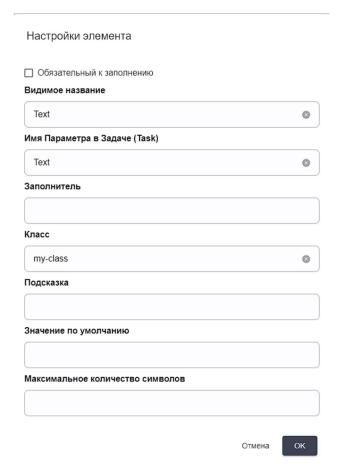
Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку ${\cal O}$, после чего откроется окно настройки элемента.



Элементы интерфейса «Дизайнер формы» Настройка элемента «Text».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.
3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
7.	текстовое поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для настраиваемого поля.
8.	поле «Максимальное количество символов»	Позволяет указать максимальное количество символов, которое можно ввести в поле.





Кнопка позволяет добавить в форму действия новый элемент – поле для ввода электронной почты.



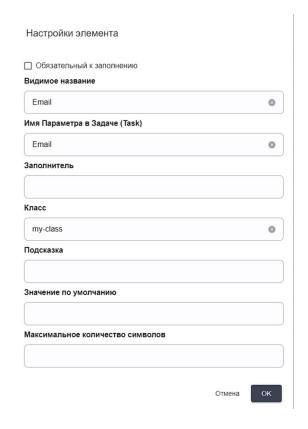
Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку ${\cal O}$, после чего откроется окно настройки элемента.



Элементы интерфейса «Дизайнер формы» Настройка элемента «Email».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.
3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например, для того чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
7.	текстовое поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для настраиваемого поля.
8.	поле «Максимальное количество символов»	Позволяет указать максимальное количество символов, которое можно ввести в поле.





Кнопка Раssword позволяет добавить в форму действия новый элемент – поле для ввода пароля.



Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку $^{\mathcal{O}}$, после чего откроется окно настройки элемента.

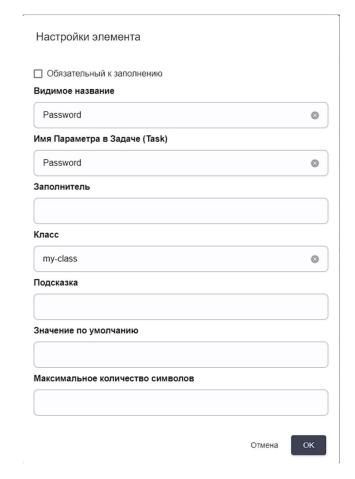
Элементы интерфейса «Дизайнер формы» Настройка элемента «Password».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.

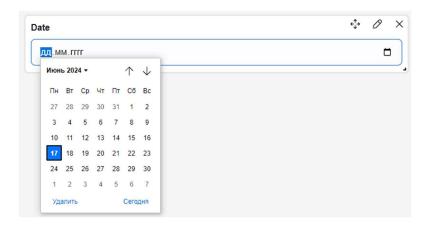


3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
7.	текстовое поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для настраиваемого поля.
8.	поле «Максимальное количество символов»	Позволяет указать максимальное количество символов, которое можно ввести в поле.





Кнопка — Date позволяет добавить в форму действия новый элемент – поле для ввода даты, также дату можно выбрать с помощью календаря.



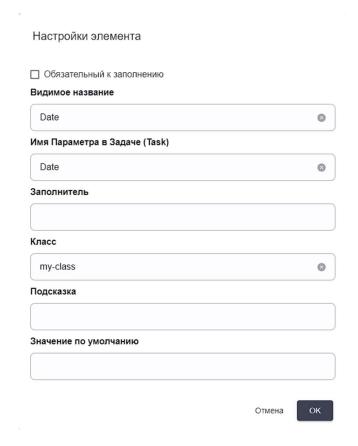
Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку ${\cal O}$, после чего откроется окно настройки элемента.



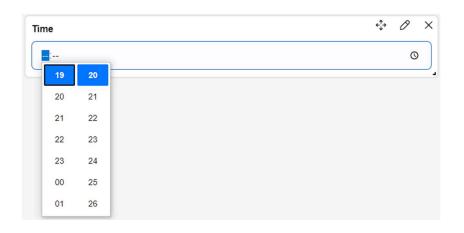
Элементы интерфейса «Дизайнер формы» Настройка элемента «Date».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.
3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
7.	текстовое поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для настраиваемого поля.





Кнопка позволяет добавить в форму действия новый элемент - поле для ввода времени, также дату можно выбрать с помощью часов.



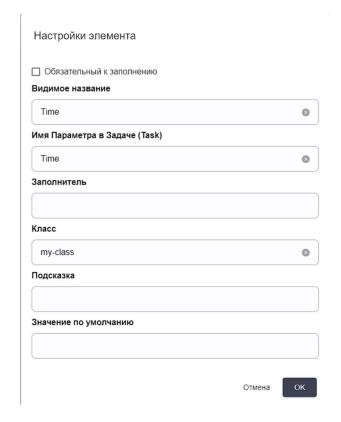
Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку ${\cal O}$, после чего откроется окно настройки элемента.



Элементы интерфейса «Дизайнер формы» Настройка элемента «Тіте».

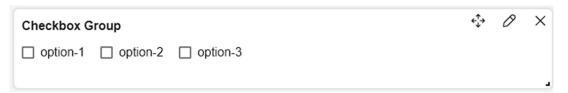
№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.
3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
7.	текстовое поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для настраиваемого поля.





Checkbox Group

Кнопка позволяет добавить в форму действия новый элемент - поле с одним или несколькими флагами.



Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку $^{\it O}$, после чего откроется окно настройки элемента.

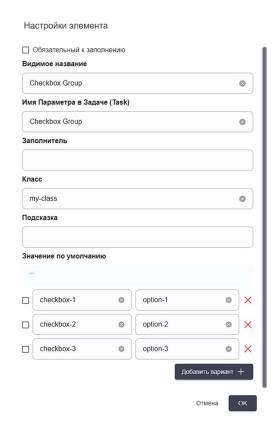
Элементы интерфейса «Дизайнер формы» Настройка элемента «Checkbox Group».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.



3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
7.	поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для каждого из флагов.
7.1	текстовое поле «checkbox»	Позволяет указать подпись для каждого из флагов.
7.2.	текстовое поле «option»	Позволяет указать опцию, которую выполняет каждый из флагов, в случае их установки.
8.	кнопка «Добавить вариант»	Позволяет добавить новую пару флагопция.





Кнопка Radio Group позволяет добавить в форму действия новый элемент - поле с одним или несколькими переключателями.



Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку $^{\mathcal{O}}$, после чего откроется окно настройки элемента.

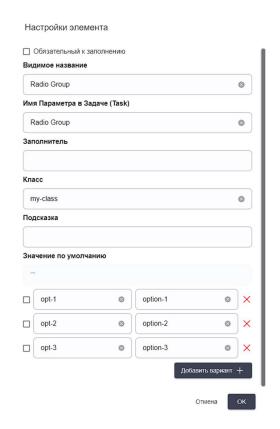
Элементы интерфейса «Дизайнер формы» Настройка элемента «Radio Group».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.



3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
7.	поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для каждого из переключателей.
7.1	текстовое поле «opt»	Позволяет указать подпись для каждого из переключателей.
7.2.	текстовое поле «option»	Позволяет указать опцию, которую выполняет каждый из переключателей, в случае их установки.
8.	кнопка «Добавить вариант»	Позволяет добавить новую пару переключатель-опция.





Кнопка позволяет добавить в форму действия новый элемент – поле для загрузки PDF-файла.



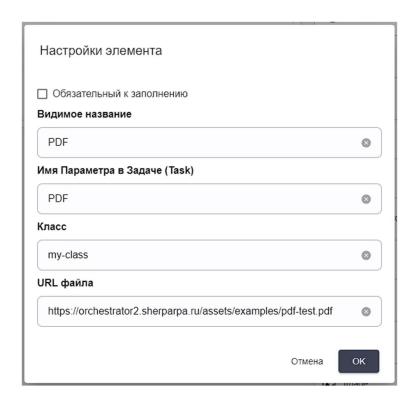
Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку ${\cal O}$, после чего откроется окно настройки элемента.

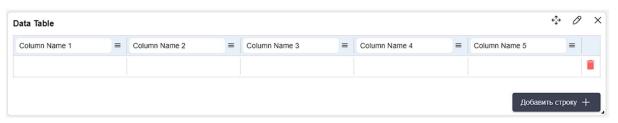
Элементы интерфейса «Дизайнер формы» Настройка элемента «PDF».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.



2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.
3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
5.	текстовое поле «URL файла»	Позволяет указать URL загружаемого PDF- файла (из Хранилища Оркестратора или из сети Интернет)





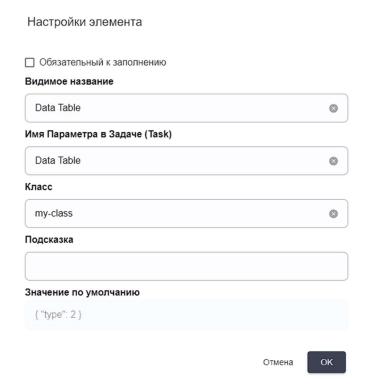


Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку $^{\it O}$, после чего откроется окно настройки элемента.

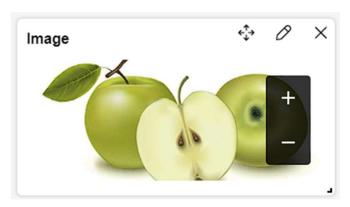
Элементы интерфейса «Дизайнер формы» Настройка элемента «Data Table».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.
2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.
3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
5.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
6.	поле «Значение по умолчанию»	Позволяет указать значение по умолчанию для настраиваемого поля.





Кнопка позволяет добавить в форму действия новый элемент – поле для загрузки изображения.



Затем необходимо настроить новый элемент формы. Для этого необходимо нажать кнопку $^{\it O}$, после чего откроется окно настройки элемента.

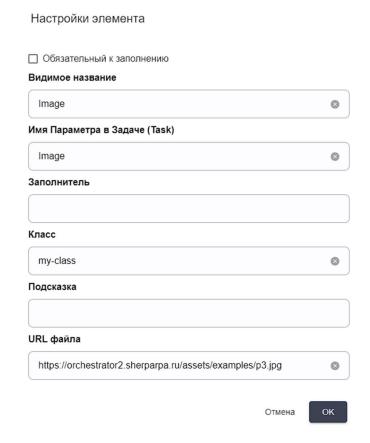
Элементы интерфейса «Дизайнер формы» Настройка элемента «Ітаде».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Обязательный к заполнению»	Позволяет сделать настраиваемое поле обязательным для заполнения.



2.	текстовое поле «Видимое название»	Позволяет указать название настраиваемого поля.
3.	текстовое поле «Имя Параметра в Задаче (Task)»	Позволяет указать имя параметра в задаче.
4.	текстовое поле «Заполнитель»	Позволяет указать placeholder (подсказку внутри поля, которая будет отображаться в момент, когда поле не заполнено).
5.	текстовое поле «Класс»	Позволяет указать класс, который будет добавлен элементу данного поля в HTML-коде страницы. Нужен, например для того, чтобы можно было кодом найти данное поле по селектору.
6.	текстовое поле «Подсказка»	Позволяет указать подсказку для настраиваемого поля.
7.	текстовое поле «URL файла»	Позволяет указать URL загружаемого изображения (из Хранилища Оркестратора или из сети Интернет).

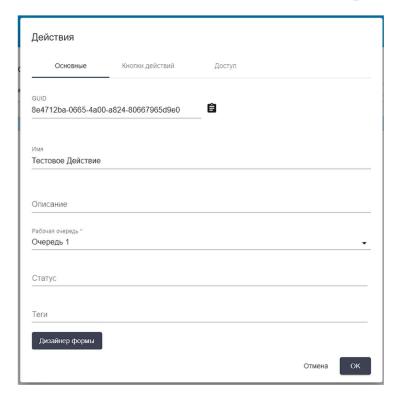




Редактирование ранее созданного действия

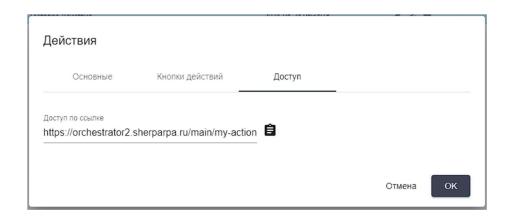
Для просмотра и редактирования ранее созданного действия необходимо выбрать его в списке и нажать на кнопку . После этого откроется форма с параметрами выбранного действия, в которую можно внести необходимые изменения. Помимо полей, заполняемых при создании нового действия, при редактировании в форме указан GUID (уникальный идентификатор, присвоенный действию после создания, его нельзя отредактировать), а также дополнительная вкладка «Доступ».





Вкладка «Доступ»

Во вкладке «Доступ» в поле «Доступ по ссылке» указана ссылка, по которой конечный пользователь попадет в рабочее пространство, созданное в рамках это действия. Данное поле нельзя отредактировать.



4.3.3.9.1 Пример создания действия и его настройки

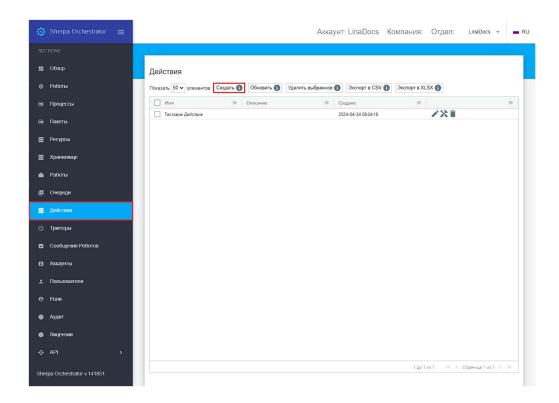
В данном примере описан вариант взаимодействия роботов и сотрудников, работающих на рабочем месте через экран «Действия».

Сначала один из роботов получает и распознает накладные, а затем создает Задачи по итогам распознавания и устанавливает в них теги 1CNew, после чего



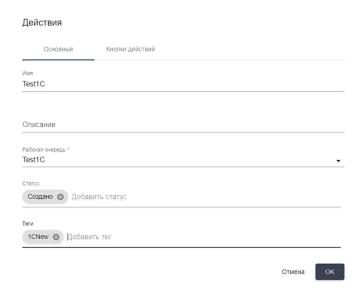
производится проверка правильности распознавания (и исправлен в случае необходимости) данных на рабочем месте, созданном в рамках действия, после чего в задаче меняется тег с 1CNew на 1CModified. Соответственно, другой робот будет забирать задачи с тегом 1CModified и вносить данные, которые в них есть в 1C.

Откройте экран «Действия» и нажмите кнопку «Создать».

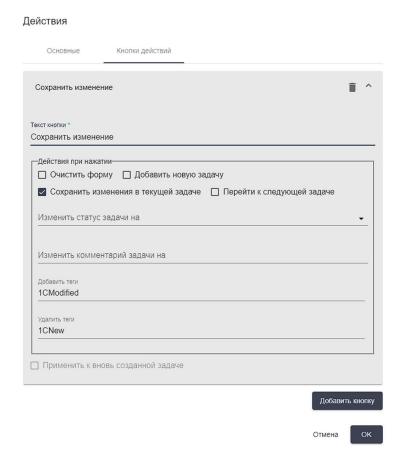


В открывшейся форме укажите название, рабочую очередь (необходимо создать ее заранее), статус задач и тег.



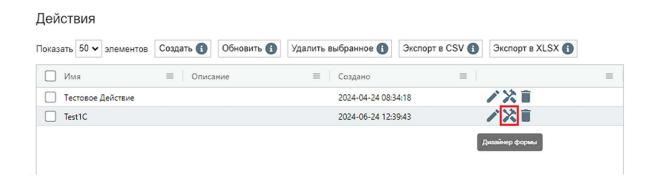


Перейдите во вкладку «Кнопки действий». Для первой кнопки укажите название и установите флаг «Сохранить изменения в текущей задаче». Укажите тег, который нужно добавить и тег, который нужно удалить в соответствующих текстовых полях.

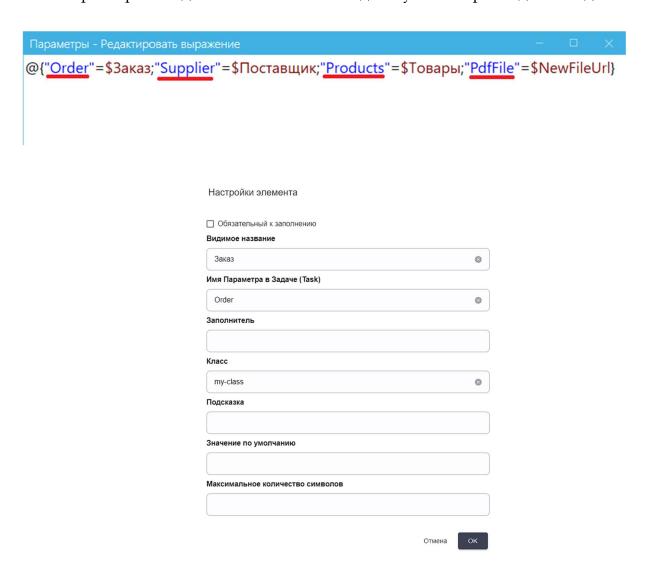




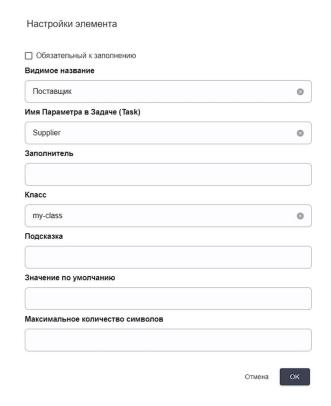
Сохраните и закройте форму, нажав кнопку «ОК». Затем перейдите в «Дизайнер формы» у только что созданного действия, с помощью кнопки $\overline{\lambda}$.



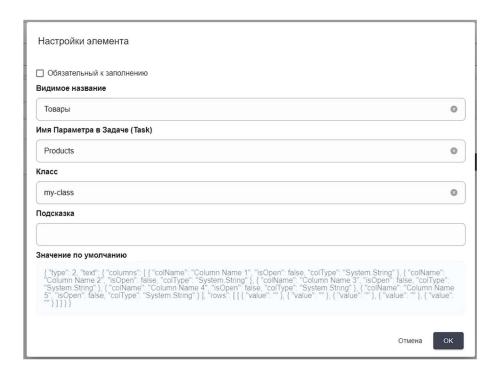
В «Дизайнере формы» добавьте два текстовых поля (Text) и укажите в них имена параметров в задаче. Эти имена необходимо указать при создании задачи.





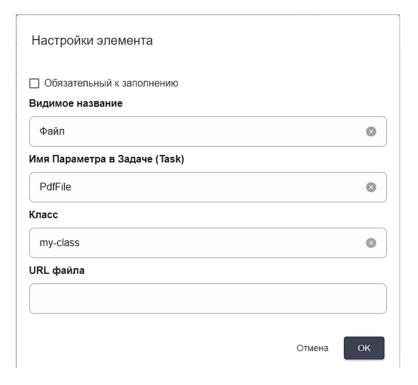


После этого добавьте поле табличных данных (Data Table) и укажите параметр задачи «Товары»

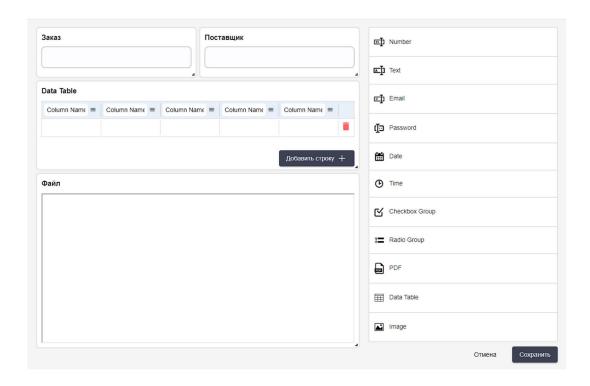


Далее добавьте поле «PDF» и укажите параметр задачи файла PDF.





При необходимости измените размеры полей (зажав левой кнопкой мыши значок - в правом нижнем углу выбранного поля) и переместить их (зажав левой кнопкой мыши кнопку .).



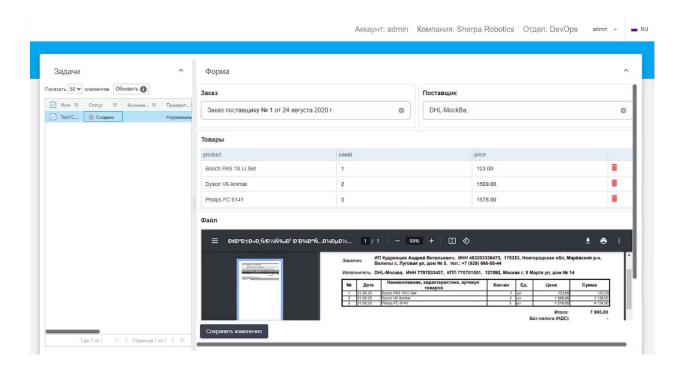


Нажмите кнопку «Сохранить». После этого нажмите кнопку , чтобы зайти в настройки созданного «Действия», и во вкладке «Доступ» скопируйте ссылку для доступа к действию.

Основные				
	Кнопки действий	Доступ		
Доступ по ссылке https://orchestrator2.sk	nerparpa.ru/main/my-acti	on 自		
			Отмена	ОК

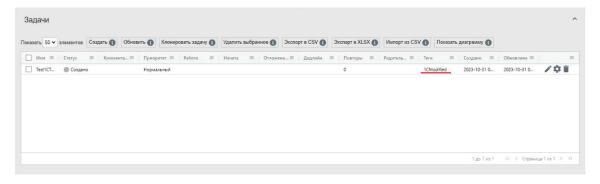
При переходе по ссылке доступна созданная форма.

Если Вы запустили роботов в процессе получения и распознавания накладных, то в очереди уже будет создана задача и при ее выборе будут показаны распознанные данные.



После проверки данных и внесения корректировки нажмите кнопку «Сохранить изменения». У задачи поменяется тег по алгоритму, который Вы указали в настройках «Действия» у этой кнопки.

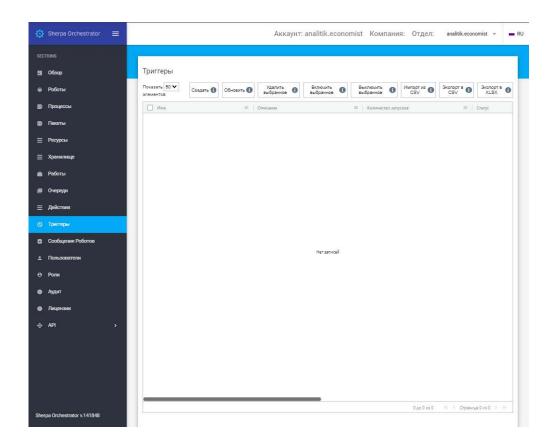




После выполнения указанных выше действий, данная задача больше не будет отображаться в списке формы рабочего места Действия. Можно приступать к внесению данных из обработанных задач в 1С.

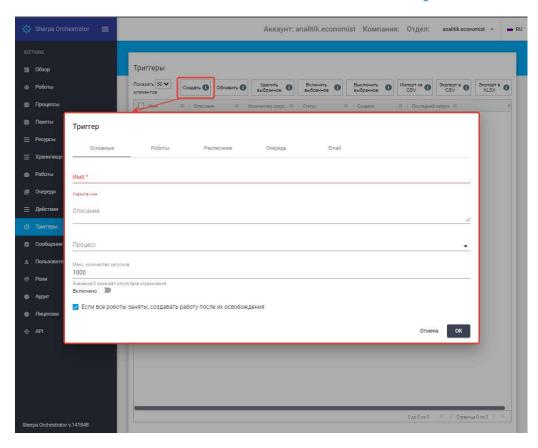
4.3.3.10 Экран «Триггеры»

Тригтеры необходимы для того, чтобы автоматически по заданным условиям создавать Работы, которые, далее, запускают роботов.



Для создания нового триггера необходимо нажать на кнопку «Создать» и в открывшейся форме во вкладках заполнить поля.





Вкладка «Основные»



У нового тригтера на вкладке, кроме наименования, требуется указать процесс (сценарий), который будет выполняться.



Триггер Основные Роботы Расписание Email Очередь * RMN Укажите имя Описание Тестовый процесс 2 Тестовый процесс 2 SimpleError Выберите процесс Макс, количество запусков LongWork Значение 0 означает отсутствие ограничения Включено

Если необходимо, чтобы данный тригтер выполнялся ограниченное количество раз, то в «Макс. количество запусков» проставить ограничение (число больше 0) или 0, если ограничения нет.

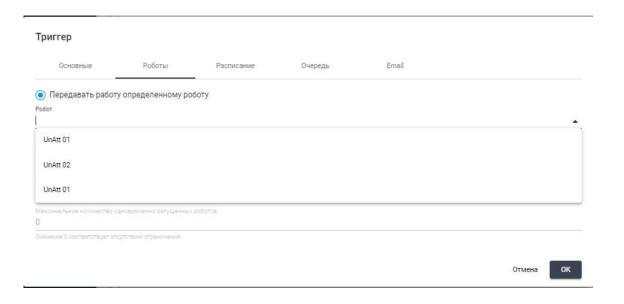


Вкладка «Роботы»



При необходимости тригтер позволяет создать работу:

• конкретному роботу – «Передавать работу определенному роботу»: из списка выбрать нужный;

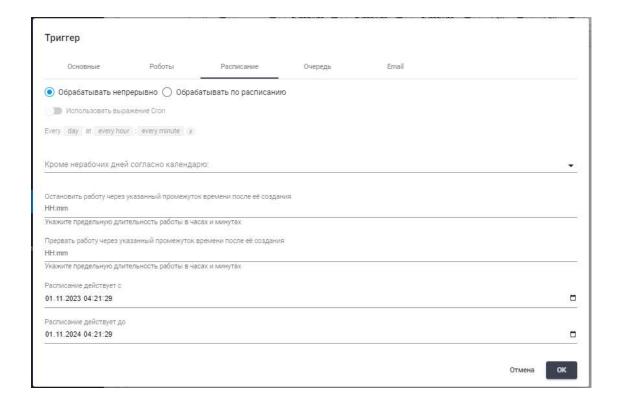


• группе роботов: в этом случае работу «подхватит» первый освободившийся из заданной группы робот. Здесь же возможно ограничить количество (максимум) роботов, которые будут выполнять указанную работу. Такая настройка позволяет произвести балансировку нагрузки на роботов.





Вкладка «Расписание»



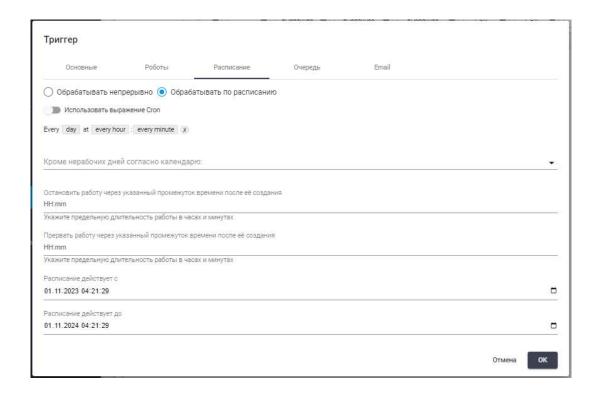
Настройка «Обрабатывать непрерывно» позволяет проводить работы с учетом ограничений, заданных на других вкладках.

Если требуется создавать работы в какие-то дни/часы или ввести ограничения по времени, то в меню «Расписание» можно задать необходимые настройки для триггеров, которые позволяют:

• выбрать каждый день/час/минуты;



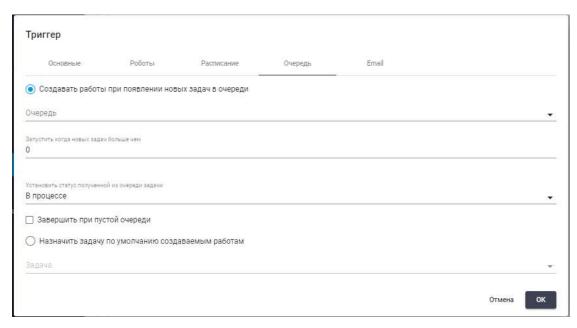
- использовать имеющиеся календари;
- задать временной промежуток для «мягкой» (остановить работу) или «жесткой» (прервать работу) остановки;
 - ограничить действие триггера конкретным периодом.



Вкладка «Очередь»

При использовании очередей можно также настроить триггер, исходя из списка очередности задач. Вкладка «Очередь» позволяет создавать работы при появлении новых задач в очереди.

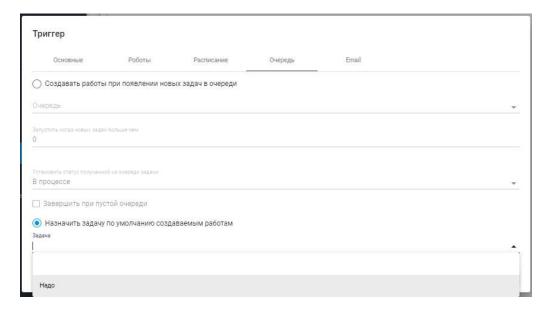




Для этого необходимо настроить параметры:

- Создавать работы при появлении новых задач в очереди:
- Очередь выбрать из списка необходимую очередь;
- Запустить когда новых задач больше чем указать значение больше 0, если робот будет запускаться не сразу при появлении задачи в выбранной очереди, а через заданное количество задач, например, через 5;
- Установить статус полученной из очереди задачи указать статус задачи (новая, в процессе);
- Завершить при пустой очереди выставить возможность выключиться/не выключиться при отсутствии задач в очереди.
- Назначить задачу по умолчанию создаваемым работам выбрать, если необходимо привязать тригтер к определенной задаче и использовать её параметры как входные данные для робота.





Вкладка «Email»

При необходимости создания работ, связанных с обработкой электронной почты, можно настроить триггер с определенными условиями.

Для этого необходимо заполнить поля на вкладке Email:

- протокол получения почты;
- кодировку;
- почтовый сервер и порт;
- логин и пароль;
- папку (если необходимо обрабатывать какую-то определенную папку с почтой);
 - параметры шифрования (использовать SSL);
 - признак новых писем (обрабатывать только непрочитанные письма);
- дополнительные параметры обработки (содержание текста в теме или теле письма, а также определенные вложения);
- обрабатывать, когда подходящих писем больше, чем (можно задать количество, при достижении которого необходимо запускать робота для обработки электронных писем).



Основные	Роботы	Расписание	Очередь	Email	
Создать работу по	поступлению Email				
Тротокол					
POP3 (IMAP					
TRIGGER_DIALOG.EMAIL_	TAB EMAIL SERVER ENG	ODING			
UTF-8 US-ASC					
Почтовый сервер					
Порт					
Если не задан, используется	порт по умолчанию для вы	бранного протокола			
Porse					
analitik.economist					
Пароль					
TRIGGER_DIALOG.EMA					
TRIGGER_DIALOG.EMA					
TRIGGER_DIALOG.EMAI TRIGGER_DIALOG.EMAIL_TAB					
TRIGGER_DIALOG.EMA TRIGGER_DIALOG EMAIL_TAB Danka	EMAIL_SHARED_MAILBOX_I				
TRIGGER_DIALOG.EMA TRIGGER_DIALOG.EMAIL_TAB DIANKA Ednii he bagaha, 6/det nposes	EMAIL_SHARED_MAILBOX_I				
TRIGGER_DIALOG.EMA TRIGGER_DIALOG.EMAIL_TAB DIANKA Ednii he bagaha, 6/det nposes	EMAIL_SHARED_MAILBOX_I				
TRIGGER_DIALOG.EMA TRIGGER_DIALOG.EMAIL_TAB DIANKA EDNIN HE SAJAHA, GYJET NOOSES UCNONDSOBATD SSL	EMAIL SHARED MAILBOX I	HIVT			
TRIGGER_DIALOG.EMAI TRIGGER_DIALOG.EMAIL_TAB TIANKA EGITH HE BABBHIR, GYGET NOOBES MCNONDBOOBBTD SSL	EMAIL SHARED MAILBOX I	HIVT			
TRIGGER_DIALOG EMA TRIGGER_DIALOG EMAIL_TAB TIANKA ECHI HE BAGBHA GIGET NOBES MCNONESOBATE SSL	EMAIL_SHARED_MAILBOX_I ИТЬОВ ПАПКА ВХОДЯЩИЕ	письма	х подстрок		
TRIGGER_DIALOG EMA ТЯПОВЕЯ_DIALOG EMAIL_TAB Папка Если не задана, будет прове; Использовать SSL Обрабатывать тол Обрабатывать, если т	EMAIL_SHARED_MAILBOX_I ЖТЬОК ПАПКА Еходицие БКО НЕПРОЧИТАННЫЕ I	письма	Un-vitoresibares		
TRIGGER_DIALOG EMA ТЯПОВЕЯ_DIALOG EMAIL_TAB Папка Если не задана, будет прове; Использовать SSL Обрабатывать тол Обрабатывать, если т	EMAIL_SHARED_MAILBOX_I ЖТЬОК ПАПКА Еходицие БКО НЕПРОЧИТАННЫЕ I	письма	Un-vitoresibares		
TRIGGER_DIALOG.EMA TRIGGER_DIALOG EMAIL_TAB Папка Если не задана, будет прове; Использовать SSL Обрабатывать тол Обрабатывать, если т Разделяйте нескольно подст	EMAIL_SHARED_MAILBOX_I октьов папка Еходищие вко непрочитанные I вма письма содержи рок запятыми, оставьте пу	письма ит любую из указанны стым если проверка не нужн			
TRIGGER_DIALOG.EMA ТЯПОВЕЯ_DIALOG EMAIL_TAB Папка Если не вадана, будет прове; Использовать SSL Обрабатывать тол Обрабатывать, если т Разделяйте месколько лодст	емать знаяер матьок папка входящие вма письма содержи рок запятыми, оставьте пу о письма содержит лк	письма ит любую из указанны стым если проверка не мужн робуго из указанных подст			
TRIGGER_DIALOG.EMA ТЯПОВЕЯ_DIALOG EMAIL_TAB Папка Если не вадана, будет прове; Использовать SSL Обрабатывать тол Обрабатывать, если т Разделяйте месколько лодст	емать знаяер матьок папка входящие вма письма содержи рок запятыми, оставьте пу о письма содержит лк	письма ит любую из указанны стым если проверка не мужн робуго из указанных подст			
TRIGGER_DIALOG.EMAIL_TAB Папка Если не задена, будет провес Использовать SSL Обрабатывать тол Обрабатывать, если т Разделяйте несколько подстро	емать знаяер матьок папка входящее в письма содерж; рок запятыми, оставьте пу о письма содержит лк запятыми, оставьтя пустым запятыми, оставьтя пустым	письма ит любую из указанны стым если проверка не мужн робуго из указанных подст	рок		
TRIGGER_DIALOG.EMAI TRIGGER_DIALOG.EMAIL_TAB Папка Если не задана, будет провез Использовать SSL Обрабатывать тол Обрабатывать, если т Разделяйте несколько подстро Обрабатывать, если тел Разделяйте несколько подстро Обрабатывать, если пис	емакц shared_mail.вож и меться папка Еходиция вма письма содержи рок запятыми, оставьте пу о письма содержит ли запятыми, оставьте пустым ьмо содержит вложен	письма ит любую из указанны тым если проверка не нужн обуго из указанных подст	рок их имен файлов	не нузіна	
TRIGGER_DIALOG.EMAI TRIGGER_DIALOG.EMAIL_TAB Папка Если не задана, будет провез Использовать SSL Обрабатывать тол Обрабатывать, если т Разделяйте несколько подстро Обрабатывать, если тел Разделяйте несколько подстро Обрабатывать, если пис	емакц shared_макцей жться папка Еходжире ько непрочитанные и ема письма содержи рок запятыми, оставьте пу о письма содержит лк запятыми, оставьте пустым ьмо содержит вложен клое запятыми, при необходи клое запятыми, при необходи клое	письма от любую из указанны отым если проверка не мужн юбую из указанных подст исле проверка че мужна им с любым из указанны им с любым из указанны	рок их имен файлов	не кухіма	

4.3.3.11 Экран «Сообщения Роботов»

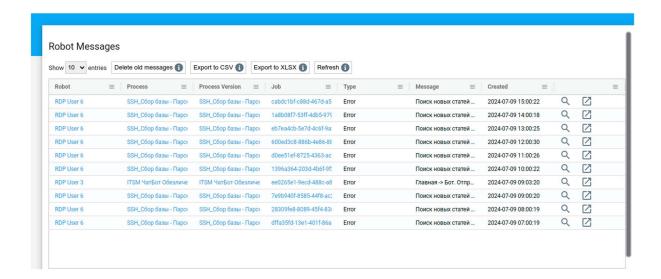
Сообщения роботов (логи) — это сообщения, которые роботы отправляют в Оркестратор по мере выполнения работ.

Экран «Сообщения роботов» содержит таблицу «Сообщения Роботов». Каждое сообщение роботов (лог) отражается в таблице «Сообщения Роботов» как одна строка.



Колонки таблицы «Сообщения Роботов».

№ п/п	Колонка	Описание колонки
1.	Робот	Отображает имя робота.
2.	Процесс	Отображает процесс, который выполняется роботом.
3.	Версия процесса	Отображает версию процесса, которая выполняется роботом.
4.	Работа	Отображает работу, которую выполняет робот.
5.	Тип	Отображает присвоенный сообщению робота (логу) тип. Возможные значения: • Info — информация; • Critical — критично; • Error — ошибка; • Warning — предупреждение; • Debug — отладка.
6.	Сообщение	Отображает сообщение робота.
7.	Создано	Отображает дату и время создания данного сообщения робота.





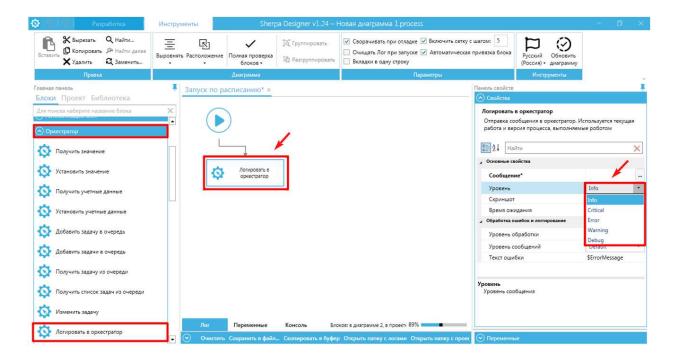
Все колонки таблицы «Сообщения роботов» могут быть настроены пользователем. Функции настройки колонок таблиц описаны в разделе «4.3.2 Рабочая область Оркестратора».

4.3.3.11.1 Логирование в Оркестратор на стороне Дизайнера

Логи в Оркестраторе — это логи, которые роботы отправляют сами. Есть некоторое количество системных логов. Системные логи записываются, если внутри сценария произошла ошибка, которая нигде не была обработана, и сценарий завершился аварийно.

Залогировать можно любые параметры. В Дизайнере есть специальный блок «Логировать в Оркестратор».

В параметрах блока «Логировать в Оркестратор» указывается сообщение (любая переменная любого типа) и уровень критичности.



При использовании данного блока вы можете выбирать уровень логирования:

- Info Информация;
- Critical Критично;

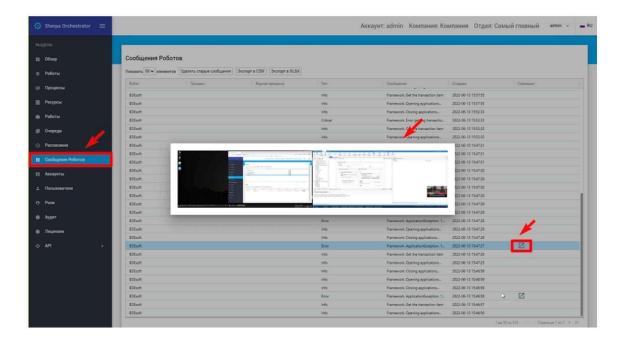


- Error Ошибка;
- Warning Предупреждение;
- Debug Отладка.

Также вы можете вместе с сообщением отправить в Оркестратор скриншот, предварительно установив галочку в окне «Скриншот».

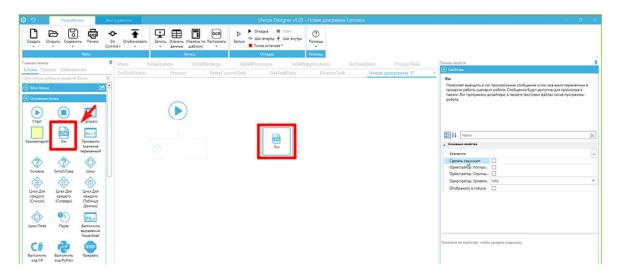
Сообщение будет отображаться на экране «Сообщения Роботов» Оркестратора.

В случае если к сообщению от робота был прикреплен скриншот, соответствующая иконка появится в колонке «Скриншот». Кликните на иконку, чтобы увидеть скриншот в исходном размере.



Другой способ логирования – в Дизайнере в блоке Лог проставить галочку «Логировать в Оркестратор».





Примечание: не рекомендуется использовать логирование в Оркестратор внутри тела циклов с коротким интервалом, т.к. это может привести к появлению большого количества лишних сообщений. В целях безопасности Оркестратор может ограничивать максимальное число получаемых в единицу времени сообщения от робота.

4.3.3.12 Экран «Аккаунты»

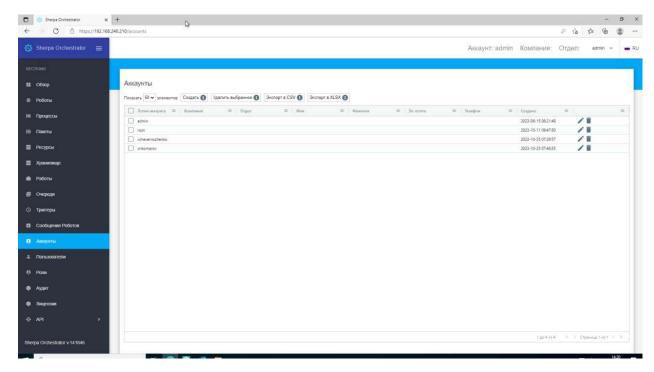
Аккаунты представляю собой разделы Оркестратора, каждый из которых обладает собственным набором работ, процессов, ресурсов, роботов, очередей и пользователей.

С помощью аккаунтов Оркестратор может быть поделен на несколько логических, не связанных друг с другом, областей (например, по департаментам), и для каждой из них вести отдельный учет и раздельное использование всех объектов.

Аккаунты представляют собой изолированные друг от друга пространства для хранения сущностей Оркестратора, доступ к которым осуществляется по логину и паролю пользователя.

Экран Аккаунты содержит таблицу «Аккаунты».





Обратите внимание! Аккаунты внутри Оркестратора не являются пользователями. Для определения прав пользователей предусмотрен отдельный экран «Пользователи».

В рамках одного Оркестратора можно завести несколько аккаунтов, в каждом из них будут свои роботы, процессы, работы, очереди и прочие сущности. Аккаунты абсолютно изолированы друг от друга: по роботам, процессам и прочим ресурсам.

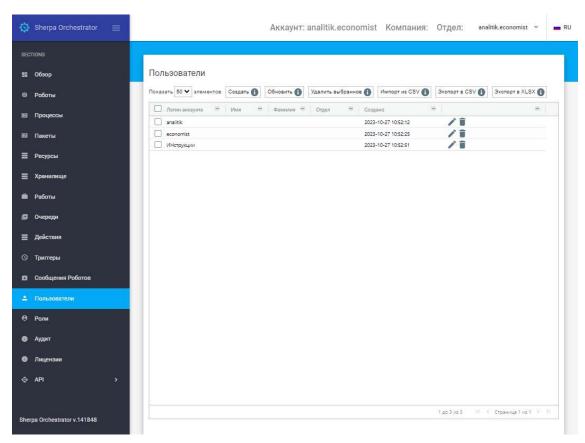
Часто используется два аккаунта, чтобы поделить Оркестратор и управляемых им роботов на используемые в тестовых целях (test road) и занятые реальными бизнес-задачами. При небольшом количестве роботов для выполнения текущих задач одного аккаунта вполне достаточно.

4.3.3.13 Экран «Пользователи»

Пользователи могут быть созданы в рамках аккаунта. Каждому пользователю присваивается определенная роль. Чтобы создать пользователя сначала требуется определить его роль на отдельном экране. Пользователю можно присвоить определенную роль или несколько ролей.

Экран «Пользователи» содержит таблицу «Пользователи».





Для создания нового пользователя необходимо нажать на кнопку «Создать» в меню в таблице «Пользователи» и в открывшейся форме заполнить поля.

Форма создания нового пользователя предлагает три варианта аутентификации, из которых можно выбрать нужную с помощью установки флага:

- Локальная аутентификация;
- LDAP аутентификация;
- OpenID аутентификация.

Важно! При создании и/или редактировании пользователя должно быть указано какой именно способ аутентификации для него использовать.

Элементы интерфейса формы создания нового пользователя.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	флаг «Локальная»	Позволяет установить вариант локальной аутентификации пользователя.



1.1	текстовое поле «Логин»	Логин пользователя. Обязательное для заполнения поле.
1.2.	текстовое поле «Пароль»	Пароль пользователя. Обязательное для заполнения поле.
2.	флаг «Использовать LDAP аутентификацию»	Позволяет установить вариант аутентификации пользователя через LDAP.
2.1	поле «Пользователи LDAP»	Позволяет указать пользователя LDAP с помощью выпадающего списка. Данный вариант аутентификации доступен, если в конфиге Оркестратора прописаны данные для подключения к LDAP-серверу. Данное поле доступно при LDAP аутентификации.
3.	флаг «Использовать OpenID ayтентификацию»	Позволяет установить вариант аутентификации пользователя через OpenID.
3.1	текстовое поле «oAuth id»	Идентификатор oAuth. Обязательное для заполнения поле. Данное поле доступно при OpenID аутентификации.
4.	текстовое поле «Эл.почта»	Адрес электронной почты для автоматической отправки сообщений.
5.	текстовое поле «Имя»	Имя пользователя.
6.	текстовое поле «Фамилия»	Фамилия пользователя.
7.	текстовое поле «Отдел»	Название отдела/департамента места работы или расположения пользователя
8.	флаги «Уведомлять о сообщениях от роботов следующих типов»	Позволяет пользователю (при указании адреса e-mail и установке флагов для типов сообщений роботов) получать автоматически отправленные уведомления на электронную почту. Возможные варианты:



		 Critical (Критическая ошибка); Error (Ошибка); Warning (Предупреждение); Notice (Уведомление); Info (Информация); Debug (Отладка).
9.	флаги «Уведомлять о сообщениях аудита следующих типов»	Позволяет пользователю (при указании адреса e-mail и установке флагов для типов сообщений Оркестратора) получать автоматически отправленные уведомления на электронную почту. Возможные варианты: • Critical (Критическая ошибка); • Error (Ошибка); • Warning (Предупреждение); • Notice (Уведомление); • Info (Информация); • Debug (Отладка).
10.	флаги «Роль»	Позволяет указать заранее настроенную роль (одну или несколько), в соответствии с которой будет настроен доступ к экранам Оркестратора и действиям на этих экранах (создание, редактирование, чтение и удаление объектов).
11.	флаг «Пользователь заблокирован»	Позволяет заблокировать пользователя.
12.	флаг «Уведомлять об истечении лицензий на email»	Позволяет включить функцию уведомления пользователя об истечении лицензий на электронную почту.

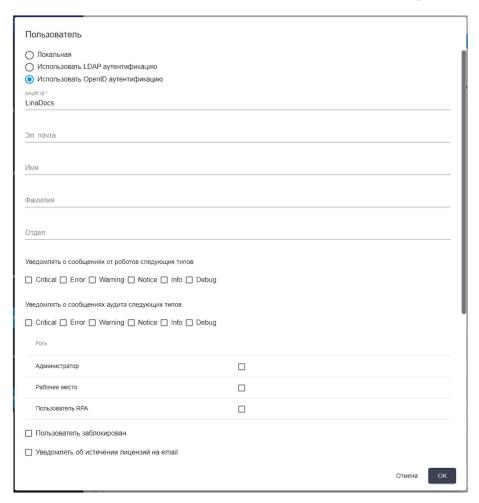


Пользователь		
Локальная		
О Использовать LDAP аутентификацию		
Оспользовать OpenID аутентификацию		
1огин аккаунта *		
inaDocs		
Пароль *		
Эл. почта		
1мя		
Фамилия		
D		
Этдел		
/ведомлять о сообщениях от роботов следуюц	дих типов	
☐ Critical ☐ Error ☐ Warning ☐ Notice ☐] Info Debug	
/ведомлять о сообщениях аудита следующих т	гипов	
☐ Critical ☐ Error ☐ Warning ☐ Notice ☐] Info Debug	
Роль		
Администратор		
Рабочее место		
Пользователь RPA		
	0	
Пользователь RPA Пользователь заблокирован	0	
□ Пользователь заблокирован		



Пользователь		
Локальная		
 Использовать LDAP аутентификацию 		
О Использовать OpenID аутентификацию		
Пользователи LDAP		•
Эл. почта		
Имя		
Фамилия		
Отдел		
Уведомлять о сообщениях от роботов следующих типс Сritical Error Warning Notice Info [Уведомлять о сообщениях аудита следующих типов Сritical Error Warning Notice Info [☐ Debug	
Роль		
Администратор		
Рабочее место		
Пользователь RPA		
□ Пользователь заблокирован		
		Отмена





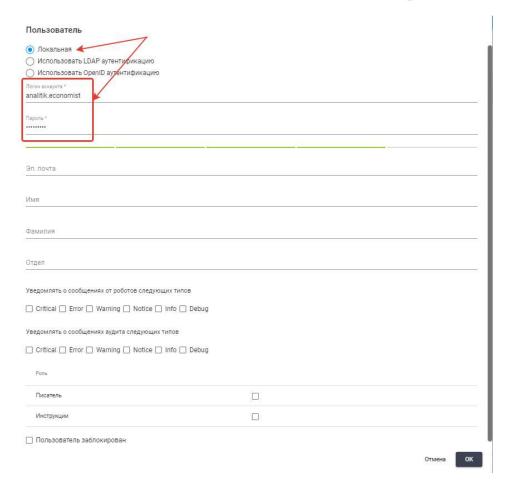
Примечание: пользователь, под которым осуществляется вход из Дизайнера, должен иметь право доступа к процессам, то есть если в своей роли он не будет иметь права доступа к процессам, у него не будет возможности опубликовать из Дизайнера новые сценарии.

4.3.3.13.1 Авторизация пользователей

Локальная авторизация

Авторизация пользователя происходит за счет логина и пароля, заданного в настройке пользователя.

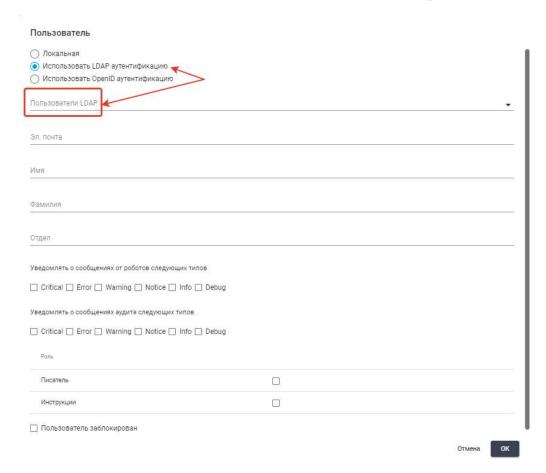




LDAP-аутентификация

При создании пользователя с методом входа в систему — LDAPаутентификация необходимо заполнить поле «Пользователи LDAP».



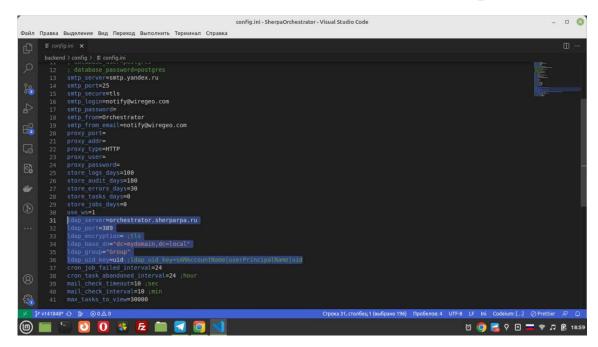


Для использования AD-аутентификации необходимо в файле config.ini, расположенном / opt/SherpaOrchestrator/backend/config/config.ini, заполнить соответствующие параметры:

- ldap_server,
- ldap_port,
- ldap_encryption,
- ldap_base_dn,
- ldap_group.

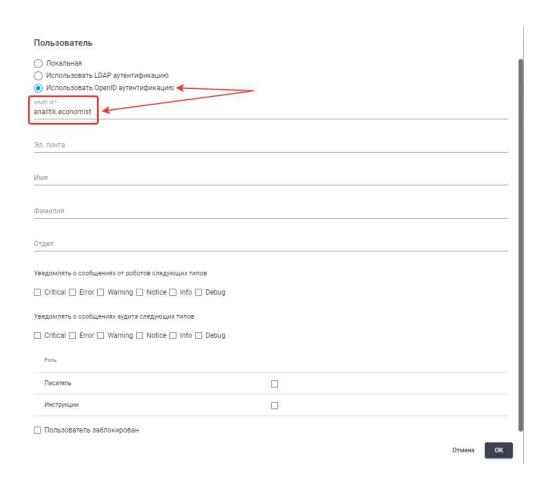
Затем перезапустить контейнер для установки через Docker (при локальной установке по необходимости перезагрузить сервер).





OpenID-аутентификация

При создании пользователя с методом входа в систему — OpenID - аутентификация необходимо заполнить поле «oAuth id».





Для подключения OpenID аутентификации необходимо в файле config.ini, расположенном по пути /opt/SherpaOrchestrator/backend/config/config.ini, заполнить соответствующие параметры:

- oAuthClientId,
- oAuthClientSecret,
- oAuthAuthUri,
- oAuthTokenUri,
- oAuthUserInfoUri

При запросе токена оркестратор передает параметр 'redirect_uri'.

Если необходимо, то указать Callback URL вручную:

• yourDomainURL/api/gui/system/oAuthHook

В настройках аккаунта или пользователя выберите «Use OpenID authentication» и укажите идентификатор пользователя в системе OpenID.

На рисунке ниже пример заполнения для аутентификации через Яндекс.

```
Config.ini - Visual Studio Code

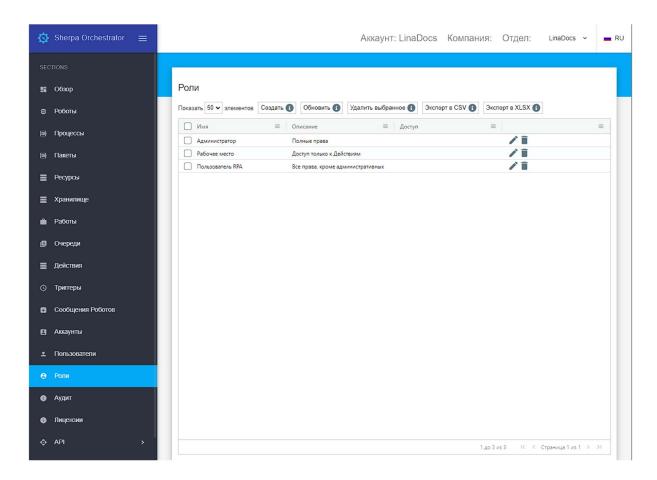
Config.ini - Visual Studio Code
```



4.3.3.14 Экран «Роли»

Роль – это параметр, который определяет набор прав доступа к экранам Оркестратора, а также функциям в рамках выбранных экранов.

Экран «Роли» содержит таблицу «Роли».



Для создания новой роли необходимо нажать на кнопку «Создать» в таблице «Роли» и заполнить открывшуюся форму.

Форма создания новой роли содержит две вкладки:

- Основные;
- Доступ.

Вкладка «Основные»

Во вкладке «Основные» находятся два текстовых поля:

• Имя (Имя роли. Обязательное для заполнения поле);



• Описание (описание роли).



Вкладка «Доступ»

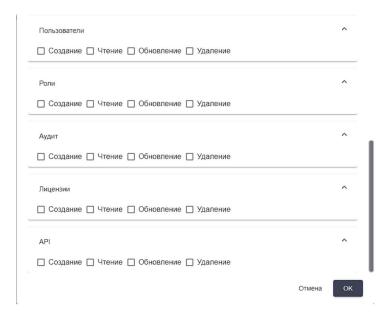
Во вкладке «Доступ» необходимо отметить флажками те функции для тех экранов, которые будут доступны пользователю с создаваемой ролью. Доступные функции для каждого экрана:

- Создание;
- Чтение;
- Обновление;
- Удаление.



Основные		Доступ				
метьте галочка	ми объекты	и операции, разр	ешенные для данн	ой роли		
Обзор						,
□ Создание	□ Чтение	□ Обновлени	е 🗌 Удаление			
Роботы						,
□ Создание	□ Чтение	□ Обновлени	е 🗌 Удаление			
Процессы						,
□ Создание	□ Чтение	□ Обновлени	е 🗌 Удаление			
Пакеты						,
□ Создание	□ Чтение	□ Обновлени	е 🗌 Удаление			
Ресурсы						,
	□ Чтение	□ Обновлени	е 🗌 Удаление			
Хранилище ☐ Создание	□ Чтение	□ Обновлени	е 🗌 Удаление		Отмена	
	□ Чтение	□ Обновлени	е 🗌 Удаление		Отмена	
□ Создание	□ Чтение	□ Обновлени	е 🗌 Удаление		Отмена	
□ Создание Роль Работы		□ Обновленив □ Обновлени			Отмена	
□ Создание Роль Работы					Отмена	
СозданиеРольРаботыСозданиСчереди	1е □ Чтени		е 🗌 Удаление		Отмена	
СозданиеРольРаботыСозданиСчереди	1е □ Чтени	е 🗌 Обновлени	е 🗌 Удаление		Отмена	
СозданиеРольРаботыСозданиОчередиСозданиДействия	ие □ Чтении	е 🗌 Обновлени	е		Отмена	
СозданиеРольРаботыСозданиОчередиСозданиДействия	ие □ Чтении	е	е		Отмена	
 Создание Роль Работы Создани Очереди Создани Действия Создани Триггеры	ие Чтении ие Чтении ие Чтении	е	е		Отмена	
 Создание Роль Работы Создани Очереди Создани Действия Создани Триггеры	ие Чтении ие Чтении ие Чтении	е Обновлени е Обновлени е Обновлени	е		Отмена	
РОЛЬ Работы Создани Очереди Создани Действия Триггеры Создани Сообщения	ие Чтении ие Чтении ие Чтении ие Чтении Роботов	е Обновлени е Обновлени е Обновлени	е		Отмена	
РОЛЬ Работы Создани Создани Очереди Создани Действия Создани Триггеры Создани Сообщения Аккаунты	не Чтении не Чтении не Чтении Роботов не Чтении	е Обновлени е Обновлени е Обновлени е Обновлени	е		Отмена	





Редактирование ранее созданной роли

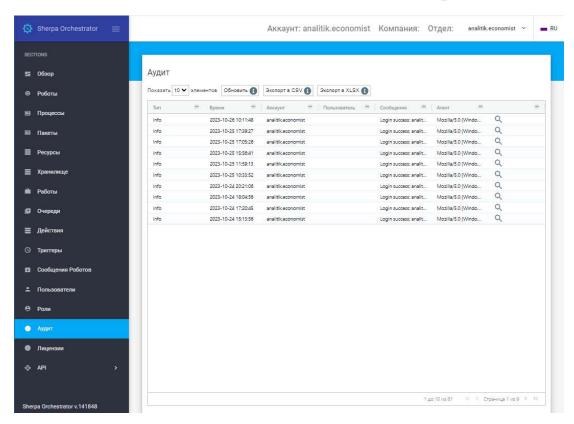
Для просмотра и редактирования роли необходимо выбрать ее в списке и нажать на кнопку . После этого откроется форма редактирования роли, в которую можно внести необходимые изменения. Помимо полей, заполняемых при создании роли, при редактировании в форме указан GUID (уникальный идентификатор, присвоенный роли после ее создания). Данное поле нельзя отредактировать.

4.3.3.15 Экран «Аудит»

Экран «Аудит» показывает собственные события Оркестратора.

Интерфейс содержит одну таблицу «Аудит», которая представляет собой список сообщений о событиях, произошедших внутри Оркестратора.

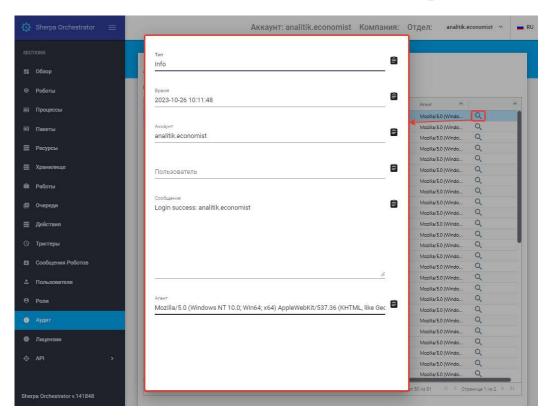




Отображается информация о создании/удалении Аккаунта (учетной записи) Пользователя, попытки ввода Логина (успешно/неуспешно) Пользователями, а также все манипуляции Пользователей с различными объектами Оркестратора.

При нажатии на значок лупы можно просмотреть в виде карточки запись.





4.3.3.16 Экран «Лицензии»

Программные роботы для запуска на рабочих местах пользователей требуют активации соответствующих лицензий (лицензионных ключей) для подтверждения законного использования ПО. Во время активации программное обеспечение Sherpa Robot или Sherpa Orchestrator передает сведения о компьютере вендору.

Лицензии (лицензионные ключи) выдаются поставщиками для компанийклиентов и представляют собой последовательности букв и цифр.

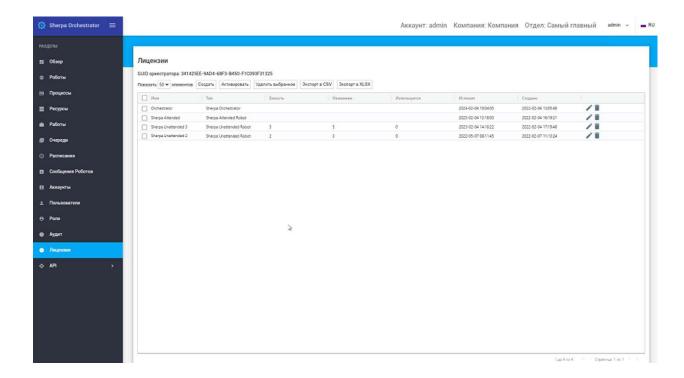
Лицензии определяют количество и тип роботов, работы для которых могут запускаться Оркестратором. Робот может использовать как локальную лицензию (которая хранится на компьютере, на котором установлен сам робот), так и лицензию из Оркестратора.

В Оркестратор можно добавить лицензии как типа Attended, так и Unattended, далее подробно описаны особенности добавления каждого типа Лицензий.



Допускается одновременное использование робота в Attended и Unattended-режиме.

Экран «Лицензии» содержит одну таблицу «Лицензии».



В Оркестратор следует добавить все лицензии, которыми требуется управлять.

Оркестратор позволяет хранить и использовать лицензии для всех видов роботов, а также лицензию для самого Оркестратора.



Таблица «Лицензии» содержит следующие колонки:



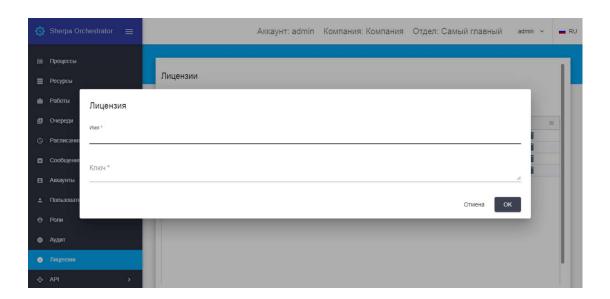
Колонки таблицы «Лицензии».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	колонка «Имя»	Имя лицензии.
2.	колонка «Тип»	Тип лицензии. Возможные варианты: Attended; Unattended; Orchestrator; Attended Floating.
3.	колонка «Емкость»	Количество роботов, которое может быть запущено одновременно под управлением этой лицензией. Данная колонка предназначена для Unattended-лицензий.
4.	колонка «Назначено»	Количество роботов из экрана «Роботы», которым была назначена эта лицензия. Количество назначенных роботов может быть больше, чем количество в колонке «Емкость». Данная колонка предназначена для Unattended-лицензий.
5.	колонка «Используется»	Количество роботов, которые используют лицензию в настоящий момент. Количество роботов, использующих лицензию не должно превышать количество, указанное в колонке «Емкость».
6.	колонка «Истекает»	Дата и время окончания действия лицензии. При добавлении новой лицензии в колонке «Истекает» будет по умолчанию стоять аналогичная дата создания до того момента, когда лицензия будет активирована.
7.	колонка «Создано»	Дата и время создания лицензии.

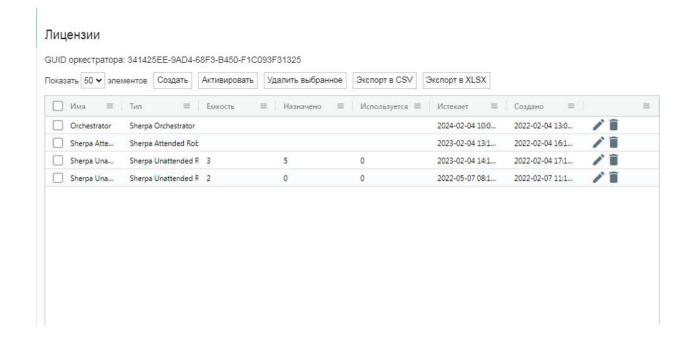


Создание новой Лицензии в Оркестраторе

Для создания новой лицензии, необходимо нажать на кнопку «Создать», ввести какое-либо название лицензии (название служит только для информационных целей), ввести в поле ниже предоставленный вам ключ, нажать кнопку «ОК».



Появится новая строка в таблице Лицензии:



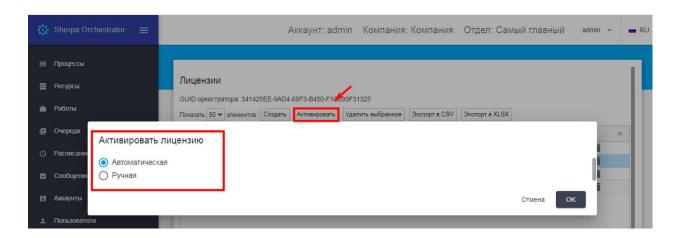


Тип лицензии определяется автоматически, всего есть четыре типа лицензий:

- Sherpa Orchestrator лицензия на Оркестратор;
- Sherpa Attended лицензия на Sherpa Attended Robot;
- Sherpa Unattended лицензия на Sherpa Unattended Robot;
- Attended Floating лицензия на Attended Floating Robot.

При добавлении новой лицензии в колонке «Истекает» дата будет по умолчанию стоять аналогичная дате создания. Это значит, что данная лицензия по умолчанию не активирована (и для ее использования требуется активация).

Далее нужно нажать кнопку «Активировать» для активации лицензии.



Далее нужно выбрать один из двух типов активации: автоматическую или ручную.

- При выборе автоматической активации нужно просто кликнуть на «ОК». Автоматическая активация возможна только в том случае, если ваш Оркестратор имеет открытый доступ в Интернет.
- При выборе ручной активации появляется код запроса, который нужно передать вендору для подтверждения ручной активации. В ответ на код запроса вам будет передан код ответа, его необходимо ввести в соответствующее поле и нажать «ОК». Лицензия будет активирована и в колонке «Истекает» изменится дата и время лицензии.

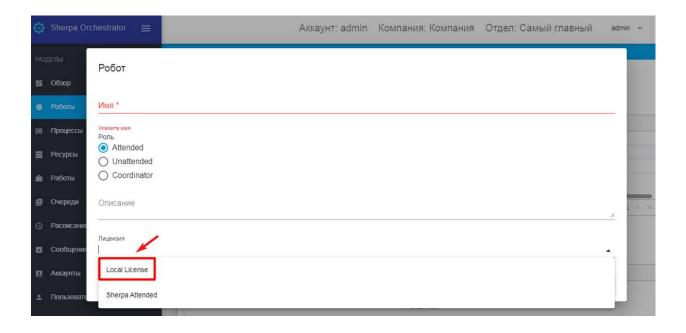


Активировать лицензию		
АвтоматическаяРучная		
Код запроса QkUyMTg5OTM4NTI0NkY2NiBFMTNENkNFNjYyNTE2QUUxNzM3MTIzNkI4NjhCN0U3QkMzM0NENEYzOTE4MzZBMDE=		
Код ответа		
	Отмена	ОК

4.3.3.16.1 Активация лицензии

Локальная активация лицензии Attended-роботов

Пункт Локальная лицензия активен по умолчанию в том случае, если у робота своя лицензия, которая была активирована локально. Для выбора локальной активации необходимо выбрать «Local License» при создании робота. Это будет означать, что лицензия будет использоваться локально, а не из Оркестратора.



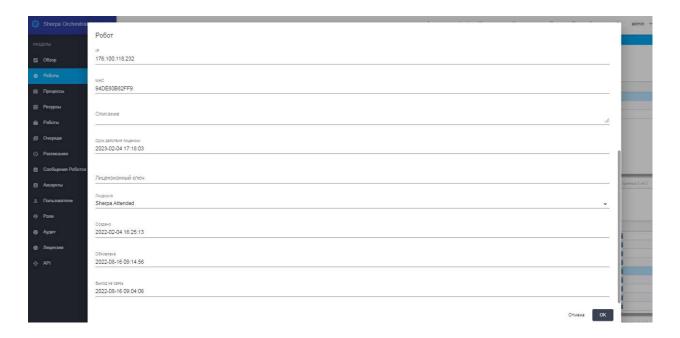
Активация лицензии Attended-Роботов из Оркестратора

Attended-роботов можно активировать и задавать им лицензии, в том числе и из Оркестратора (удаленно).



Если робот не был активирован локально, он может пользоваться лицензией из Оркестратора.

В таком случае, если вы захотите добавить Attended-робота и активировать его из Оркестратора, сначала нужно будет добавить робота (на экране «Роботы»), затем добавить лицензию на этого робота в экран «Лицензии», а затем внутри робота указать, что именно этот робот использует именно эту лицензию.



После этого нужно вернуться на экран Лицензии и нажать кнопку «Активировать».



Тогда для данного робота произойдет удаленная активация лицензии.

Для активации Attended-робота через Оркестратор вам необходимо сделать следующее:

1) Подключить робот к Оркестратору.



Для этого сначала создать робота на экране Роботы в оркестраторе и скопировать GUID из карточки робота.

Затем в окне Настройки, вкладка Оркестратор указать URL или IP адрес оркестратора (вместе с протоколом, например https://) и GUID робота, скопированный из оркестратора.

Нажать на кнопку Подключиться и убедиться, что робот успешно подключился к оркестратору.

На экране Роботы в оркестраторе статус соответствующего робота станет зелёным.

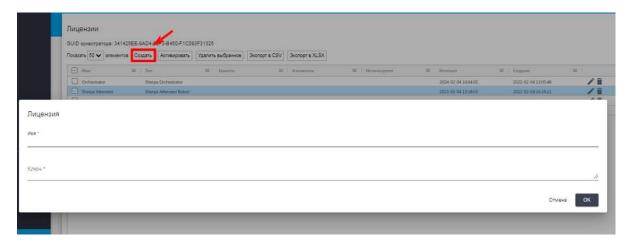
- 2) Создать лицензию для этого робота на экране «Лицензии» и ввести лицензионный ключ, который мы вам выдали.
- 3) На экране «Роботы» в свойствах робота в поле «Лицензия» выбрать из выпадающего списку ту лицензию, которую вы создали на предыдущем шаге, сохранить карточку робота.
- 4) На экране «Лицензии» поставить галочку в строку соответствующей лицензии и нажать на кнопку «Активировать».

Далее выбрать вариант активации «Автоматический» или «Ручной» (для активации через код запроса и код ответа), и пройти процедуру активации как вы делали для оркестратора и Unattended-роботов.

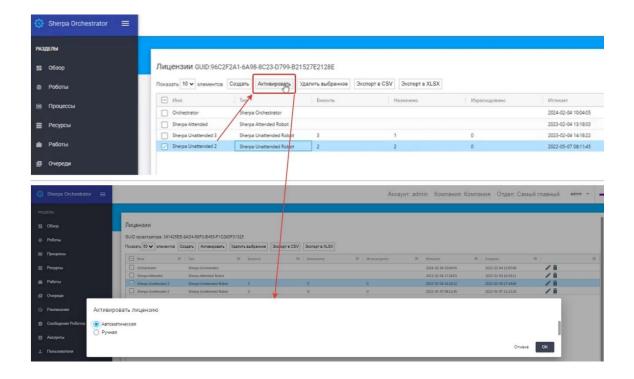
Если у вас выдаётся сообщение об ошибке что робот не подключен к оркестратору или требуется присвоить ему лицензию — подождите около минуты и попробуйте ещё раз.

Активация лицензии Unattended-роботов в Оркестраторе





Лицензию на Unattended-робота нужно сначала создать с помощью кнопки «Создать». Требуется ввести наименование, ключ лицензии. Лицензия появляется в списке лицензий, для нее сразу указывается тип робота: Unattended-робот. Срок действия Лицензии автоматически будет ограничен датой ее заведения. Для установки правильной даты и времени (согласно условиям ее приобретения) нажмите на кнопку «Активировать» и выполните те же самые шаги, как если бы активация лицензии проходила в Оркестраторе.



У Unattended-лицензии дополнительно есть параметр «Емкость», то есть количество роботов, которые одновременно могут работать на разных машинах с использованием этой лицензии.





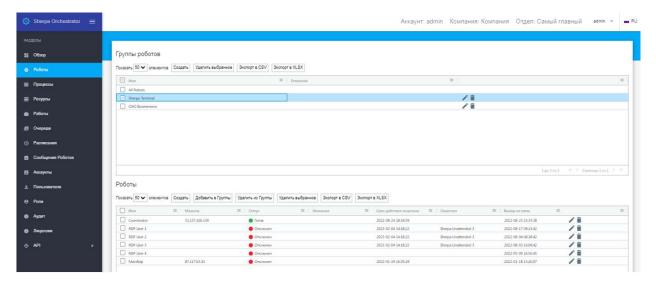
Обратите внимание на колонку «Назначено». Это означает, что лицензия была назначена какому-то конкретному или нескольким Unattended-роботам. На скриншоте ниже можно увидеть, что лицензия Sherpa Unattended 2 назначена сразу двум роботам – MainRdp и Rdp2, и они одновременно могут пользоваться этой лицензией.

Роботы			
Показать 50 ∨ элементов Создать	Добавить в Группы Удалить из Г	руппы Удалить выбранное Экспорт в CSV	Экспорт в XLSX
□ Имя Машина	Статус Описание	Срок дейс Лицензия	Выход на с
Coordinator	Готов	2022-04-08 2	2022-04-07 1
Unattended	Отключен	2023-02-04 1 Sherpa Unattended 3	2022-04-07 12
Rdp2	Отключен	2022-05-07 0 Sherpa Unattended 2	2022-01-29 14
MainRdp	Отключен	2022-05-07 0 Sherpa Unattended 2	2022-01-18 13
AttendedT	Отключен	2023-02-04 1 Sherpa Unattended 3	2022-04-07 1
RDP User 1	Отключен	2023-02-04 1 Sherpa Unattended 3	2022-03-28 11

Значение в колонке «Израсходовано» изменяется динамически по мере того, как роботы подключаются и отключаются. В ситуации, когда задание выполнено и робот отключается, значение уменьшается.

После добавление лицензии для Unattended-робота и ее активации необходимо завести робота.





В настройках робота выбрать подходящую лицензию.



4.3.3.17 Экран «API»

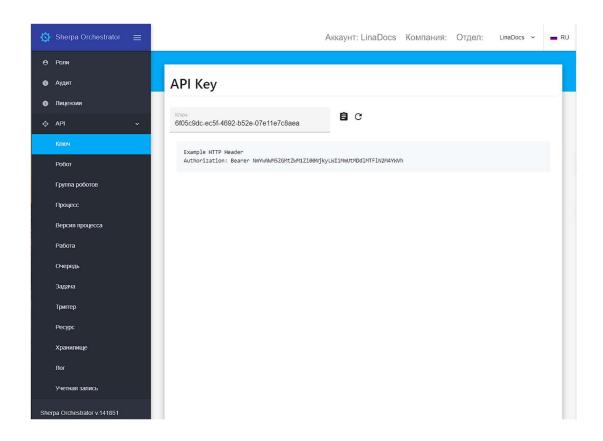
Для интеграции с внешними системами (при необходимости), Оркестратор предлагает набор инструментов для обмена данными через API.



Раздел «Ключ»

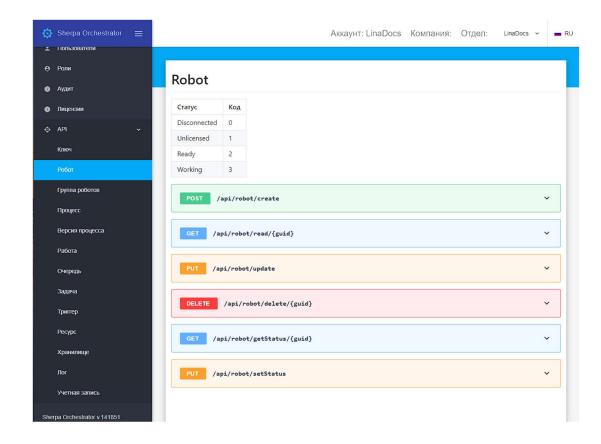
Раздел «Ключ» экрана «АРІ» предоставляет доступ к ключам (АРІ Кеу), который необходимо передать в HTTP-заголовке.

Запросы к API осуществляются с обязательной передачей API Кеу, который хранится в настройках Учетной записи. Подробные примеры запросов приведены в разделе по каждому блоку отдельно.

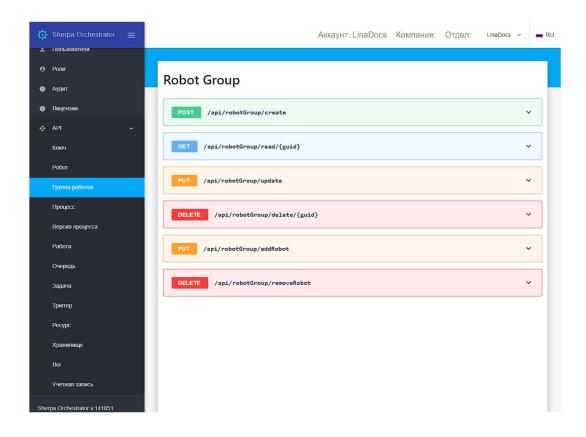




Раздел «Робот»

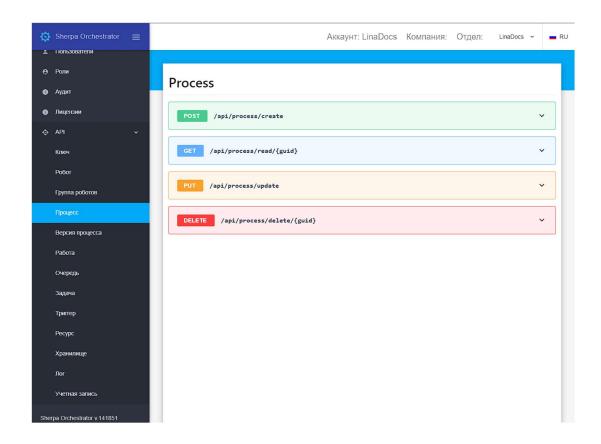


Раздел «Группа роботов»

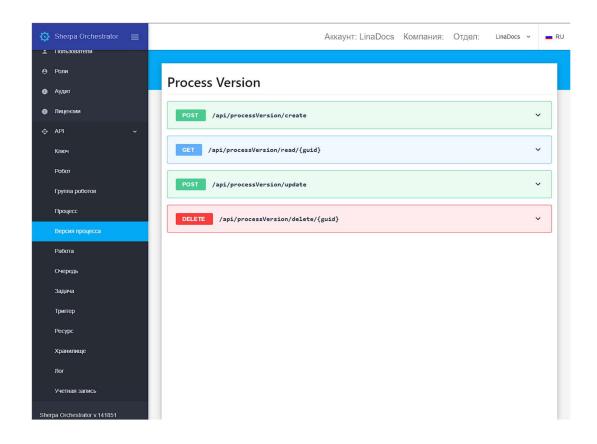




Раздел «Процесс»



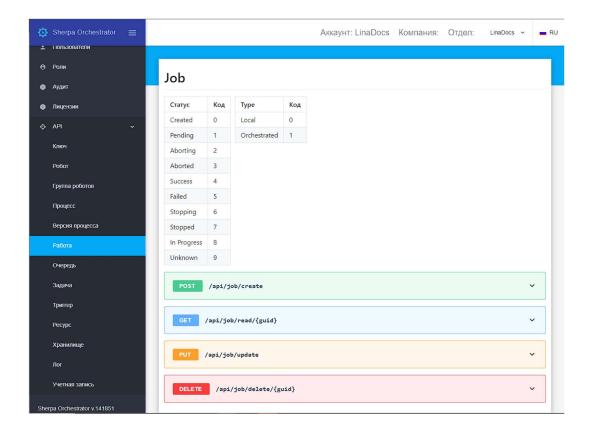
Раздел «Версия процесса»



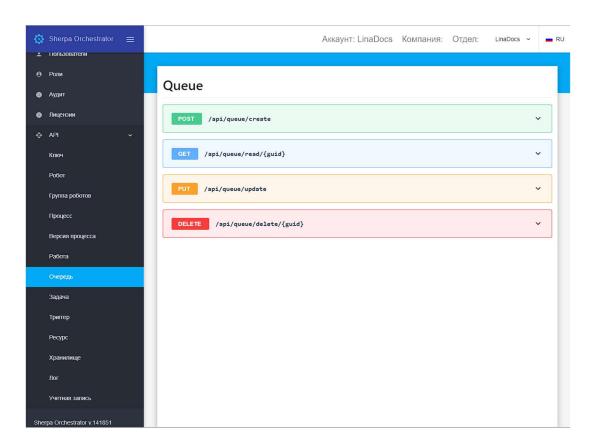




Раздел «Работа»

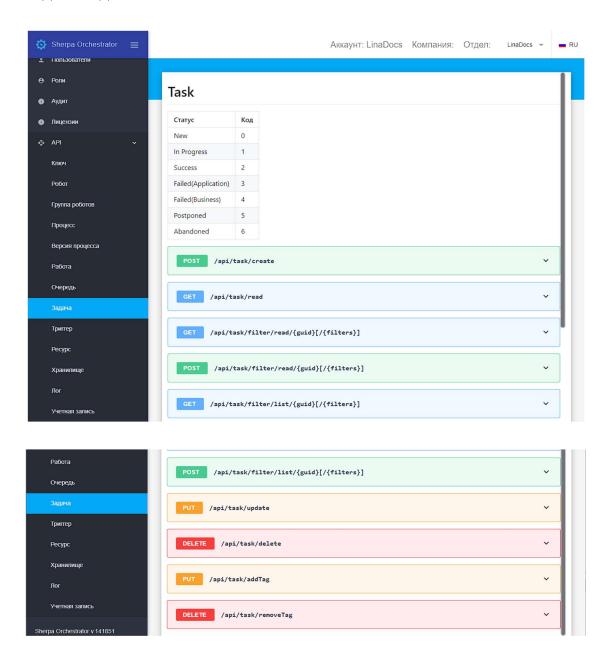


Раздел «Очередь»



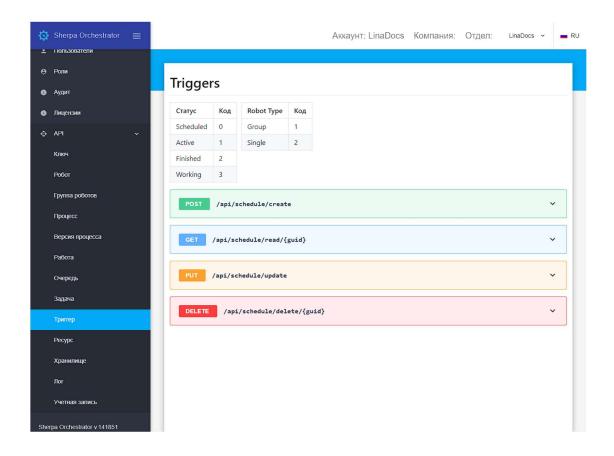


Раздел «Задача»

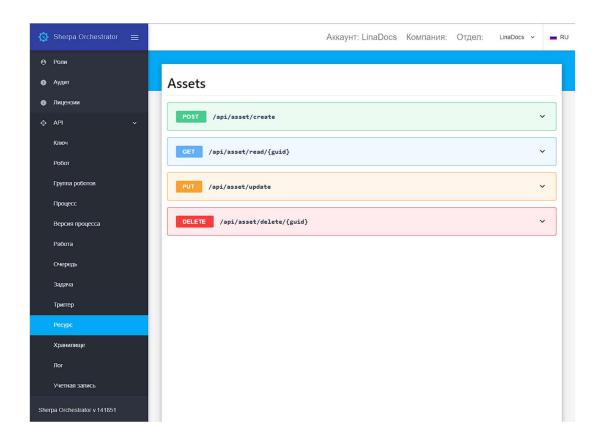




Раздел «Триггер»

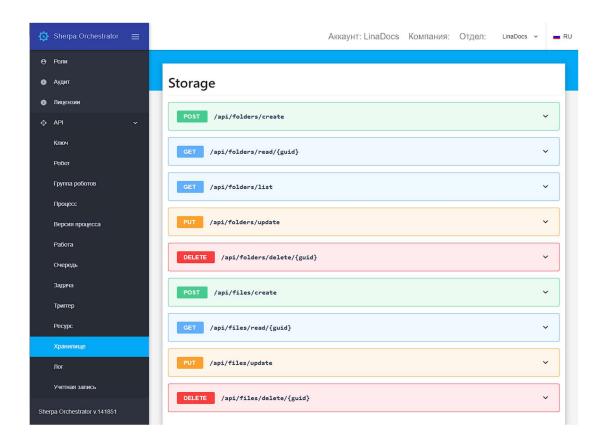


Раздел «Ресурс»

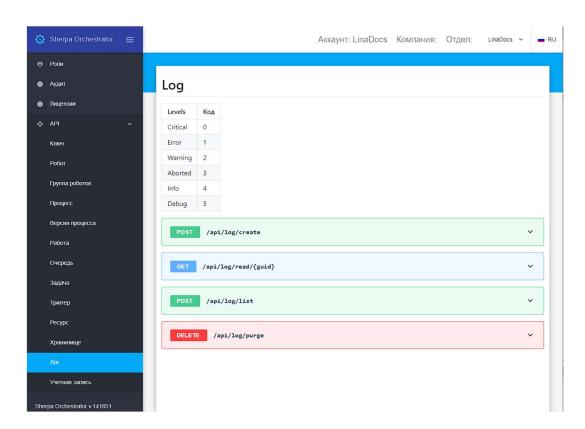




Раздел «Хранилище»

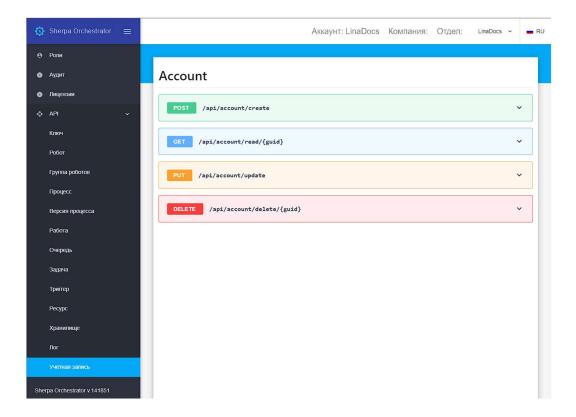


Раздел «Лог»





Раздел «Учетная запись»



4.4 Резервное копирование и восстановление

Для осуществления резервного копирования (бэкапа) в Sherpa Orchestrator (версии с Docker) необходимо осуществить резервное копирование настроек, файла cache.bin, а также создать дамп данных из базы данных. Ниже представлена подробная пошаговая инструкция для осуществления резервного копирования (бэкапа) в Sherpa Orchestrator.

1) Копирование настроек.

Скопируйте папку config, находящуюся по указанному пути: папка_установки/backend/config

(по умолчанию путь: opt/SherpaOrchestrator/backend/config).

2) Копирование файла cache.bin.

Скопируйте файл cache.bin, находящийся внутри Docker-контейнера Orchestrator. Путь, по которому находится файл внутри контейнера:



opt/SherpaOrchestrator/backend/app/cache.bin

Также можно использовать следующую команду в терминале:

docker cp orchestrator:/opt/SherpaOrchestrator/backend/app/cache.bin /путь куда осуществляем копирование

- 3) Создание дампа базы данных.
- войдите в контейнер с базой данных: docker exec -it orchestrator-db bash
- создайте дамп данных и структуры базы данных: mysqldump -u root --max-allowed-packet=1G --single-transaction orchestrator > /orchestrator db dump.sql

Примечание: β случае, если изменяли стандартные доступы о базы данных, β запросе замените -и root на -и β аш_логин -p.

- выйдите из контейнера (команда "exit").
- скопируйте дамп из контейнера на хостовую машину: docker cp orchestratordb:/orchestrator db dump.sql/путь куда осуществляем копирование

4.5 Sherpa Orchestrator: ответы на часто задаваемые вопросы

Вопрос: что делать, если есть проблема с нехваткой памяти для скрипта? (пример на скриншоте)



Ответ:

- 1) В случае использования версии с Docker: нужно в backend\config\php.ini указать memory_limit = 2048М или выше. Затем необходимо перезапустить контейнер.
- 2) В случае использования Local версии: нужно в файле php.ini (может находится по разным адресам в зависимости от используемой ОС) указать memory_limit = 2048М или выше. Затем необходимо перезапустить nginx.

Вопрос: как подключить phpMyAdmin (в версии с Docker)?

Ответ:

1) Скачайте архив и скопируйте его на сервер в /opt.

https://sherparpa.ru/downloads/private/SherpaRPAOrcDocker/phpmyadmin.tar.gz

2) Загрузите образ в докер.

cd /opt/SherpaOrchestrator docker load -input phpmyadmin.tar.gz
cd /opt/SherpaOrchestrator sudo wget -O phpmyadmin.tar.gz
https://sherparpa.ru/downloads/private/SherpaRPAOrcDocker/phpmyadmin.tar.gz docker load -input phpmyadmin.tar.gz

- 3) Раскомментируйте все строки касательно phpmyAdmin в «docker-compose.yml».
- 4) Перезапустите контейнеры:

cd /opt/SherpaOrchestrator sudo ./run.sh

- 5) Зайдите по адресу http://ваш_адрес_оркестратора:8081
- 6) Для авторизации укажите:

server: orchestrator-db

username: root



Вопрос: как перезапустить контейнер?

Ответ: Контейнеры не рекомендуется перезапускать вручную через команду Docker.

Правильным методом перезапуска для докер-версии является скрипт run.sh (запуск из директории установки по умолчанию /opt/SherpaOrchestrator). То есть, в консоли необходимо перейти в директорию установки (команда cd адрес_папки) и выполнить команду sudo ./run.sh (иногда sudo использовать не нужно).

Вопрос: что делать, если есть проблемы со скоростью обработки при большом количестве одновременно работающих роботов?

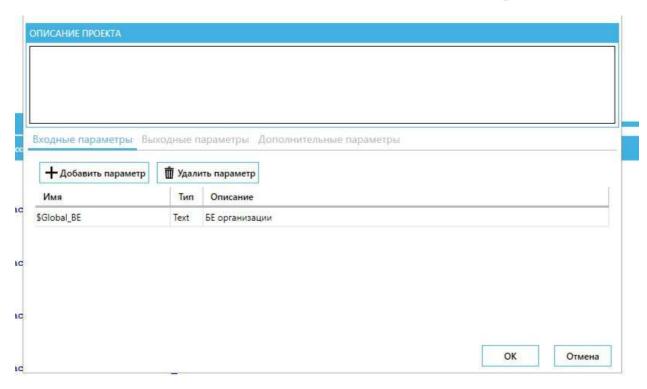
Ответ: В случае, если много одновременно работающих роботов (если есть проблемы с скоростью запросов), то в конфиге

backend/config/php-fpm.conf

параметр рт должен быть выставлен static (в новых сборках Оркестратора он выставлен так по умолчанию). Чтобы Оркестратор быстрее обрабатывал множество запросов (heartbeat) от большого числа одновременно работающих Роботов, параметр рт.max_children можно увеличить (по умолчанию стоит 8). При внесении любых изменений в конфиг, необходимо перезапустить контейнер. Если у Вас локальная установка без Docker, то необходимо найти у себя путь к рhp-fpm.conf, внести изменения, указанные выше и перезапустить php-fpm.

Вопрос: подскажите пожалуйста входной параметр возможно назначать в оркестраторе при создании работы или при определении тригтера?





Ответ: в Оркестраторе создайте задачу в какой-нибудь очереди и в параметры этой задачи запишите те параметры, которые хотите передавать. Затем в Триггере Оркестратора на вкладке "Очередь" выберите пункт "Назначить задачу по умолчанию создаваемым работам" и выберите ту самую задачу. После этого в сценарии робота используйте блок "Получить задачу по умолчанию" из палитры Оркестратор, чтобы получить Ваши параметры.

Вопрос: что делать, если из Оркестратора пропали Роботы (или другие объекты)?

Ответ: возможно они были по ошибке удалены. Большинство записей в базе данных удаляются через soft delete, то есть им в колонке is_deleted просто проставляется 1, а запись при этом остаётся в базе. Если что-то пропало - можно вернуть вручную через базу, найдя необходимый объект и поменяв is_deleted на 0.

Вопрос: подскажите как увеличить время сессии в веб-версии Оркестратора? Сейчас, как я понял, 15 мин по умолчанию?

Oтвет: можно поднять до 30 минут, для этого внесите в конфиг backend/config/php.ini следующие параметры:



```
session.gc_maxlifetime = 1800
session.cookie lifetime = 1800
```

После - необходимо перезапустить контейнеры.

Можно установить и большие значения (в секундах), но на уровне скрипта реализовано прекращение сессии при бездействии более 30 минут. Данный способ актуален для версии с Docker. Для версии без Docker, соответственно, надо вносить изменения в те же параметры (но путь к конфигу будет другим и будет зависеть от используемой ОС). И в таком случае после внесения изменений надо будет перезапустить веб-сервер.

5 Шаблонизатор Sherpa RPA

Шаблонизатор представляет собой инструмент создания шаблонов (визуальный редактор) и является неотъемлемой частью Дизайнера Sherpa RPA. Шаблонизатор не привязан к каким-либо облачным сервисам и входит в комплект поставки.

Основные возможности

- Определение типа документа (текстовый, табличный; одно- или многостраничный и т.д.). В случае, когда роботу требуется создать несколько шаблонов по различным типам документа (например, Акт отгрузки товара, Акт оказания услуг), Шаблонизатор автоматически подбирает наиболее подходящий шаблон из списка;
- Исправление перспективы и перекосов скан-копии (фото) обрабатываемого документа;
 - Зонирование и сегментация документов;
- Автоматический поиск и разметка таблиц, в том числе без видимых границ;
 - Динамическая привязка содержимого;
 - Проверка наличия печатей, подписей, рукописных текстов;



• Извлечение корректной информации из скан-копии/фото документа.

Обработка сложных документов

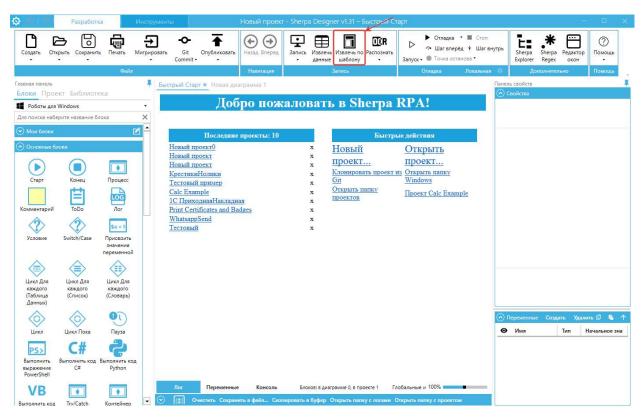
Процесс обработки сложных документов выглядит следующим образом:

№ п/п	Этап обработки	Описание
1.	Коррекция и типизация	Определение типов документов, разбор по типам и страницам Коррекция перекосов и перспективы Зонирование и сегментация
2.	Предобработка Шаблонизатором	Большая часть шаблона создаётся автоматически Автоподбор подходящего шаблона Автоопределение границ таблиц
3.	Обработка OCR	Онлайн или офлайн 5 модулей (движков) на выбор Внешние или встроенные в платформу
4.	Постобработка документа	ML & NLP Man in the loop Проверка наличия подписей, печатей, галочек, рукописных текстов
5.	Занесение в ИС	SAP Oracle R12 1C Битрикс 24 и другие

5.1 Интерфейс Шаблонизатора

Запуск Шаблонизатора осуществляется на верхней панели основного меню Дизайнера во вкладке «Разработка» — раздел «Запись» — «Извлечь по шаблону».



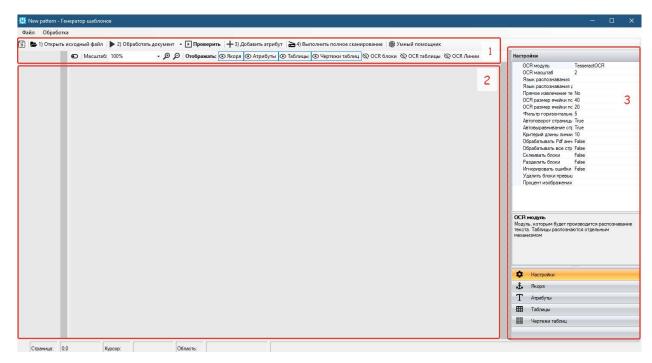


При нажатии на кнопку «Извлечь по шаблону» открывается обработка «Генератор шаблонов» (далее Шаблонизатор).

Пользовательский интерфейс Шаблонизатора состоит из следующих основных областей:

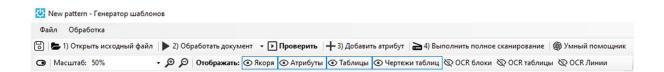
- 1) Основное меню (верхняя панель);
- 2) Центральная рабочая область область для работы с шаблоном обрабатываемого документа;
 - 3) Панель настроек.





5.1.1 Основное меню Шаблонизатора

Верхняя панель основного меню позволяет увидеть данные, вычисленные встроенным OCR — модулем, а также предлагает основные действия для создания и обработки шаблонов.





Элементы интерфейса основного меню Шаблонизатора.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	меню «Файл»	Позволяет использовать базовые функции по управлению файлами документов и шаблонов.
2.	меню «Обработка»	Позволяет производить основные действия по обработке документа.
3.	кнопка «Открыть исходный файл»	Позволяет выбрать и открыть ранее сохраненный сканированный документ, который будет использоваться в качестве образца при создании шаблона для распознавания роботом определенного типа документа.
4.	кнопка «Обработать документ»	Позволяет запустить обработку документа по выбранным параметрам для отображения (Якоря, Атрибуты, Таблицы, Чертежи таблиц, ОСК блоки, ОСК таблицы, ОСК линии).
5.	кнопка «Проверить»	Позволяет визуально обновить макет шаблона после обработки документа и внесения каких-либо правок в шаблон (создание/редактирование якоря/атрибута/таблицы) без повторной отправки документа на обработку ОСR.
6.	кнопка «Добавить атрибут»	Позволяет определить и обозначить области документа, данные которых будут распознаваться роботом и импортироваться в таблицу «Результаты».
7.	кнопка «Выполнить полное сканирование»	Позволяет выполнить полное сканирование.
8.	кнопка «Умный помощник»	Позволяет проанализировать документ и создать шаблон для документа.



9.	переключатель «Масштаб»	Позволяет задать масштаб при работе со сканированным документом.
10.	Меню «Отображать»	Меню «Отображать» позволяет управлять отображением якорей, атрибутов, таблиц и чертежей таблиц, ОСК блоков, ОСК таблиц, ОСК линий в рабочей области.

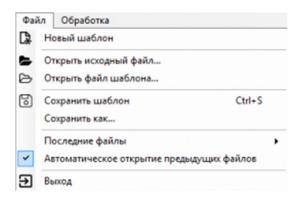
5.1.1.1 Меню «Файл»

Элементы интерфейса основного меню Шаблонизатора. Меню «Файл».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	функция «Новый шаблон»	Позволяет создать новый шаблон.
2.	функция «Открыть исходный файл»	Позволяет выбрать исходный файл для создания шаблона. При нажатии на кнопку открывается окно для поиска исходного файла с возможность указать путь и название файла.
3.	функция «Открыть файл шаблона»	Позволяет выбрать один из ранее созданных шаблон по указанному пути.
4.	функция «Сохранить шаблон»	Позволяет сохранить созданный шаблон по указанному пути. Имя шаблона задается вручную. Также, для сохранения можно воспользоваться сочетанием клавиш Ctrl+S или нажать на в левом верхнем углу панели основного меню. Имя файла нужно указывать, исходя из вида документа, например, «ТОРГ 12». Расширение сохраняемого файла .json
5.	функция «Сохранить как»	Позволяет сохранить созданный шаблон по указанному пути. Имя шаблона задается вручную. Расширение сохраняемого файла .json



6.	функция «Последние файлы»	Позволяет выбрать один из ранее сохраненных файлов из выпадающего списка.
7.	функция «Автоматическое открытие предыдущих файлов»	Позволяет включить автоматического открытие предыдущих файлов. При каждом открытии Шаблонизатора автоматически открывает файл, с которым велась работа до закрытия окна Шаблонизатора. Для активации функции необходимо кликнуть левой клавишей мыши по строке до появления
8.	функция «Выход»	Позволяет выйти из Шаблонизатора.



5.1.1.2 Меню «Обработка»

Элементы интерфейса основного меню Шаблонизатора. Меню «Обработка».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	функция «Обработать документ»	Позволяет выполнить обработку текущего файла по заданным инструментам и атрибутам, выбранным в верхней панели основного меню (Якоря, Атрибуты, Таблицы, Чертежи таблиц, ОСК-блоки, ОСК-таблицы, ОСК-линии). Также, для обработки документа можно воспользоваться сочетанием клавиш Ctrl+R.

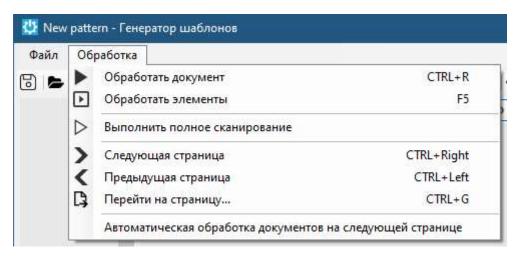


2.	функция «Обработать элементы»	Позволяет выбрать один из ранее созданных шаблонов по указанному пути для его дальнейшей обработки по заданным инструментам и атрибутам, выбранным в верхней панели основного меню. Также, для обработки элементов можно нажать F5.
3.	функция «Выполнить полное сканирование»	Позволяет выполнить полное сканирование. После выполнения действия открывается окно «Результат полного сканирования».
4.	функция «Следующая страница»	Позволяет перейти на следующую страницу в случае, если документ многостраничный. Также, для перехода на следующую страницу можно воспользоваться сочетанием клавиш Ctrl+Right.
5.	функция «Предыдущая страница»	Позволяет перейти на предыдущую страницу в случае, если документ многостраничный. Также, для перехода на следующую страницу можно воспользоваться сочетанием клавиш Ctrl+Left.
6.	функция «Перейти на страницу»	Позволяет перейти на страницу по ее номеру. Также, для перехода на страницу по ее номеру можно воспользоваться сочетанием клавиш Ctrl+G.
7.	функция «Автоматическая обработка документов на следующей странице»	Позволяет выполнить автоматическую обработку следующей открытой страницы по заданным инструментам и атрибутам, выбранным в верхней панели основного меню для текущей страницы. Используется в случае, когда в одном файле находится несколько однотипных одностраничных документов, размещенных на одной странице (например, документ ТОРГ 12). При нажатии кнопки открывается Следующая страница и распознается следующий документ.



При работе с многостраничным документом, содержащим в себе разные типы документации, не рекомендуем использовать эту функцию, т.к. робот будет распознавать все якоря, созданные на каждой странице документа.

Многократный переход между страницами может увеличивать время работы робота.

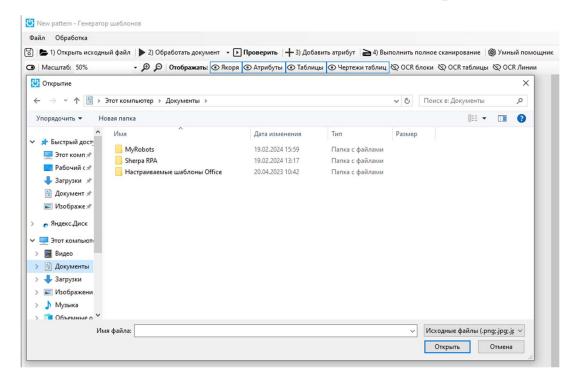


5.1.1.3 Открыть исходный файл

Шаблонизатор позволяет выбрать и открыть ранее сохраненный сканированный документ, который будет использоваться в качестве образца при создании шаблона для распознавания роботом определенного типа документа.

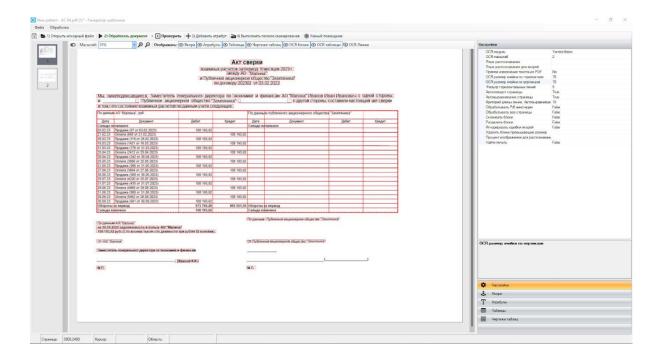
Файл может быть представлен в одном из форматов: .pdf, .png, .jpeg, .jpg. Файл .pdf может быть многостраничным.





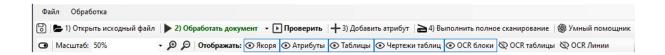
5.1.1.4 Обработать документ

Кнопка «Обработать документ» позволяет запустить обработку документа по выбранным параметрам для отображения якорей, атрибутов, таблиц, чертежей таблиц, ОСR блоков, ОСR таблиц, ОСR линий в рабочей области. Обработанный документ выглядит как показано на скриншоте.

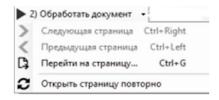




Также, можно выбрать один или несколько критериев на верхней панели основного меню Шаблонизатора.



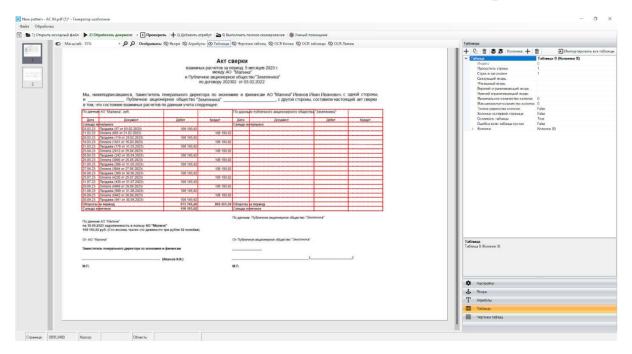
При нажатии на кнопку появляется возможность произвести обработку следующей страницы (Ctrl+Right) или предыдущей страницы (Ctrl+Left), перейти на страницу (Ctrl+G). Кнопка «Открыть страницу повторно» позволяет открыть исходный документ без внесенных изменений в настройках Шаблонизатора. Данная кнопка используется для проверки корректности заданных настроек в Шаблонизаторе. Таким образом можно проверить, например, срабатывает ли настройка «Автоповорот страницы». Если при нажатии кнопки «Открыть страницу повторно» мы видим, что документ изначально был неверно повернут, значит настройки Шаблонизатора были заданы верно и при обработке сработал Автоповорот страницы.



5.1.1.5 Проверить

Кнопка «Проверить» позволяет проверить корректность отображения табличной части, из которой будут импортироваться данные. Также проверку можно запустить нажатием клавиши F5.



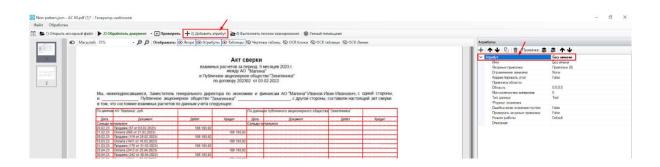


5.1.1.6 Добавить атрибут

Атрибуты позволяют определить и обозначить области документа, данные из которых будут распознаваться роботом и импортироваться в таблицу «Результаты».

Существует несколько способов добавления Атрибута:

1) Нажатием кнопки на верхней панели Основного меню.



2) Нажатием кнопки + на правой рабочей панели меню во вкладке «Атрибуты».





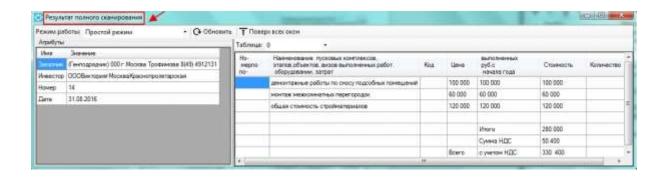
5.1.1.7 Выполнить полное сканирование

Кнопка «Выполнить полное сканирование» позволяет выполнить полное сканирование документа. После нажатия на кнопку открывается окно «Результат полного сканирования», в котором отображаются распознанные данные из документа в виде заполненной таблицы. Если при проверке обнаружены неточности, то их, в любой момент, можно исправить.



5.1.1.7.1 Результат полного сканирования

Результат полного сканирования представляет собой окно с результатами, полученными в процессе работы робота со сканированным документом по созданному шаблону. В появившейся форме отображаются все атрибуты и таблицы, обнаруженные в документе.



Окно с результатами поделено на две табличные части:

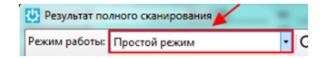
1) Атрибуты (слева) - в данной табличной части отображаются данные, полученные роботом по заданным Атрибутам и Якорям. Чаще всего, это общая информация о Заказчике, Подрядчике, Инвесторе, например, наименование и реквизиты. Также в Атрибутах указываются Номер и Дата документа.



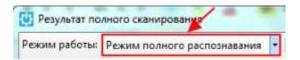
2) Таблица (справа) – в табличной части отображаются данные, содержащиеся в таблицах сканированного документа, полученные роботом в результате работы по созданному шаблону при помощи ОСR-модулей.

Эти данные могут быть числового формата, например, цена, стоимость, сумма и т.д.

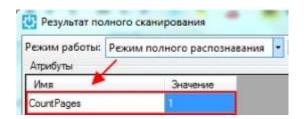
По умолчанию указан режим «Простое сканирование».



При работе с многостраничными документами (если файл содержит в себе два и более документа), на одной из страниц которого указан якорь «LastPage», необходимо использовать «Режим полного распознавания».



При этом будет отображаться выходной параметр – количество страниц, которые участвовали в работе Шаблонизатора.



Полное распознавание будет проводиться с текущей страницы до момента, пока не будет найден якорь «LastPage».

В случае, если мы запускаем «Полное сканирование», находясь не на первой странице многостраничного документа, и, если на одной из страниц указан якорь «LastPage», Шаблонизатор выдаст окно с запросом: «Вы хотите выполнить распознавание с первой страницы?».



При нажатии кнопки «Да» — алгоритм Шаблонизатора будет распознавать документ с первой страницы документа до момента, пока не будет найден якорь «LastPage».

При нажатии кнопки «Нет» — алгоритм Шаблонизатора будет распознавать документ с текущей страницы документа до момента, пока не будет найден якорь «LastPage».



5.1.1.8 Умный помощник

Кнопка «Умный помощник» позволяет проанализировать документ и создать шаблон для документа.



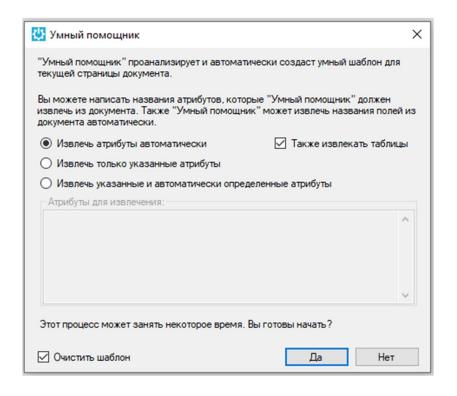
При нажатии на кнопку «Умный помощник» доступны следующие настройки.

Элементы интерфейса основного меню Шаблонизатора. Умный помощник. Настройки.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	настройка «Извлечь атрибуты автоматически»	Позволяет роботу определить значения атрибутов автоматически, без прямого указания.



2.	настройка «Извлечь только указанные атрибуты»	Позволяет задать определенные атрибуты для извлечения их из документа (например, «покупатель» или «продавец»).
3.	Настройка «Извлечь указанные и автоматические определенные атрибуты»	Позволяет указать прямо атрибуты, необходимые для извлечения, а также позволить роботу самому определить поля, которые присутствуют в анализируемом документе.
4.	флаг «Также извлекать таблицы»	Позволяет извлечь заголовки таблицы из документа. Если флаг не стоит, то таблица в документе рассматриваться не будет.
5.	флаг «Очистить шаблон»	Позволяет при каждом запуске очистить в шаблоне Якоря, Атрибуты, Чертежи, Чертежи таблиц.



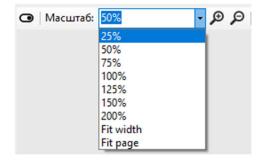
Умный помощник поможет в работе с Шаблонизатором: он покажет, как анализируется документ и создается шаблон.



5.1.1.9 Масштаб

Переключатель «Масштаб» позволяет задать масштаб при работе со сканированным документом. Размер изображения может быть задан несколькими способами:

- Выбрать значение из выпадающего списка;
- Указать значение вручную;
- Увеличить или уменьшить масштаб при помощи лупы.



5.1.1.10 Меню «Отображать»

Меню «Отображать» позволяет управлять отображением якорей, атрибутов, таблиц и чертежей таблиц, ОСК блоков, ОСК таблиц, ОСК линий в рабочей области.



Элементы интерфейса основного меню Шаблонизатора. Меню «Отображать».

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	Якоря	Отображает точки на сканированном
		документе, с помощью которых робот
		определяет, в какой части документа
		размещаются определенные данные.
		Якорей может быть сколь угодно много, в
		зависимости от типа документа.
1		



2.	Атрибуты	Отображает области документа, данные из которых будут распознаваться роботом и импортироваться в таблицу «Результаты».
3.	Таблицы	Отображает границы таблиц, данные из которых будут распознаваться роботом и импортироваться в таблицу «Результаты».
4.	Чертежи таблиц	Отображает чертежи таблиц.
5.	OCR блоки	Отображает блоки, распознанные OCR- модулем.
6.	OCR таблицы	Отображает таблицы, распознанные ОСR- модулем.
7.	OCR линии	Отображает линии, распознанные OCR- модулем.

Кнопки Якоря, Атрибуты, Таблицы и Чертежи таблиц относятся напрямую к шаблону. Данные, отображаемые по ним, являются вычисленными после обработки шаблона.



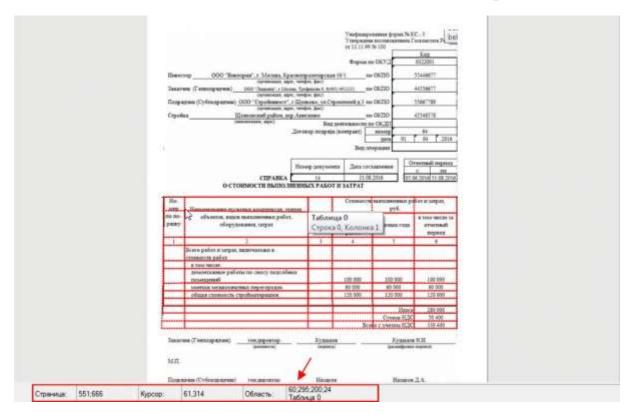
Кнопки ОСR блоки, ОСR таблицы и ОСR линии используются перед обработкой шаблона для настройки шаблона. Они отображают данные, полученные после обработки документа ОСR модулем.



5.1.2 Центральная рабочая область Шаблонизатора

Область для работы с шаблоном обрабатываемого документа. Здесь отображается исходный файл/сканированный документ, по которому создается шаблон для дальнейшей работы робота.





При наведении курсора на нижней панели отображаются координаты страницы, курсора и области. Это удобно, когда нужно проверить или скорректировать координаты какого-либо атрибута.

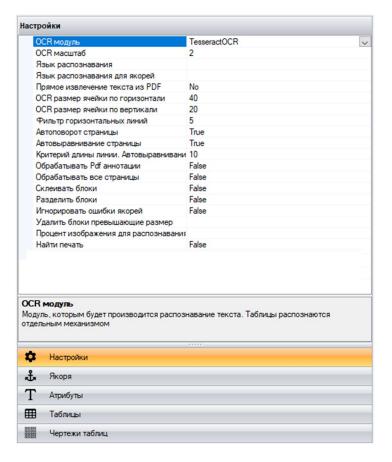
В центральной рабочей области отображаются результаты всех манипуляций, которые мы производим на верхней панели основного меню и на боковой панели настроек Шаблонизатора.

5.1.3 Панель настроек

В правой панели основного меню Шаблонизатора находятся 5 основных блоков для создания и настройки шаблона:

- Настройки;
- Якоря;
- Атрибуты;
- Таблицы;
- Чертежи таблиц.





5.1.3.1 *Настройки*

Шаблонизатор использует несколько видов модулей для извлечения данных из исходных файлов.

OCR модуль

ОСК модуль — модуль, которым будет производиться распознавание текста.

В платформу Sherpa RPA интегрированы несколько ОСR-модулей. Два из них поставляются вместе с роботом: Tesseract OCR и Microsoft OCR. Данные модули могут работать офлайн, без подключения к интернету).

Tesseract OCR — это движок оптического распознавания символов (OCR) с открытым исходным кодом, является самой популярной и качественной OCR-библиотекой. OCR использует нейронные сети для поиска и распознавания текста на изображениях.

Yandex Vision и ABBYY OCR – онлайн-модули, которые используют функции соответствующих облачных сервисов.



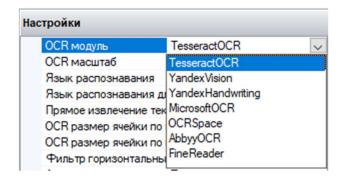
Yandex Handwriting — модуль, позволяющий распознавать рукописный текст.

Microsoft OCR — модуль, позволяющий распознавать текст на изображениях и сканированных документах с помощью оптического распознавания символов (OCR).

OCR Space — модуль, позволяющий распознавать кириллический шрифт на изображениях формата .jpg.

ABBYY FineReader – это коммерческий оффлайн-модуль, использование которого требует отдельной лицензии.

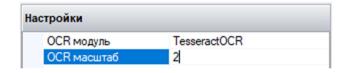
Платформа Sherpa RPA позволяет настроить работу сценария с распознаванием изображений и переключаться между этими ОСR-модулями в любой момент времени.



ОСК масштаб

OCR масштаб — параметр, позволяющий улучшить качество распознавания, если документы имеют плохое разрешение.

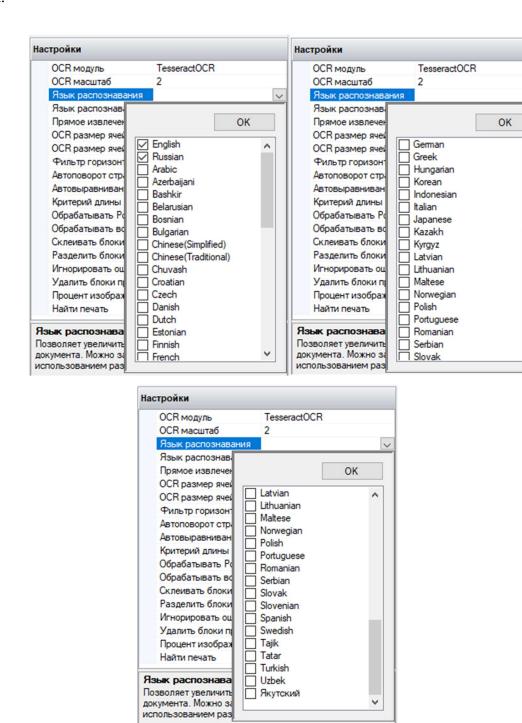
В случае высокого и среднего качества сканированного документа рекомендуется оставить значение масштаба равное «2».





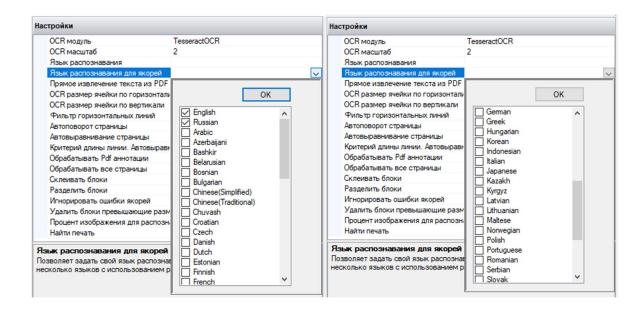
Язык распознавания – параметр, позволяющий увеличить точность распознавания документа. Также, можно задать несколько языков, использовав в качестве разделителя запятые.

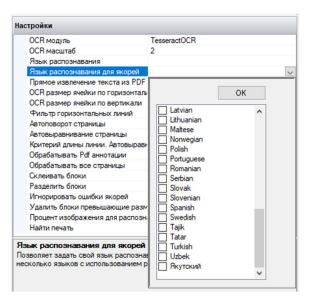
При выборе настройки «Язык распознавания» открывается выпадающий список, в нем с помощью флагов можно отметить нужные для распознавания языки.





Язык распознавания – параметр, позволяющий указывать конкретный язык распознавания для якорей. Также, можно задать несколько языков, использовав в качестве разделителя запятые.





Прямое извлечение текста из PDF

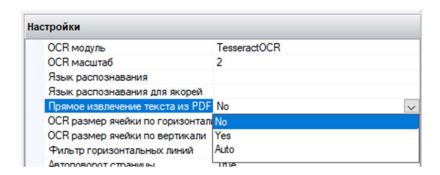
Прямое извлечение текста из PDF – параметр, позволяющий управлять прямым извлечением текста со страницы.

Возможные значения:

- No отключено;
- Yes используется только прямое извлечение текста;



• Auto - автоматический режим (если на странице нет текста, то будет реализовано распознавание текста с заданным ОСК модулем).

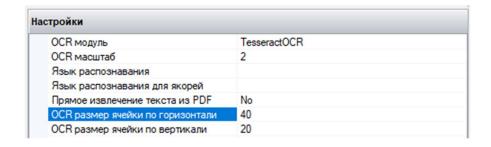


OCR размер ячейки по горизонтали

OCR размер ячейки по горизонтали – параметр, позволяющий указать делитель по горизонтали, по которому определяются ячейки таблиц на странице. Значение должно быть больше или равно 1.

Размер ячейки определяется как размер изображения, деленное на это значение.

Шаблонизатор использует два параметра (по горизонтали и по вертикали) для более точного определения таблиц.



По умолчанию заданы параметры:

40 - OCR размер ячейки по горизонтали

Это значение является оптимальным при распознавании документов со стандартными (или близкими к стандартным) размерами ячеек таблиц.



Рекомендуем оставлять эти значения без изменений и менять их только в случае некорректного распознавания (после проверки).

Если при заданных параметрах (значение по горизонтали 40) таблица не будет распознана, то алгоритм Шаблонизатора увеличит это значение на 10 автоматически и осуществит попытку повторно.

OCR размер ячейки по вертикали

OCR размер ячейки по вертикали – параметр, позволяющий указать делитель по вертикали, по которому определяются ячейки таблиц на странице. Значение должно быть больше или равно 1.

Размер ячейки определяется как размер изображения, деленное на это значение.

Шаблонизатор использует два параметра (по горизонтали и по вертикали) для более точного определения таблиц.

астройки		
OCR модуль	TesseractOCR	
OCR масштаб	2	
Язык распознавания		
Язык распознавания для якорей		
Прямое извлечение текста из PDF	No	
OCR размер ячейки по горизонтали	40	
OCR размер ячейки по вертикали	20	

По умолчанию заданы параметры:

20 — OCR размер ячейки по вертикали.

Это значение является оптимальным при распознавании документов со стандартными (или близкими к стандартным) размерами ячеек таблиц.

Рекомендуем оставлять эти значения без изменений и менять их только в случае некорректного распознавания (после проверки).



Если при заданных параметрах (значение по вертикали 20) таблица не будет распознана, то алгоритм Шаблонизатора увеличит это значение на 10 автоматически и осуществит попытку повторно.

Фильтр горизонтальных линий

Фильтр горизонтальных линий – параметр, указывающий процент, меньше которого линии будут проигнорированы. Определяется как отношение горизонтальной линии к ширине изображения. Данный параметр используется для атрибутов.

астройки		
OCR модуль	TesseractOCR	
OCR масштаб	2	
Язык распознавания		
Язык распознавания для якорей		
Прямое извлечение текста из PDF	No	
OCR размер ячейки по горизонтали	40	
OCR размер ячейки по вертикали	20	
Фильтр горизонтальных линий	5	
	_	

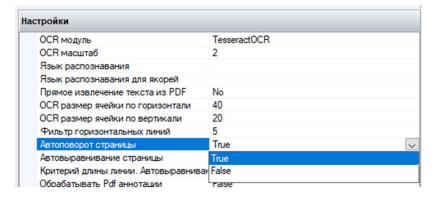
По умолчанию значение данного параметра равно 5.

Часто в сканированных документах присутствуют линии, не являющиеся границами таблиц (различные артефакты, которые необходимо отфильтровать). При заданном параметре все лишние артефакты (линии, нарисованные ручкой или линии, появившиеся при некачественном сканировании) будут проигнорированы.

Автоповорот страницы

Автоматический поворот страницы – параметр, позволяющий автоматически разворачивать страницы на угол, кратный 90 градусам (90, 180 и 270). Автоповорот страницы не разворачивает документ на малые углы. По умолчанию значение параметра «True», рекомендуем оставлять его без изменения.

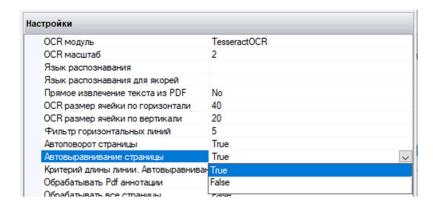






Автовыравнивание страницы

Автоматическое выравнивание страницы – параметр, позволяющий выравнивать содержимое страницы при некорректном сканировании листа. В отличие от Автоповорота страницы, Автовыравнивание разворачивает документ на малые углы.



При автовыравнивании страницы распознается самая длинная ОСК линия, имеющаяся на странице (чаще всего это линия из таблицы или атрибута) и определяет угол ее наклона относительно горизонтали. После этого происходит разворот документа, чтобы найденная линия стала параллельна линии горизонта.

По умолчанию значение параметра «True», рекомендуем оставлять его без изменения.

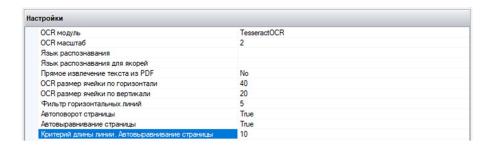
Этот параметр нежелательно использовать в случае, если сканированный документ сильно повернут и угол поворота составляет более 40, в этом случае Шаблонизатор не сможет распознать, в какую сторону нужно выровнять документ и выдаст ошибку. В такой ситуации можно использовать параметр «Корректировка угла».

Критерий длины. Автовыравнивание страницы

Критерий длины. Автовыравнивание страницы – параметр, позволяющий ограничивать длину линии. Данный параметр используется как делитель для ширины страницы. Если длина линии меньше вычисленного значения, то она

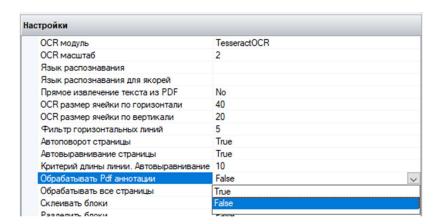


пропускается в алгоритме. Если страница изначально сильно повернута, то следует указать или большее значение или 0 для отключения параметра. По умолчанию значение равно 10.



Обрабатывать PDF аннотации

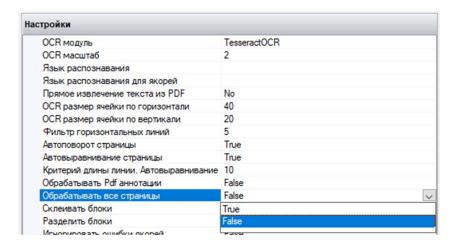
Обрабатывать PDF аннотации – параметр, позволяющий включить обработку аннотаций PDF-файла.



Обрабатывать все страницы

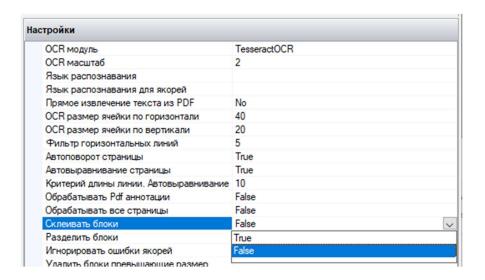
Обрабатывать все страницы – параметр, позволяющий включить обработку всех страниц в документе. Этот режим не делает проверку якоря LastPage.





Склеивать блоки

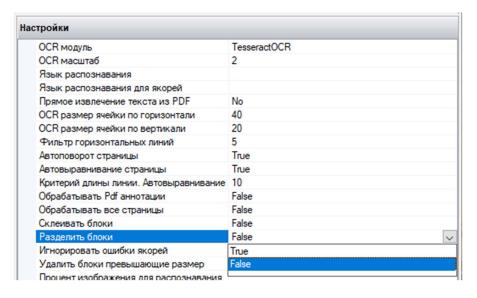
Склеивать блоки – параметр, позволяющий склеивать рядом стоящие блоки в один блок.



Разделить блоки

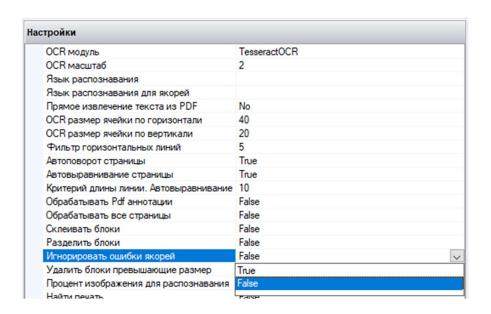
Разделить блоки – параметр, позволяющий разделить блоки содержащие пробелы на блоки без пробелов.





Игнорировать ошибки якорей

Игнорировать ошибки якорей – параметр, позволяющий отключить генерацию ошибок, если якори (любые) не найдены. В этом случае, область якоря считается нулевой.



Удалить блоки превышающие размер

Удалить блоки превышающие размер – параметр, позволяющий удалить блоки превышающие указанный размер. Поле для ввода расположено рядом с наименованием параметра. Если указать одно число, то будут удалены блоки, у которых ширина или высота будут больше этого значения. Также, можно указать значения через запятую в формате: ширина, высота.



OCR модуль	TesseractOCR
OCR масштаб	2
Язык распознавания	
Язык распознавания для якорей	
Прямое извлечение текста из PDF	No
OCR размер ячейки по горизонтали	40
OCR размер ячейки по вертикали	20
Фильтр горизонтальных линий	5
Автоповорот страницы	True
Автовыравнивание страницы	True
Критерий длины линии. Автовыравнивание	10
Обрабатывать Pdf аннотации	False
Обрабатывать все страницы	False
Склеивать блоки	False
Разделить блоки	False
Игнорировать ошибки якорей	False

Процент изображения для распознавания

Процент изображения для распознавания – параметр, позволяющий указать процент изображения, который пойдет на распознавание ОСR движком. Поле для ввода расположено рядом с наименованием параметра. Значение необходимо записывать как одно число или два числа через тире.

Например:

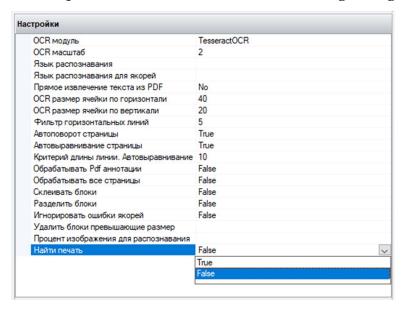
30 (0-30) - будет распознано 0-30% изображения; 30-70 - будет распознана часть изображения; 70-100 - будет распознано нижние 30% изображения.

OCR модуль	TesseractOCR
OCR масштаб	2
Язык распознавания	
Язык распознавания для якорей	
Прямое извлечение текста из PDF	No
OCR размер ячейки по горизонтали	40
OCR размер ячейки по вертикали	20
Фильтр горизонтальных линий	5
Автоповорот страницы	True
Автовыравнивание страницы	True
Критерий длины линии. Автовыравнивание	10
Обрабатывать Pdf аннотации	False
Обрабатывать все страницы	False
Склеивать блоки	False
Разделить блоки	False
Игнорировать ошибки якорей	False

Найти печать



Найти печать – параметр, позволяющий включить поиске печатей на документе. В атрибуты будет добавлен ключ "Stamps", который возвратит массив объектов StampItem со свойствами X, Y, Width, Height, PageIndex.



5.1.3.2 Якоря

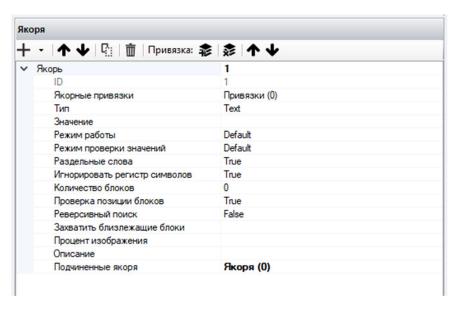
Якоря — блоки на обр аботанном документе, с помощью которых робот определяет, в какой части документа размещаются определенные данные.

При помощи якорей задаются границы определенных атрибутов. Якорей может быть неограниченное количество, в зависимости от типа документа.

Инструменты для работы с якорями

При создании нового якоря на рабочей панели открывается список свойств и инструментов для работы с якорем.





Основные инструменты для работы с якорями.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	кнопка + •	Позволяет добавить новый якорь. Кнопка , которая находится рядом позволяет добавить новый подчиненный якорь, при этом нужно сделать клик на свойство "Подчиненные якоря" у того якоря, которому нужно добавить подчиненный якорь
2.	кнопка 🛧	Позволяет переместить якорь вверх.
3.	кнопка♥	Позволяет переместить якорь вниз.
4.	кнопка 🤼	Позволяет копировать выделенный якорь и автоматически добавить новый якорь вместе с заданными настройками.
5.	кнопка 🛅	Позволяет удалить выделенный якорь.
6.	кнопка 🕏	Позволяет добавить новую привязку к выделенному якорю.



7.	кнопка 🕏	Позволяет удалить выделенную привязку.
8.	кнопка 🔨	Позволяет переместить привязку вверх.
9.	кнопка♥	Позволяет переместить привязку вниз.



Настройки якоря

Параметры настроек якоря.

№ п/п	Параметр	Значение параметра
1.	ID	Идентификатор якоря. Присваивается автоматически.
2.	Якорные привязки	Список якорных привязок.
3.	Тип	 Тип якоря. Выбирается из выпадающего списка: Техt - текст; Table - таблица или ячейка таблицы; Раскеt - пакет взаимозаменяемых якорей (при привязке к этому якорю будут взяты его подчиненные якоря, первый успешно найденный из списка).



4.	Значение	Значение якоря. Для якоря типа Text возможно указывать подстановочные знаки (*)(?)
		Для якоря типа Table значение записывается в формате «Индекс таблицы:Индекс строки:Индекс
		колонки». Например, для Text «*обл* страх*» (два
		слова при включении «Раздельные слова»), для
		Table – «0:1:2» (таблица 0, строка 1, колонка 2),
		«1:2» (таблица 1, строка 2), «2» (вся таблица 2)
5.	Режим работы	Режим работы якоря. Возможные значения:
		• Default – якорь обязателен на первой
		странице многостраничного документа;
		• LastPage – якорь будет обязателен в
		многостраничном документе и укажет на
		последнюю страницу;
		• Optional - необязательный якорь и если он
		не будет найден на странице, то ошибка не
		генерируется.
6.	Режим	Выбор режима проверки значений. Возможные
	проверки	значения:
	значения	• Default (по умолчанию);
		• EveryWordContains (проверка каждого слова
		в значении как вхождение в строку.
		Пример: "тест" => "*тест*");
		• RemoveNonWordSumbols (при проверке
		удаляются все не числовые и буквенные
		символы).



7. 8.	Раздельные слова Игнорировать регистр символов	При включенной настройке значение якоря разбивается на слова и каждое слово ищется отдельно. При включенной настройке будет игнорироваться регистр символов.
9.	Количество блоков	Если включена настройка «Раздельные слова», то при поиске якоря возможна ситуация, когда в заданной области могут быть найдены несколько значений блоков, соответствующие заданным словам. Данное свойство позволяет ограничить максимальное количество слов между блоками.
10.	Проверка позиции блоков	При включенной настройке проверяется позиция блоков относительно координаты Y и берутся блоки с наименьшим значением.
11.	Реверсивный поиск	При включенной настройке поиск якоря будет идти снизу страницы.
12.	Захватить близлежащие блоки	Захват ближних блоков якоря. Позволяет захватить в якорь ближние блоки от уже захваченного значения. Например, если указать +2-1, то к уже захваченному якорю будут присоединены ближние блоки: 2 блока справа и 1 блок слева. Данное свойство может использоваться, когда не удается установить полное устойчивое значение для якоря, но часть его имеет устойчивое значение.



13.	Процент изображения	Процент изображения, в котором будет производиться поиск якоря. Значение необходимо записывать как одно число или два числа через тире. Например:
		30 (0-30) - будут использоваться 0-30% изображения; 30-70 - будут использоваться часть изображения; 70-100 - будут использоваться нижние 30% изображения.
14.	Описание	Пользовательское описание якоря.
15.	Якоря	Якоря, напрямую зависящие от родительского якоря. Поиск подчиненных якорей будет происходит только после того, как будет найден главный/родительский якорь. В случае, если в настройках родительского якоря указан режим «Optional» (т.е. он не является обязательным), и если родительский якорь не был найден, то и поиск подчиненных якорей не будет осуществлен, они будут пропускаться. Эта параметр используется, если атрибут находится в середине многостраничного документа.

5.1.3.3 Атрибуты

Атрибуты позволяют определить и обозначить области документа, данные из которых будут распознаваться роботом и импортироваться в таблицу «Результаты».



Основные инструменты для работы с атрибутами.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	кнопка +	Позволяет добавить новый атрибут.
2.	кнопка 🔨	Позволяет переместить атрибут вверх.
3.	кнопка 🗣	Позволяет переместить атрибут вниз.
4.	кнопка	Позволяет копировать выделенный атрибут.
5.	кнопка 🗖	Позволяет удалить выделенный атрибут.
6.	кнопка 🕏	Позволяет добавить новую привязку к выделенному атрибуту.
7.	кнопка 🕏	Позволяет удалить выделенную привязку.
8.	кнопка 🄨	Позволяет переместить привязку вверх.
9.	кнопка♥	Позволяет переместить привязку вниз.



Свойства атрибутов

При создании нового атрибута на рабочей панели открывается список свойств и инструментов для работы с атрибутом.



Свойства и инструменты для работы с атрибутом.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	Имя	Имя атрибута, которое будет отражено в результатах
2.	Якорные привязки	Список якорных привязок атрибута.
3.	Ограничение линиями	При включенной настройке осуществляется поиск ограничения линиями. Атрибут может быть ограничен с любой стороны, относительно линий, распознанных ОСК модулем. Эти линии отображаются при нажатии кнопки «ОСК Линии». Возможные значения: • None – отключено; • UpDown – сверху и снизу; • Up – сверху; • Down — снизу.
4.	Корректировать угол	При включенной настройке осуществляется корректировка угла наклона области. Область корректируется, если она соприкасается с границами таблицы. При включенном значении «Автовыравнивание» корректировку угла использовать не нужно.
5.	Привязка области	Привязка к области осуществляется по 4-м параметрам: координата X, координата Y,



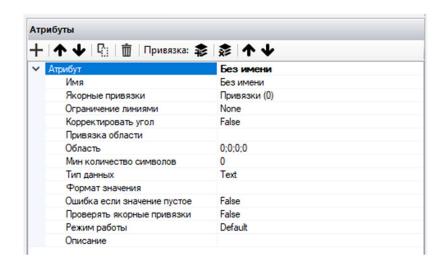
		ширина, высота области. Исходная точка области - вверху слева.
6.	Область	Прямоугольник области атрибута на странице.
7.	Мин количество символов	Минимальное количество символов, содержащихся в атрибуте. По умолчанию указано значение 0. Настройка используется, если атрибут должен содержать в себе определенное количество символов, например, атрибут ИНН. Если вычисленное значение меньше указанного будет сгенерирована ошибка.
8.	Тип данных	 Тип данных для атрибута. Возможные значения: Техt - текст; Int - число; Float - вещественное число; Date Time - дата и время; Chars - только буквы (исключаются цифры и другие символы); CountColorHSV - количество точек в заданном интервале цвета. Значение должно быть указано в формате «H1-H2;S1-S2;V1-V2», где H1, S1, V1 - это минимальные значения, а H2, S2, V2 - максимальные значения; Regex - регулярное выражение.



9.	Формат значения	Формат значения атрибута. Задается
		форматирование значения. Например:
		 если тип значения дата и время то, чтобы получить только дату, необходимо указать «dd.MM.yyyy»; если тип значения CountColorHSV, то необходимо указать интервал цвета в формате «H1-H2;S1-S2;V1-V2». Формат указывается без кавычек; Если тип значения Regex, то необходимо указать регулярное выражение.
10.	Ошибка если значение	Если при обработке шаблона значение
	пустое	атрибута пустое, то будет сгенерирована
		ошибка распознавания.
11.	Проверять якорные	Проверка якорных привязок атрибута.
	привязки	При включенной настройке будет
		производиться проверка якорных
		привязок. Если хоть одна из привязок не
		будет найдена, то атрибут, также, не будет
		найден.
12.	Режим работы	Режим работы атрибута. Возможные
		значения:
		• Default - стандартная работа атрибута (его значение будет заполнено при первом успешном определении якорей);



		 Аррend - добавление значения в атрибут при каждом успешном определении якорей на следующих страницах; Overwrite - перезапись, а не добавление значения атрибута.
13.	Описание	Пользовательское описание атрибута.



5.1.3.4 Таблицы

На рабочей панели справа отображаются все таблицы шаблона, которые были распознаны ранее. Их может быть неограниченное количество. Каждой таблице автоматически присваивается номер, нумерация начинается с 0. Также, отображается количество колонок каждой распознанной таблицы. Именно по этому значению можно определить с какой (или с какими) таблицами продолжить работу.

Основные инструменты для работы с таблицами.

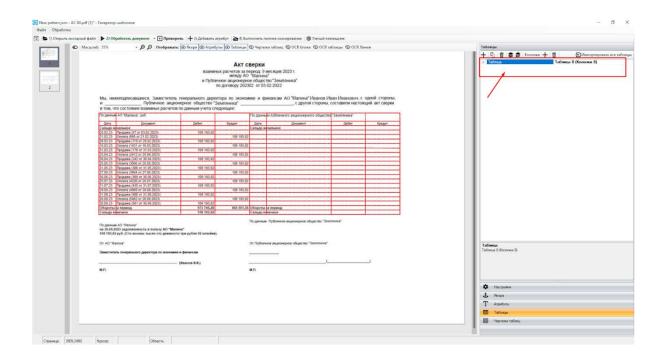
№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	кнопка +	Позволяет добавить новую таблицу.

426



2.	кнопка С	Позволяет копировать выделенную таблицу.
3.	кнопка 🗖	Позволяет удалить выделенную таблицу.
4.	кнопка 🕏	Позволяет привязать якорь к выбранному свойству.
5.	кнопка 🏂	Позволяет удалить якорь из выбранного свойства.
6.	кнопка +	Позволяет добавить колонку таблицы.
7.	кнопка 🛅	Позволяет удалить колонку таблицы.
8.	кнопка Đ	Импортировать все таблицы из документа.







Свойства таблиц

На рабочей панели открывается список свойств и инструментов для работы с таблицей.

Свойства и инструменты для работы с таблицами.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	Индекс	Индекс ОСК таблицы. Для улучшения привязки рекомендуется указывать минимальное количество колонок. В случае, если Шаблонизатор извлекает данные лишь из одной таблицы сканированного документа, необходимо указывать индекс, равный 0.
2.	Пропустить строки	При необходимости возможно указать количество первых строк, которые нужно пропустить для того, чтобы данные этих строк не попали в результаты работы робота. При выборе значения, равного 1, — первая строка будет пропущена. Наименования колонок будут соответствовать значениям первой строки. Если задать названия колонок в шаблоне, то эти значения не будут изменены.
3.	Строк в заголовке	При построении таблицы заданное количество строк будет использовано для создания заголовков колонок. При указании значения 0, названия будут взяты из шаблона колонок, в противном случае название будет суммироваться из ячеек колонки по заданному количеству строк.



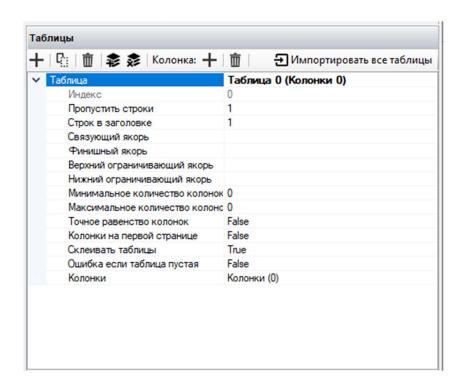
		Например, если в заголовок должна попасть информация, размещенная в двух верхних строках таблицы, — необходимо указать количество строк в заголовке, равное 2.
4.	Связующий якорь	При работе с документом, в котором есть несколько таблиц можно указать связующий якорь, имеющий тип Optional. Якорь укажет на какие-либо слово, заголовок, пункт, после которых начинается нужная таблица. При нахождении такого якоря таблица, следующая за ним, будет распознаваться.
5.	Финишный якорь	Финишный якорь ограничивает извлечение таблицы до якоря, который может иметь тип Optional. Таким якорем, например, может служить строка в таблице "Всего к оплате". Таблица будет извлечена из документа до финишного якоря.
6.	Верхний ограничивающий якорь	Якорь, после которого таблица начнет строиться.
7.	Нижний ограничивающий якорь	Якорь, до которого таблица будет строиться.
8.	Минимальное количество колонок	Минимальное количество колонок в ОСR таблице. Все таблицы с количеством



		KOHOHOK MAHLIHO MKASAIMIOPO
		колонок меньше указанного
		пропускаются.
9.	Максимальное	Максимальное количество колонок в OCR
	количество колонок	таблице. Все таблицы с количеством
		больше указанного пропускаются.
10.	Точное равенство	При включенной настройке извлекаться
	колонок	будут только те таблицы, у которых
		количество колонок в документе меньше
		или равно заданному в шаблоне.
11.	Колонки на первой	При включенной настройке колонки
	странице	будут браться только с первой страницы.
		Данная настройка указывает на то, что на
		следующих страницах у таблицы нет
		заголовков.
12.	Склеивать таблицы	При включенной настройке таблицы с
		одинаковым количеством колонок будут
		склеиваться. Если настройка не включена,
		такие таблицы будут добавляться как
		новые.
13.	Ошибка если таблица	При обработке шаблона, если таблица
	пустая	пустая, то будет сгенерирована ошибка
		распознавания.
14.	Колонки	Колонки таблицы. По каждой таблице
		отображается соответствующее ей
		количество колонок. Также, здесь будут
		указываться:
		• Индекс (задается
		автоматически);



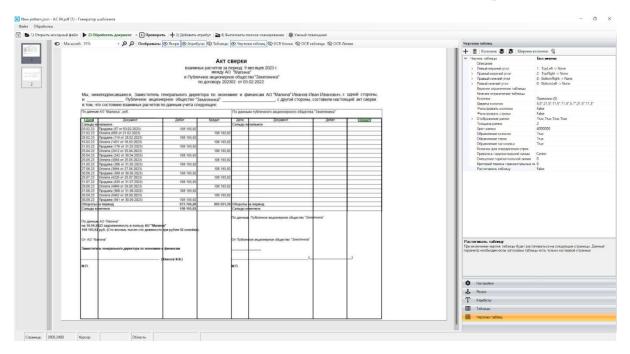
• Имя (задается пользователем);
• Тип данных (Text - текст; Int -
число; Float - вещественное число; Date
Time - дата и время; Chars - только буквы
(исключаются цифры и другие символы)).
В случае, если в результатах полного
сканирования отсутствует несколько
колонок, то мы можем их добавить.



5.1.3.5 Чертежи таблиц

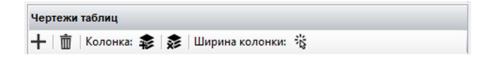
Чертежи таблиц применяются для документов, в которых по каким-либо причинам отсутствуют границы таблиц. Это могут быть иностранные документы (например, Invoice), в которых линии таблиц не предусмотрены. Также это могут быть российские документы, в которых частично не прорисованы линии таблицы, как на скриншоте ниже (отсутствуют межстрочные линии).





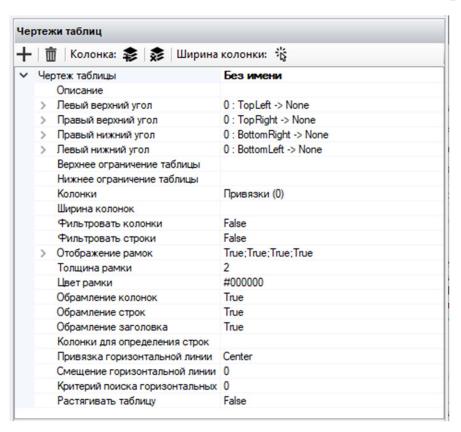
Основные инструменты для работы с чертежами таблиц.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	кнопка +	Позволяет добавить чертеж таблицы.
3.	кнопка 🗖	Позволяет удалить чертеж таблицы.
4.	кнопка 🕏	Позволяет добавить привязку колонки.
5.	кнопка 🕏	Позволяет удалить привязку колонки.
6.	кнопка	Позволяет добавить ширину колонки.



На рабочей панели справа отображаются все параметры, которые позволяют произвести настройку чертежа таблицы.





Свойства и инструменты для работы с чертежами таблиц.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	Описание	Описание чертежа таблицы.
2.	Левый верхний угол	Якорная привязка левого верхнего угла.
3.	Правый верхний угол	Якорная привязка правого верхнего угла.
4.	Правый нижний угол	Якорная привязка правого нижнего угла.
5.	Левый нижний угол	Якорная привязка левого нижнего угла.
6.	Верхнее ограничение таблицы	Ограничение таблицы сверху страницы. Можно указать значения в пикселях или процент, используя символ *. Например, 200*.
7.	Нижнее ограничение таблицы	Ограничение таблицы снизу страницы. Можно указать значения в пикселях или



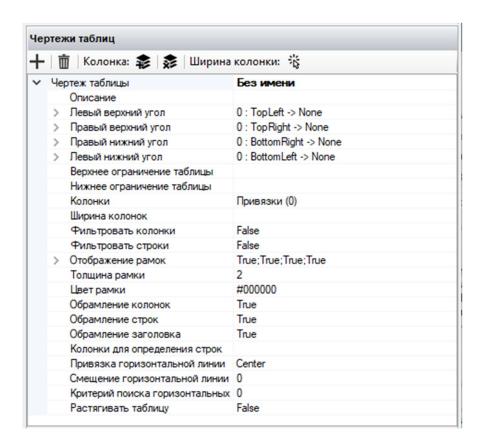
		процент, используя символ *. Например, 200*.
8.	Колонки	Якорные привязки для колонок.
9.	Ширина колонок	Ширина колонок. Можно указать фиксированное значение или процент от всей ширины таблицы в формате " 50*", " 10*". Вся ширина таблицы равна 100. Если сумма значений процентов всех указанных колонок равна или больше 100, то ширина таблицы делится пропорционально указанным значениям. Иначе, последняя колонка будет заполняться автоматически до значения 100.
10.	Фильтровать колонки	При включенной настройке для распознавания таблицы будут учитываться только те линии, которые нарисованы в чертежи таблицы.
11.	Фильтровать строки	При включенной настройке для распознавания таблицы будут учитываться только те линии, которые нарисованы в чертеже таблицы.
12.	Отображение рамок	Управление отображением рамок чертежа таблицы.
13.	Толщина рамки	Толщина рамки таблицы в пикселях.
14.	Цвет рамки	Цвет рамки таблицы в формате RGB. Например, «#000000» — черный цвет, «#FF0000» — красный цвет.



15.	Обрамление колонок	При включенной настройке отображаются линии колонок на чертеже таблицы.
16.	Обрамление строк	При включенной настройке отображаются линии строк на чертеже таблицы.
17.	Обрамление заголовка	При включенной настройке отображается первая строка на чертеже страницы.
18.	Колонки для определения строк	Индексы колонок, по которым будут определяться строки. Если значение будет пустое, то будут участвовать все колонки. Индекс начинается с 0.
19.	Привязка горизонтальной линии	Параметр привязки горизонтальной линии. Возможные значения: • Сепter - по центру между блоками текста; • Тор - верх нижнего блока текста; • Воttom - низ верхнего блока текста.
20.	Смещение горизонтальной линии	Смещение горизонтальной линии по оси Y в пикселях. Для точного смещения следует указывать значения в формате + Например, +10-20.
21.	Критерий поиска горизонтальных линий	Смещение блоков текста для определения их пересечения. По умолчанию значение равно 10.
22.	Растягивать таблицу	При включенной настройке чертеж таблицы будет растягиваться на следующие страницы. Данный параметр



необходим если заголовки таблицы есть только на первой странице.



Для осуществления привязки границ таблицы для каждого угла чертежа необходимо указать якорь и задать параметры расположения линий относительно выбранного якоря.



Параметры якорных привязок углов таблицы.

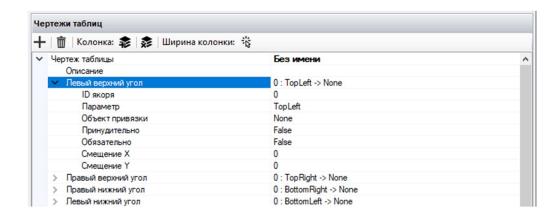
№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	ID якоря	ID якоря, к которому происходит привязка.
2.	Параметр	 Параметр привязки. Возможные значения: ТорLeft - привязка левого верхнего угла якоря к выбранному объекту привязки; ТорRight - привязка правого верхнего угла якоря к выбранному объекту привязки; ВоttomLeft - привязка левого нижнего угла якоря к выбранному объекту привязки; ВоttomRight - привязка правого нижнего угла якоря к выбранному объекту привязки; Left - привязка левой грани якоря (посередине) к выбранному объекту привязки Тор - привязка верхней грани якоря (посередине) к выбранному объекту привязки; Right - привязка правой грани якоря (посередине) к выбранному объекту привязки; Воttom - привязка верхней грани якоря (посередине) к выбранному объекту привязки; Воttom - привязка верхней грани якоря (посередине) к выбранному объекту привязки;



		 LeftArea - привязка всей левой области якоря к выбранному объекту привязки; RightArea - привязка всей правой области якоря к выбранному объекту привязки; ВottomArea - привязка всей нижней области якоря к выбранному объекту привязки; ТорArea — привязка всей верхней области якоря к выбранному объекту привязки; FullArea - привязка якоря со всех сторон к выбранному объекту привязки.
3.	Объект привязки	Точка или грань области атрибута для привязки выбранного параметра.
4.	Принудительно	При включенной настройке указанный параметр якоря будет задан к выбранному объекту привязки принудительно и без проверок.
5.	Обязательно	При включенной настройке ошибка данной привязки поиск всего элемента будет считаться неудачным.
6.	Смещение X	Смещение по оси X в пикселях. Для точного смещения следует указывать значения в формате + Например, +10-20. Данная настройка используется в случае, когда после якоря указывается какой-либо



		символ (скобка или двоеточие), который не должен попасть в область атрибута.
7.	Смещение Ү	Смещение по оси Y в пикселях. Для точного смещения следует указывать значения в формате + Например, +10-20.



Для отображения рамок чертежа таблицы необходимо задать параметры отображения каждой из рамок.

Параметры отображения рамок чертежа таблицы.

№ п/п	Элемент интерфейса	Описание элемента интерфейса
1.	Левая рамка	Отображение левой рамки чертежа таблицы.
2.	Верхняя рамка	Отображение верхней рамки чертежа таблицы.
3.	Правая рамка	Отображение правой рамки чертежа таблицы.
4.	Нижняя рамка	Отображение нижней рамки чертежа таблицы.



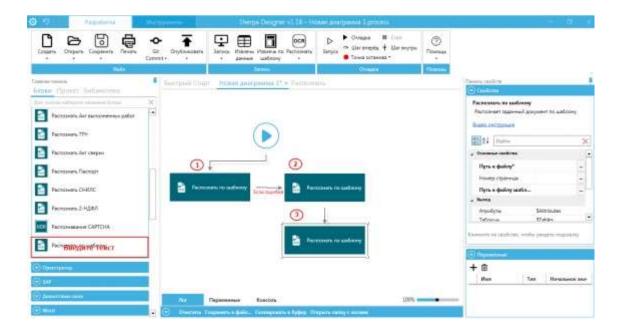
 Отображение рамок 	True;True;True;True
Левая рамка	True
Верхняя рамка	True
Правая рамка	True
Нижняя рамка	True

5.2 Действия в Sherpa Designer после создания шаблона

5.2.1 Значения переменных

После создания шаблонов для каждого типа документа (их может быть несколько 5, 10 и т.д.), можно пойти двумя путями:

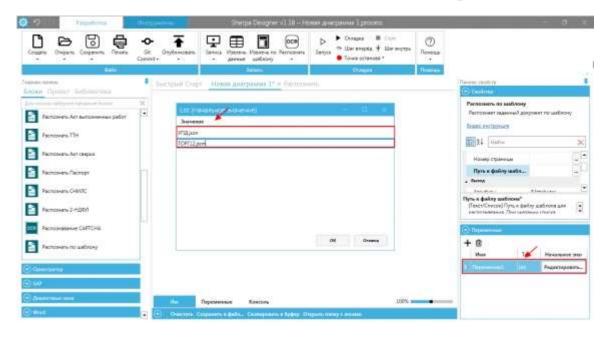
1) Настроить работу сценария проекта с поочередным распознаванием по каждому шаблону, при этом добавлять в сценарий столько блоков, сколько было создано шаблонов



2) Запустить распознавание по списку шаблонов в одном блоке.

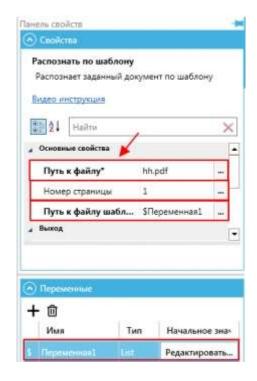
Прежде чем использовать список, необходимо его задать в Переменных. Выбираем Тип переменной List (список), задаем имя переменной и редактируем значение переменной: добавляем в список названия созданных нами шаблонов, по которым будет производиться распознавание. Например, ТОРГ12.json, УПД.json и т.д.





На панели свойств указываем путь к файлу PDF. Файл может быть многостраничным. Также поддерживаются файлы формата .png и .jpg.

Далее указываем путь к файлу шаблона для распознавания. При указании списка файлов шаблонов будут последовательно использоваться каждый файл, пока распознавание не закончится успешно. Если ни один из шаблонов не подойдет для документа, то будет сгенерировано исключение с ошибкой первого шаблона из списка.





Указываем путь к файлу в формате .pdf

В строке Путь к файлу шаблона указанную созданную нами переменную со списком шаблонов.

Номер страницы – 1, тогда распознавание начнется с первой страницы документа.

При запуске роботы блока «Распознать по шаблону», первым распознается документ, указанный в списке Переменной. В случае возникновения ошибки по одному из шаблонов, робот переходит к распознаванию следующего шаблона.

При использовании списка возможно увеличение скорости обработки нескольких шаблонов, если параметры распознавания страницы у этих шаблонов одинаковые.

Если параметры распознавания различны, то время распознавания документа увеличивается.

Например, если для одного из шаблонов указан ОСК-модуль Tesseract, а для другого шаблона – Yandex Vision (или разный масштаб или язык), то для каждого из этих шаблонов робот будет заново распознавать документ, т.к. указаны разные параметры.

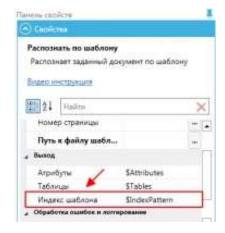
Поэтому рекомендуем при создании шаблонов для типовых стандартизированных документов оставлять настройки, заданные по умолчанию и менять их только в случае необходимости.

В случае, если параметры настроек некоторых шаблонов были изменены, то рекомендуем компоновать их по группам: сначала указать несколько шаблонов с одним видом настроек, затем несколько шаблонов с другим типом настроек.

Индекс шаблона

После распознавания документа будет определен индекс шаблона.

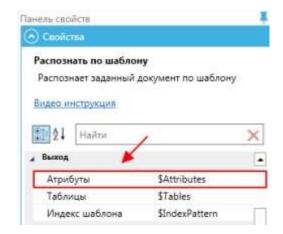




Используя блок «Условие» мы задаем траекторию дальнейшего сценария согласно полученным индексам. (например, если индекс 0 – то это УПД, если индекс 1 – Счет-фактура и т.д.)

Если для распознавания использовался один документ, а не список, то индекс по умолчанию всегда будет 0 (нулевой).

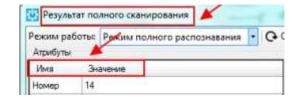
Атрибуты



После обработки шаблона необходимо продолжить работу с созданными атрибутами.

Атрибуты отображаются в окне «Результаты полного сканирования» Шаблонизатора (*Имя*, *Значение*).





Чтобы получить значения атрибутов мы должны обратиться к словарю и присвоить значение переменной. Используем блок «Присвоить значение переменной».

На панели свойств выбираем значение переменной, кликаем на «...» и редактируем выражение. В открывшемся окне выбираем категорию \$Attributes.

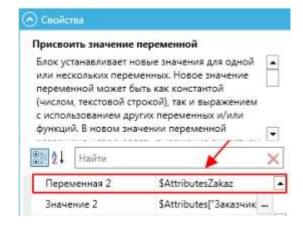


Указываем Имя атрибута, которое ранее было задано в Шаблонизаторе, используя кавычки.



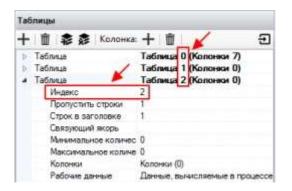
Далее необходимо задать имя переменной, с которой мы продолжим работать. Имя переменной может быть задано произвольно, например, \$AttributesZakaz или \$MyOrganization.





Таблица

Присваивая значение переменной для таблицы (если таблиц несколько, — то массив таблиц), необходимо использовать индекс таблицы, который был присвоен в Шаблонизаторе автоматически (по порядку). Вручную присвоить значение переменной невозможно.



На панели свойств выбираем значение переменной, кликаем на «...» и редактируем выражение. В открывшемся окне выбираем категорию \$Tables. Добавляем столько таблиц, сколько было добавлено в Шаблонизаторе и указываем индекс каждой таблицы.





Далее необходимо задать имя переменной, с которой мы продолжим работать. Имя переменной может быть задано произвольно, например, \$TablesProducts или \$MyTables.



После того, как Имя и Значение переменных были заданы, можно приступать к дальнейшей работе.

5.2.2 Работа с многостраничными документами

При создании шаблонов для многостраничных документов необходимо определить уникальные якоря, поиск по которым будет осуществляться только в определенном типе документа. Например, в документе УПД может встречаться слово счет-фактура, соответственно, не нужно использовать это слово в качестве якоря ни для УПД, ни для Счет-фактуры.

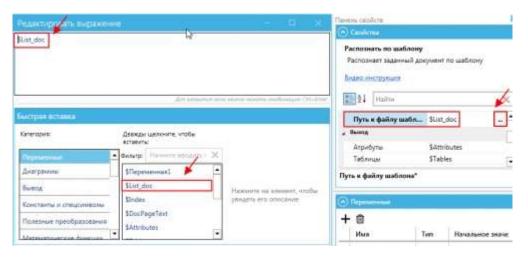
При необходимости работы с многостраничным документом, бывает, что один документ находится на нескольких страницах.

Для каждого типа документов необходимо создавать отдельный шаблон (например, 1 — Торг12, 2 – Счет-фактура, 3 – УПД), и указать все типы документов в значениях переменной. При этом тип переменной выбрать List (список).

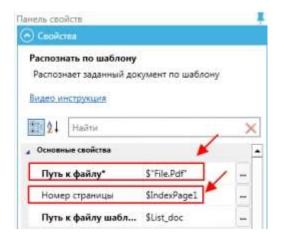


Далее указываем путь к файлу шаблона:





Указываем Номер страницы – 1, для того чтобы распознавание начиналось с первой страницы. Также указываем путь к файлу.



Завершение работы робота при возникновении ошибки

По окончании работы по создания шаблона, необходимо прописать дальнейший сценарий его распознавания и вывода результатов.

Если робот не смог произвести распознавание документа, то сценарий может быть остановлен. Для этого в Sherpa Designer добавляем блок «Конец». Также можно добавить блок «Лог» для записи ошибки в лог.



Ошибка распознавания может возникать в случаях, когда ни один из шаблонов не подошел или, когда не осталось документов для распознавания. 447



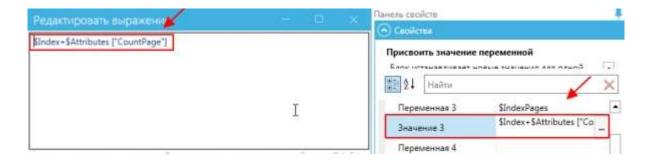
В случае, если в PDF файле находятся несколько документов, и один из типов документов неизвестен роботу (т.е. на этот тип документа нет шаблона), робот будет пропускать этот документ и будет распознавать лишь те документы, для которых есть шаблоны.

Поиск второго и последующих документов в многостраничном файле

После того, как в многостраничном документе был распознан один из шаблонов, необходимо внести изменения в прописанный индекс (прибавить), чтобы дальнейшее распознавание началось не со следующей страницы, а после найденного документа.

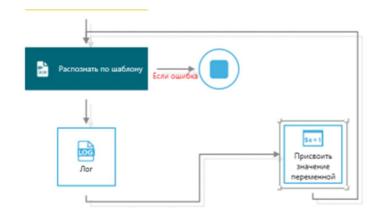
Так как в атрибутах существует параметр CountPage, который соответствует количеству страниц, участвующих в распознавании шаблоном, то для продолжения процесса распознавания нужно добавить этот параметр к индексу.

Таким образом, если в многостраничном документе был распознан первый документ, занимающий несколько страниц, при добавлении параметра CountPage дальнейшее распознавание начнется со страницы, следующей после распознанного документа.



После этого настраиваем сценарий на повторное распознавание документа.

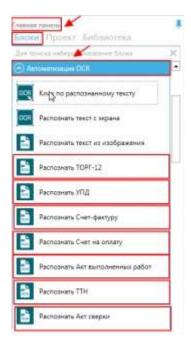




5.3 Примеры

5.3.1 Создание шаблона на примере документа «Справка о стоимости выполненных работ»

В случае, когда одним из действий сценария является распознавание и импорт данных из сканированного документа, необходимо воспользоваться блоками из вкладки Автоматизация ОСR, которые находятся на главной панели Дизайнера (Sherpa Designer).



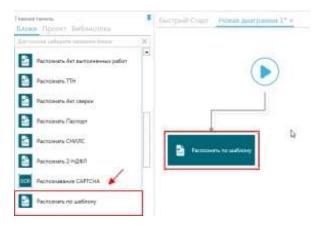
В Дизайнере есть предустановленные шаблоны по следующим видам документов:

- Торг12;
- УПД;
- Счет-фактура;



- Счет на оплату;
- Акт выполненных работ;
- TTH;
- Акт сверки;
- Паспорт;
- СНИЛС;
- НДФЛ;
- CAPCHA.

В случае, когда роботу необходимо распознать информацию из иного вида документа, не указанного во вкладке «Автоматизация ОСК», потребуется создать шаблон документа. При этом в сценарий проекта добавляется блок «Распознать по шаблону».



По созданному шаблону в дальнейшем робот будет распознавать и обрабатывать информацию из всех документов данного вида (Справка о стоимости выполненных работ и затрат, Акт об оказании услуг и т.д.).

Для каждого вида документов создается отдельный шаблон.

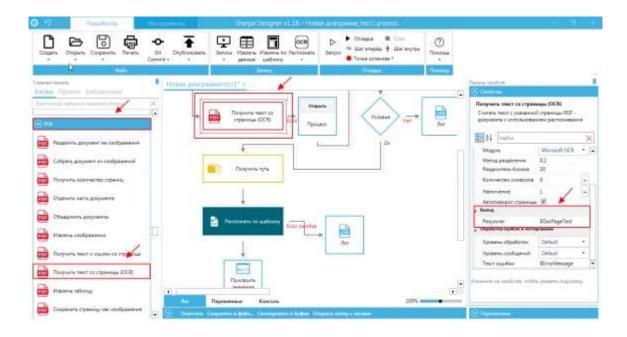
5.3.1.1 Предобработка документа

Рассмотрим создание шаблона на примере документа «Справка о стоимости выполненных работ и затрат».

Прежде чем робот приступит к созданию шаблона, документ должен быть распознан роботом и сохранен по указанному пути.

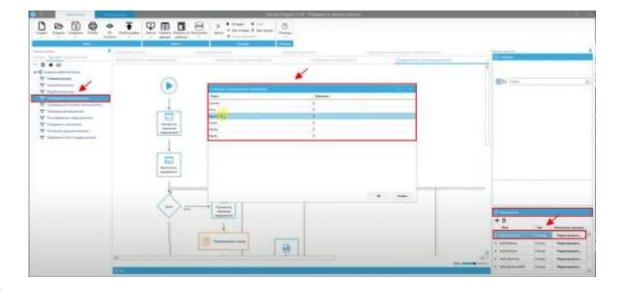


Например, при распознавании документа в формате .pdf в проекте сценария добавляем блок «Получить текст со страницы ОСК». В настройках результатов во вкладке «Выход» указываем \$DocPageText



Важно понимать, что в некоторых распознаваемых файлах название столбцов таблиц может не совпадать с названием колонок, которые заданы для вывода результата роботом. Также некоторые столбцы могут отсутствовать вовсе или могут отсутствовать некоторые границы таблиц.

Проверить и сопоставить названия столбцов необходимо в сценарии проекта «Определить колонки.process», они задаются в настройках справа: «Панель свойств» — «Переменные».





Например, после обработки документа «Справки о стоимости работ и затрат» в результате роботом должны выводиться следующие колонки:

Summa	сумма
Price	цена
SummaNDS	сумма с НДС
Stavka	ставка
Name	имя/название
Count	количество

Но при создании шаблона мы видим, что в самом документе часть этих данных отсутствует.

5.3.1.2 Создание шаблона

Создание шаблона происходит на верхней панели основного меню Дизайнера во вкладке «Разработка» — раздел «Запись» — «Извлечь по шаблону».



Извлечь по шаблону

При нажатии на кнопку «Извлечь по шаблону» открывается обработка «Генератор шаблонов» (Шаблонизатор).

При создании нового шаблона происходит автоматическое распознавание документа. Так как мы создаем шаблон документа, с которым ранее не работали, нам необходимо создать новые якоря и привязки к тем значениям, которые будут обрабатываться роботом именно в данном типе документа. Поэтому на сообщение во всплывающем окне «Вы хотите импортировать все таблицы из документа в шаблон» отвечаем «Нет».





Распознавание текста

Распознавание текста в Шаблонизаторе производится с помощью встроенных и внешних модулей распознавания:

- Tesseract OCR
- ABBYY Cloud OCR
- ABBYY Fine Reader
- Microsoft OCR
- YandexVision

В платформе Sherpa интегрированы несколько ОСR -модулей. Два из них - офлайн (поставляются вместе с роботом), и будут работать без подключения к интернету. Это Tesseract OCR и Microsoft OCR.

Yandex Vision и ABBYY OCR – онлайн-модули, т.е. используют функции соответствующих облачных модулей.

ABBYY Fine Reader – это коммерческий офлайн-модуль, использование которого требует отдельной лицензии.

Платформа позволяет настроить работу сценария с распознаванием изображений и переключаться между этими движками в любой момент времени. Для этого необходимо выбрать желаемый движок из списка.



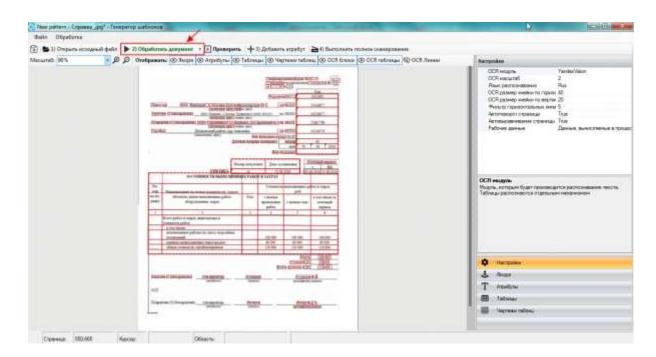




В случае, если распознаваемый документ высокого качества, с высоким расширением, то достаточно использовать бесплатные OCR-модули.

При обработке документов низкого качества лучше использовать платные решения.

После того, как основные параметры настроек заданы, документ необходимо обработать. Действие запускается нажатием кнопки «Обработать документ».

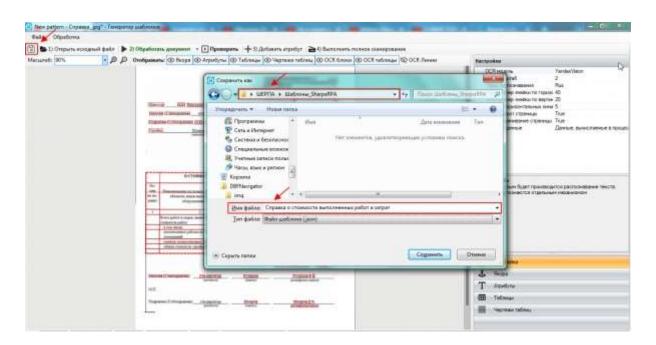


Обработанный документ выглядит так, как показано на скриншоте выше.

Сохранить шаблон



Для дальнейшей работы с шаблоном его нужно сохранить, нажав на иконку в левом верхнем углу основной панели действий. Далее указываем путь для сохранения шаблона и имя файла. Имя файла указываем исходя из вида документа, например, «Справка о стоимости выполненных работ и затрат».



5.3.1.3 Работа с таблицами

После сохранения шаблона переходим к работе с таблицами.

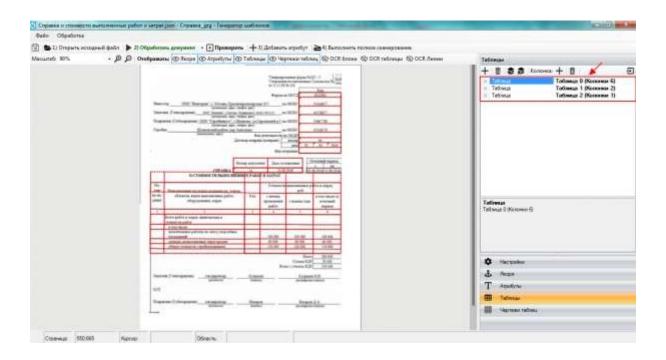




В левой части основной панели настроек выбираем вкладку «Таблицы» — нажимаем кнопку «Импортировать» [1].

Импортируем все таблицы документа, для этого в появившемся окне «Вы хотите импортировать все таблицы документа в шаблон?» выбираем «Да».

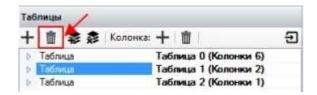
На рабочей панели справа отобразятся все таблицы шаблона, которые были распознаны. Их может быть одна-две и более. В нашем случае на скриншоте ниже мы видим, что ОСR-модулем распознаны 3 таблицы шаблона.



Каждой таблице автоматически присваивается номер, нумерация начинается с 0. Также отображается количество колонок каждой распознанной таблицы. Именно по этому значению мы можем определить с какой (или с какими) таблицами мы продолжим работать.

В нашем примере нас интересуют данные из таблицы с 6-ю колонками, соответственно, остальные таблицы (с 2 и с 1 колонками) мы можем удалить из окна панели.

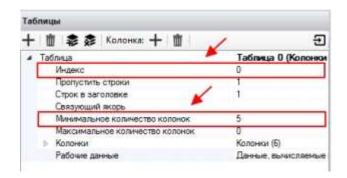




Удалить таблицу можно нажав на значок на верхней панели окна, предварительно выделив таблицу, которую нужно удалить.

Индекс

Индекс ОСR-таблицы. В случае, если Шаблонизатор извлекает данные лишь из одной таблицы сканированного документа, необходимо указывать индекс, равный 0. При этом для улучшения привязки рекомендуется указывать минимальное количество колонок.



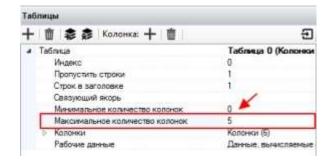
Минимальное количество колонок

Указывается минимальное количество колонок в ОСR-таблице. Все таблицы с количеством меньше указанного будут пропущены.

Максимальное количество колонок

Указывается максимальное количество колонок в ОСR-таблице. Все таблицы с количеством больше указанного будут пропущены.





При создании шаблона «Справки о стоимости выполненных работ и затрат» указываем минимальное количество колонок равное 5, т.к. нас интересуют данные из таблицы с 6 колонками. Остальные таблицы данного документа содержат менее 5 колонок.

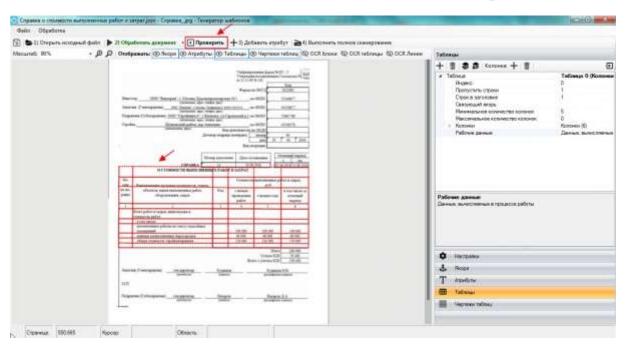
Таким образом Шаблонизатор продолжит работу с интересующей нас таблицей.

При создании шаблонов для других типов документов (с большим количеством таблиц и колонок в таблицах) можно указывать и максимальное и минимальное количество колонок, тем самым задавая интервал, в который будут попадать таблицы с интересующими нас данными.

Проверить

Для проверки корректности отображения табличной части, из которой будут импортироваться данные нажимаем кнопку «Проверить» на верхней рабочей панели.





Также проверку можно запустить нажатием клавиши F5.

Пропустить строки

Указываем количество строк, которые нужно пропустить для того, чтобы данные этих строк не попали в результаты работы робота.

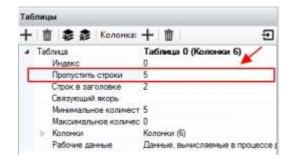
При выборе значения, равного 1, — первые строки будут пропущены. Наименования колонок будут соответствовать значениям первой строки.

Если задать названия колонок в шаблоне, то эти значения не будут изменены.

В нашем примере необходимо пропустить 5 верхних строк таблицы, т.к. значения интересующих нас данных указаны в таблице документа начиная с 6 строки.

Для этого в строке «Пропустить строки» указываем число 5.

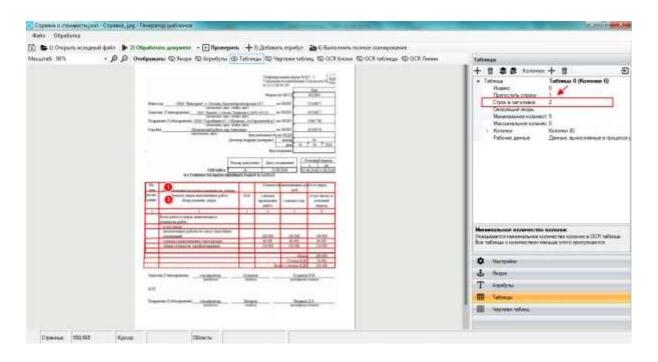




Строк в заголовке

При построении таблицы, заданное количество строк будет использовано для создания заголовков колонок. При указании значения 0 заголовки будут взяты из шаблона колонок, в противном случае название будет суммироваться из ячеек колонки по заданному количеству строк.

Например, в шаблоне «Справки о стоимости выполненных работ и затрат» необходимо указать количество строк заголовке равное 2, т.к. в заголовок должна попасть информация, размещенная в двух верхних строках таблицы.

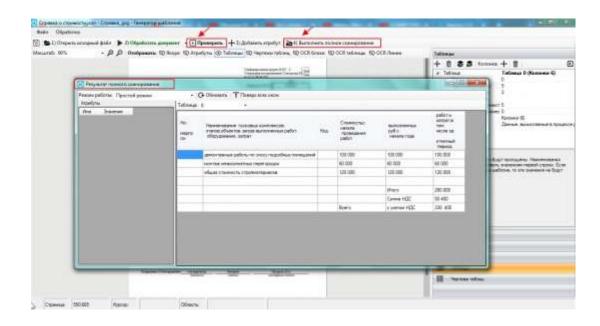


После выполненных действий необходимо проверить промежуточный результат. Нажимаем «Проверить» — далее «Выполнить полное сканирование».

Результат полного сканирования



Результат полного сканирования отобразится в виде таблицы с импортированными данными из табличной части сканированного документа.

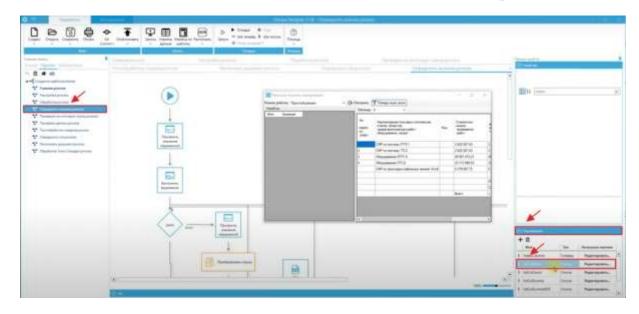


Определить колонки

На этом этапе часть данных может отражаться некорректно, в частности, названия колонок могут не совпадать с теми значениями, которые ранее были заданы в сценарии проекта «Определить колонки.process» — в настройках «Панель свойств» — «Переменные».

Для корректировки выводимых результатов необходимо вернуться в сценарий проекта «Определить колонки.process», сопоставить значения переменных, заданных в настройках «Панель свойств», добавить указанные здесь колонки и определить для них названия.

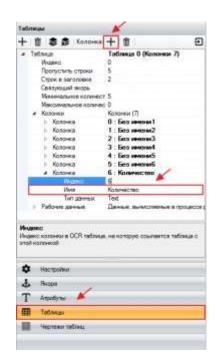




Добавление колонок «Количество», «НДС Сумма», «Ставка».

В нашем примере в результатах полного сканирования отсутствует несколько колонок, в том числе колонка «Количество».

Название колонки должно соответствовать значению, присвоенному данной переменной «количество».







Добавление недостающих колонок производится на правой рабочей панели Шаблонизатора во вкладке «Таблицы». Нажимаем на кнопку +, добавляем колонку, присваиваем колонке имя «Количество».

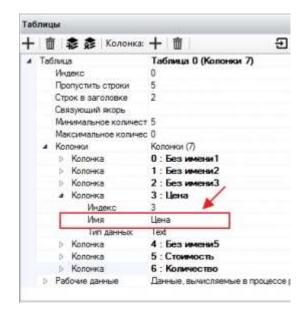
При повторной проверке в результатах полного сканирования Шаблонизатора появится колонка «Количество».

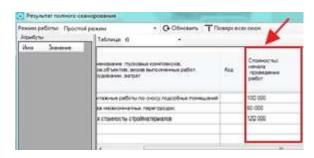
Производим проверку остальных колонок и корректируем их названия в зависимости от значений переменных в сценарии проекта «Определить колонки.process».

При создании шаблона «Справка о стоимости выполненных работ и затрат» аналогичным образом добавляем колонки: «НДС Сумма», «Ставка».

В случае, если название колонки не соответствует названию переменной, но содержание колонки соответствует значению переменной, то создавать дополнительную колонку не нужно, достаточно изменить название колонки в Шаблонизаторе.





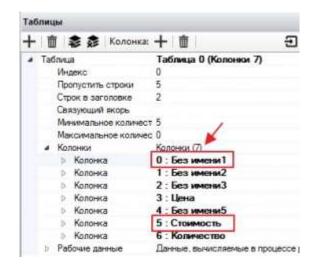


Например, в таблице с результатами Шаблонизатора название колонки «Стоимость начала проведения работ», а название переменной «Цена», тогда мы вносим изменения в название колонки, предварительно убедившись в том, что именно в этой колонке отображается цена.

Аналогичным образом необходимо изменить названия всех колонок, которые не совпадают с названиями переменных.

Нумерация колонок таблицы в Шаблонизаторе начинается с 0.





В случае если название колонки сохранено по умолчанию и содержит в себе числовое значение (например, «Без имени1»), то в таблице результатов Шаблонизатора название указанной колонки определится автоматически и будет содержать в себе текст из соответствующей колонки сканированного документа.

Если название колонки мы изменили вручную, то в таблице результатов Шаблонизатора будет отображаться именно заданное нами название колонки.

Поэтому, рекомендуем обратить особое внимание на часть табличного документа, в котором название колонки указано в объединенных ячейках. Даже если при создании шаблона документа название колонки из объединенных ячеек было распознано корректно, в других документах название может быть указано со смещением.

В нашем примере слово «Стоимость» располагается в 4 ячейке таблицы, но в некоторых документах может сместиться в 5 ячейку. В таких случаях рекомендуем переименовать все ячейки, названия которых могут быть смещены.





По завершению работы с таблицей в Шаблонизаторе необходимо произвести проверку и удостовериться в корректности результатов полного сканирования.

5.3.1.4 Якоря

Точки на сканированном документе, с помощью которых робот определяет, в какой части документа размещаются определенные данные.

При помощи якорей мы задаем границы определенных атрибутов. Якорей может быть сколь угодно много, в зависимости от типа документа.



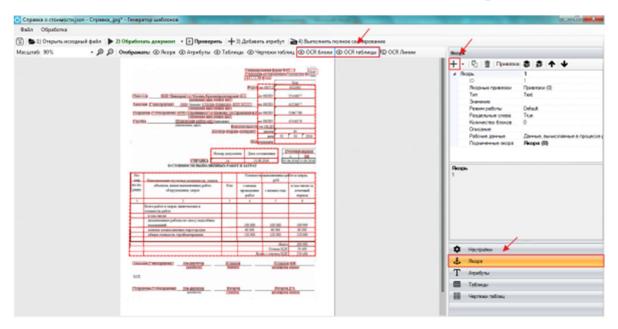
В нашем примере необходимо получить данные следующих атрибутов:

- «Инвестор»
- «Заказчик (Генподрядчик)»
- «Номер документа»
- «Дата»

Создание якоря (атрибут «Инвестор»)

Рассмотрим процесс создания якоря на примере атрибута «Инвестор».





Для создания якоря переходим во вкладку «Якоря» в правой рабочей области Шаблонизатора и нажимаем кнопку в верхней части панели. В этот момент становятся активны «ОСК блоки» и «ОСК таблицы» для удобства выбора определенного блока в качестве якоря.

Для того, чтобы Шаблонизатор мог корректно определить местоположение атрибута в документе, необходимо ограничить его границы. Ограничение верхней границы задается по тексту, находящемуся на строку выше самого атрибута. В нашем случае это блок «Форма по».

Для присвоения значения якорю достаточно кликнуть левой клавишей мыши по ОСК-блоку (слову), которое мы выбрали в качестве якоря. Также можно указать значение вручную, написав якорное слово в строке «Значение».

В случае, если мы укажем в качестве якоря слово «Форма», то якорь будет установлен неверно, т.к. в документе есть несколько ОСR-блоков, распознавших это слово.

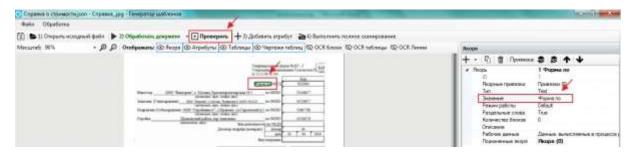


Поэтому меняем указанное значение якоря на «Форма по». После этого необходимо произвести проверку, чтобы убедиться, что якорное слово определилось корректно.



Когда значение якоря задается нажатием клавишей мыши на ОСR-блок, — к значению добавляется знак «*», если значение указывается вручную, — добавлять этот знак необязательно. При использовании якорных привязок знак «*» нужно удалить и после этого добавить привязку.

В случае, если значение содержит два и более слова, — между ними необходимо ставить пробел, т.к. он выступает разделителем слов.



Далее добавляем якорь для атрибута «Инвестор». Для корректного распознавания информации об Инвесторе, необходимо ограничить якорными точками данный атрибут слева и справа.

Слева якорем будет выступать слово «Инвестор». Добавляем якорь, нажимаем левой клавишей мыши на ОСК-блок «Инвестор» и Шаблонизатор автоматически заполняет строку «Значение», тем самым определяя якорную точку.



Справа якорем будет выступать слово «по». По аналогии добавляем якорь в рабочей области Шаблонизатора, нажимаем на ОСК-блок «по» и производим проверку.

После проверки видим, что якорь установлен неверно, т.к. в документе есть несколько ОСR-блоков, распознавших слово «по».

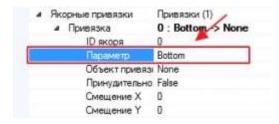
Для корректного определения якоря осуществляем якорную привязку к якорю «Форма по», который находится строкой выше того блока, который мы хотим указать в качестве якоря. Для этого нажимаем на кнопку «Добавить привязку», затем кликаем по якорю «Форма по», тем самым привязываем наш



новый якорь к ОСR-блоку, который располагается строкой выше нужного нам якоря.



Для того, чтобы Шаблонизатор определил координаты создаваемого якоря относительно якоря «Форма по» указываем параметр привязки «Bottom». Этот параметр говорит о том, что создаваемый якорь будет привязан к якорю «Форма по» и находится строкой ниже.

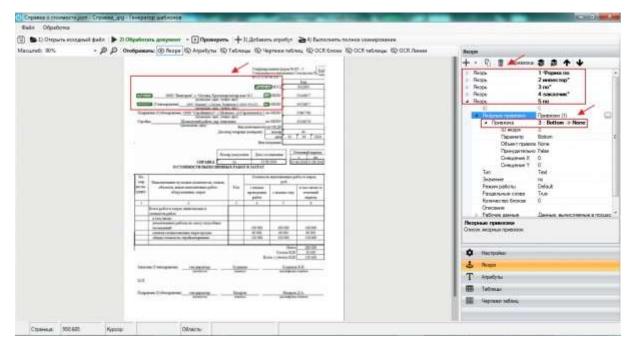


Создание якоря (атрибут «Заказчик»)

По аналогии создаем якоря для атрибута «Заказчик». Слева якорем будет выступать слово «Заказчик». Добавляем якорь, нажимаем левой клавишей мыши на ОСR-блок «Заказчик» и Шаблонизатор автоматически заполняет строку «Значение», определяя якорную точку.

Справа якорем будет выступать слово «по». По аналогии добавляем якорь в рабочей области Шаблонизатора, нажимаем на ОСК-блок «по», производим проверку и добавляем якорную привязку. В этом случае привязку осуществляем к якорному слову «по», которое располагается строкой выше в документе. Указываем параметр привязки «Bottom» и производим проверку.



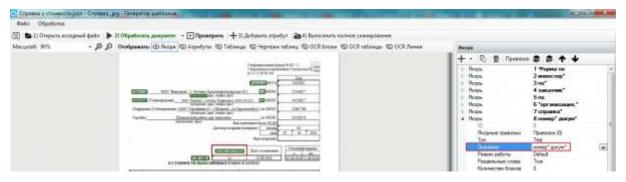


Атрибут «Заказчик» необходимо ограничить не только справа и слева, но и сверху. Для этого в качестве якоря выбираем слово «организация». Создаем еще один якорь, редактируем название так, чтобы с обеих сторон слова стоял знак «*» — «*организация*». Проверяем корректность расположения якорей.



При создании якорей, в значении которых содержится несколько слов, необходимо указывать знак «*» в конце каждого слова.

Если сочетание слов встречается в документе единожды, то второе слово можно написать сокращенно. Например, в качестве якоря по номеру документа можем указать «номер* докум*».



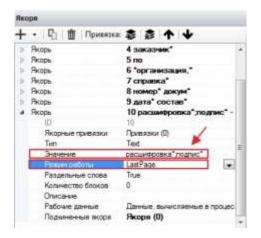


Если в начале значения якоря присутствует лишний символ (чаще всего это скобка), мы можем удалить этот символ и вместо него поставить знак «*». Тогда получим значение *организация* вместо (организация*.

Завершающий якорь

При создании шаблона необходимо указать якорь, который будет завершать документ. Т.е. по этому якорю мы можем определить, что в документе таблиц больше нет и работу с таблицами можно завершать.

В нашем примере завершающим якорем будет слово расшифровка.



В нашем примере завершающим якорем будет слово расшифровка*.

Шаблонизатор позволяет задать для якоря несколько альтернативных вариантов, по которым робот будет распознавать завершающий атрибут.

Добавим слово **подпис***, укажем намеренно без окончания, т.к. в документе может быть указано как подпись, там и подписи.

Если в документе отсутствует слово $pасшифровка^*$, то робот будет искать слово $nodnuc^*$

Для завершающего якоря указываем *Режим работы – LastPage*.

5.3.1.5 Создание атрибутов

Следующий этап в создании шаблона - создание атрибутов.

Атрибуты — позволяют определить и обозначить области документа, данные из которых будут распознаваться роботом и импортироваться в таблицу «Результаты».

Атрибуты «Инвестор» и «Заказчик»





В нашем примере в таблицу с результатами должны импортироваться данные об Инвесторе, Заказчике, а также Дата и Номер документа

В Шаблонизаторе необходимо добавить эти атрибуты на рабочей панели справа: выбираем «Атрибуты», нажимаем кнопку добавить, присваиваем имя атрибута «Инвестор».

Для корректного распознавания области документа, данные из которой будут импортироваться в результаты, для атрибута необходимо осуществить привязку к ранее созданным якорям.



В нашем примере якорем справа будет выступать якорь «Инвестор», слева - якорь «по».

Якорные привязки к атрибуту «Инвестор»

Для создания привязок в созданном атрибуте нажимаем «Якорные привязки», далее «Добавить привязку», затем левой клавишей мыши кликаем на 472



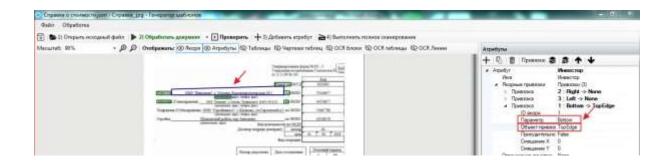
якорное слово «Инвестор». Аналогичным образом создаем привязку к якорному слову «по». Производим проверку, в результате которой область данных нашего атрибута должны подсветиться синей рамкой, как на скриншоте выше.

Чтобы ограничить область данных атрибута сверху, создаем третью якорную привязку к ОСR-блоку «Форма по». Нажимаем «Добавить привязку», кликаем левой клавишей мыши на созданный ранее якорь «Форма по». В параметрах привязки выбираем значение «Bottom», т.к. в качестве ориентира мы будем использовать нижнюю границу данного якоря.



Помимо параметра в привязке для данного атрибута необходимо указать «Объект привязки». Т.к. данные атрибута могут располагаться выше строки, которую мы выделили при помощи первых двух якорей слева и справа.

В строке «Объект привязки» указываем «ТорЕdge» — верхняя грань.



Якорные привязки к атрибуту «Заказчик»

По аналогии создаем атрибут «Заказчик», устанавливаем привязки к ранее созданным якорям «Заказчик» справа, «по» слева и «организация» сверху. Для привязки к якорю сверху устанавливаем параметр «Bottom», «Объект привязки» указываем «ТорЕdge».



В результате по каждому атрибуту должно быть указано по три якорные привязки: справа, слева и сверху.

После проверки в шаблоне каждый атрибут будет отображаться в виде синей рамки. В случае, если границы рамки не захватывают всю область данных атрибута, необходимо внести изменения в настройки.



Ограничение линиями

В некоторых документах названия полей, которые мы используем в качестве якорей могут располагаться не на нижней линии с описанием поля, а сверху или посередине, как на рисунке ниже.



В таких случаях для корректной привязки рекомендуем воспользоваться дополнительным параметром «Ограничение линиями». Предварительно нужно проверить присутствуют ли в шаблоне ОСR-линии, ограничивающие созданные нами атрибуты.

Нажимаем кнопку OCR-линии на верхней рабочей панели и проверяем наличие линий.





Линии подсвечены зеленым цветом, и мы можем использовать их в качестве дополнительного ориентира в работе с атрибутами.

На верхней рабочей панели выбирает «Атрибуты», кликаем левой клавишей мыши на один из атрибутов, после чего он становится активным и в правой рабочей панели открываются инструменты для работы с выбранным атрибутом.

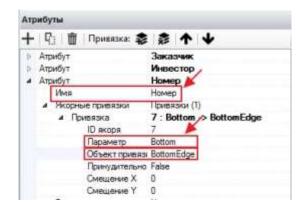


Выбираем «Ограничение линиями», из выпадающего списка выбираем значение «Down». Таким обратом мы ограничиваем нижнюю границу атрибута, которая теперь будет ориентироваться на ОСК-линии.

Обязательно проверяем корректность отображения границ атрибута. При необходимости меняем значения данного параметра.

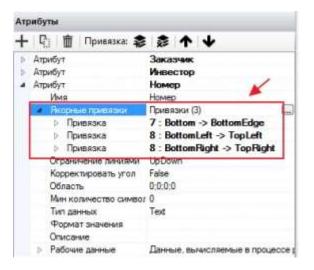
Создание атрибутов «Номер» и «Дата». Якорные привязки





Добавляем в правой рабочей панели атрибут, задаем имя атрибута *Номер*. Добавляем привязки к ранее созданным якорям.

В качестве привязки слева используем якорь *Справка**, в качестве параметра выбираем *Воttom*, также указываем *Объект привязки – BottomEdge*.



Создаем вторую привязку к якорю *номер* докум**. В качестве параметра выбираем *BottomLeft* (нижний левый угол). Объект привязки указываем *TopLeft* (верхний левый угол).

Таким же образом производим привязку атрибута к правому углу. Создаем третью привязку также к якорю *номер* докум**. В качестве параметра выбираем *BottomRight* (нижний правый угол). Объект привязки указываем *TopRight* (верхний правый угол).

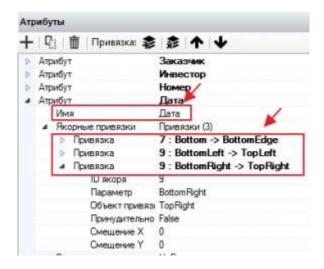
Таким образом, для атрибута Номер должны быть созданы три якорные привязки.



После проверки в шаблоне должен появиться синий прямоугольник, внутри которого отображены данные нашего атрибута.



Создание атрибута Дата.



Добавляем в правой рабочей панели атрибут, задаем имя атрибута *Дата*. Добавляем привязки к ранее созданным якорям.

В качестве привязки слева используем якорь *Справка**, в качестве параметра выбираем *Bottom*, также указываем *Объект привязки – BottomEdge*.

Создаем вторую привязку к якорю дата* *состав**. В качестве параметра выбираем *BottomLeft* (нижний левый угол). Объект привязки указываем *TopLeft* (верхний левый угол).

Таким же образом производим привязку атрибута к правому углу. Создаем третью привязку также к якорю дата* $cocma\theta$ *. В качестве параметра выбираем BottomRight (нижний правый угол). Объект привязки указываем TopRight (верхний правый угол).

В результате для атрибута *Номер* должны быть созданы три якорные привязки.

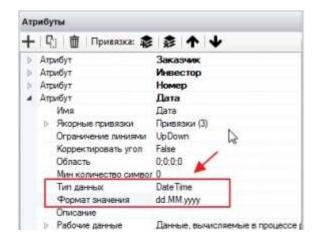


После проверки в шаблоне должен появиться синий прямоугольник, внутри которого отображены данные нашего атрибута.



Тип данных - DateTime.

Формат значения – вручную указываем формат, необходимый для атрибута. В нашем примере это *dd.MM.yyyy*



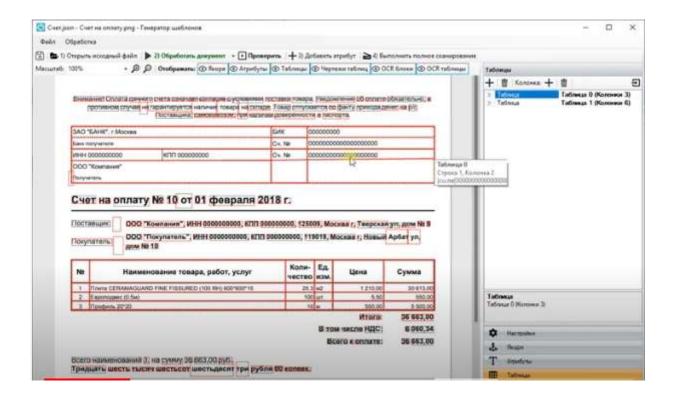
В случае, если в каком-либо документе дата документа будет указана как 30 ноября 2022г., в результате эта дата будет преобразована в 30.11.2022.

5.3.2 Создание шаблона с фиксированными атрибутами на примере документа «Счет на оплату»

Фиксированные атрибуты имеют положение и размеры, привязанные только к сторонам документа (масштабу), но не привязанные к конкретным якорям, поэтому их можно свободно перемещать по документу. Фиксированные атрибуты допустимо использовать только если во всех возможных образцах документа необходимые для извлечения данные находятся примерно на одном и том же месте и не смещаются по документу.



При создании шаблона «Счет на оплату», Шаблонизатор распознает все таблицы, которые содержатся в документе. Эта информация отображается на вкладке «Таблицы».

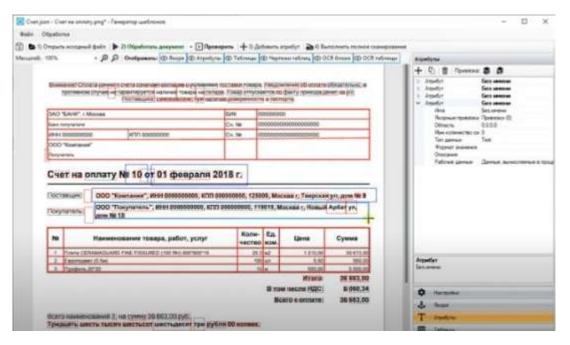


В случае, если роботом будут использоваться данные лишь из одной таблицы, вторая таблица может быть удалена. Для этого достаточно выделить ненужную таблицу и нажать на кнопку « в верхней части рабочей области «Генератора шаблонов».

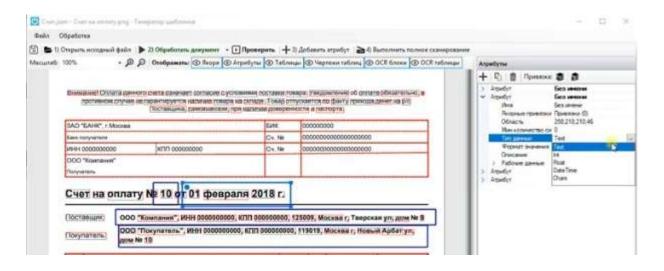


Вкладка «Атрибуты» позволяет определить и обозначить области документа, данные из которых будут распознаваться роботом и импортироваться в документы «Результаты».





Для каждого атрибута (выделенной области, данные из которой будут импортированы) необходимо определить Тип данных из выпадающего списка. Например, для атрибута «дата документа» выбираем «Date Time».



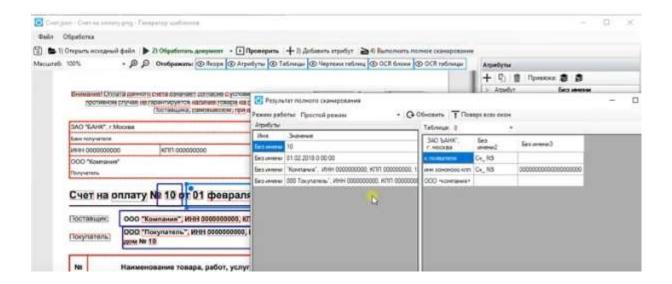
При необходимости любой атрибут может быть удален или добавлен новый.

После завершения работы во вкладке «Атрибуты» производится проверка шаблона и выполняется полное сканирование. Нажимаем «Проверить», затем «Выполнить полное сканирование».

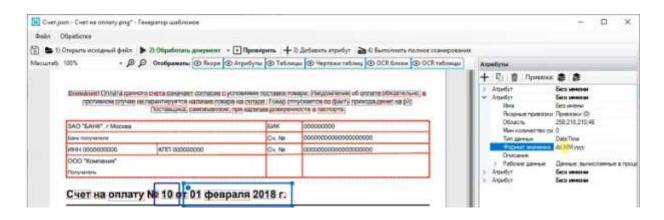
Открывается окно «Результат полного сканирования», в котором отображаются распознанные данные из документа в виде заполненной таблицы. 480



Если при проверке обнаружены неточности, то их в любой момент можно исправить.



При выборе типа данных «дата документа» дополнительно можно указать «Формат значения» вручную. После чего провести повторную проверку и убедиться в корректности значений.



При распознавании данных двух и более таблиц обрабатываемого документа, в окне «Результат полного сканирования» будут отображаться значения всех таблиц. Для просмотра значений по всем таблицам предусмотрена возможность переключения таблиц в выпадающем списке.

