ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

ООО «КОРТЕКС»

Руководство пользователя программного обеспечения платформы автоматизации рабочих процессов «Кортекс Потоки»

г. Краснодар, 2026

ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

1. Назначение системы

Платформа автоматизации рабочих процессов «Кортекс Потоки» (далее — Система) предназначена для визуального конструирования процессов обработки данных из функциональных узлов, получения данных из внешних источников (веб-сервисы, почта, базы данных, файловые хранилища), их преобразования, ветвления и циклической обработки, запуска процессов вручную, по расписанию и по внешним запросам, контроля хода выполнения в реальном времени, хранения файлов в общем хранилище с доступом по протоколу SFTP и разграничения доступа пользователей к процессам.

Система работает как веб-приложение и не требует установки программ на рабочее место пользователя. Для работы используется браузер: Chrome 110+, Яндекс Браузер 23+, Firefox 110+, Opera 95+, Safari 16+, Microsoft Edge на базе Chromium 110+. Internet Explorer и устаревшие браузеры не поддерживаются. Язык интерфейса — русский. Рекомендуемое разрешение экрана — 1600×900 и выше: визуальный редактор процессов использует всю ширину окна.

Пользователь выполняет вход по адресу электронной почты и паролю, после чего получает доступ к разделам в зависимости от роли. Основные пользователи Системы — специалисты, автоматизирующие обработку и передачу данных, и администраторы.

В интерфейсе Системы используются следующие наименования: процесс обработки данных называется **пайплайном**, функциональный узел процесса — **нодой**. В настоящем руководстве термины «процесс» и «пайплайн», «узел» и «нода» употребляются как равнозначные; названия кнопок, полей и сообщений приводятся точно в том виде, в котором они отображаются на экране.

1.1. Перечень возможностей

- вход по адресу электронной почты и паролю, самостоятельная регистрация, выход, раздел «О программе»;
- реестр процессов (раздел «Пайплайны»): создание, дублирование, удаление, статус и время последнего запуска;
- визуальный редактор процессов: холст с узлами и связями, палитра узлов, окно выбора узла, панель настроек узла, переменные процесса, проверка графа перед запуском, сохранение;
- библиотека узлов: триггеры (ручной запуск, расписание, вебхук, получение почты IMAP), действия (REST API, отправка почты SMTP), файлы и веб (загрузка веб-страницы, чтение файла, чтение и запись во внутреннее хранилище, чтение из S3-совместимого хранилища объектов, чтение по FTP), данные (SQL-запрос к PostgreSQL и MySQL, извлечение данных из HTML, разбор JSON, CSV, XML, HTML, Excel, шаблоны, изменение полей), логика (условие, множественный выбор, циклы «для каждого» и «пока», установка переменной, вложенный процесс), запрос к сервису большой языковой модели, вывод (журнал, Telegram);
- подстановка значений `{{узел.поле}}` и `{{vars.ИМЯ}}` с подсказками при вводе;

- выполнение с отображением хода на холсте в реальном времени, журнал запуска, выходные данные узлов, история запусков; повторы, таймауты, продолжение при ошибке, защита от бесконечной вложенности;
- запуск по вебхуку и по расписанию cron;
- совместный доступ (чтение, запись), автоматические версии процесса и их восстановление;
- файловое хранилище (раздел «Файлы») с доступом по протоколу SFTP;
- ИИ-ассистент: формирование и доработка процесса по текстовому описанию;
- администрирование: пользователи, настройки, журнал событий, диагностика, глобальные константы, настройки ИИ-ассистента.

2. Вход в систему

2.1. Вход

Для входа в Систему откройте в браузере адрес веб-приложения, предоставленный администратором. Если сеанс работы не открыт, отображается страница входа: в центре экрана — форма с названием «Кортекс Потоки» и подписью «Вход в систему», внизу экрана — строка с информацией о правообладателе и ссылкой «О программе».

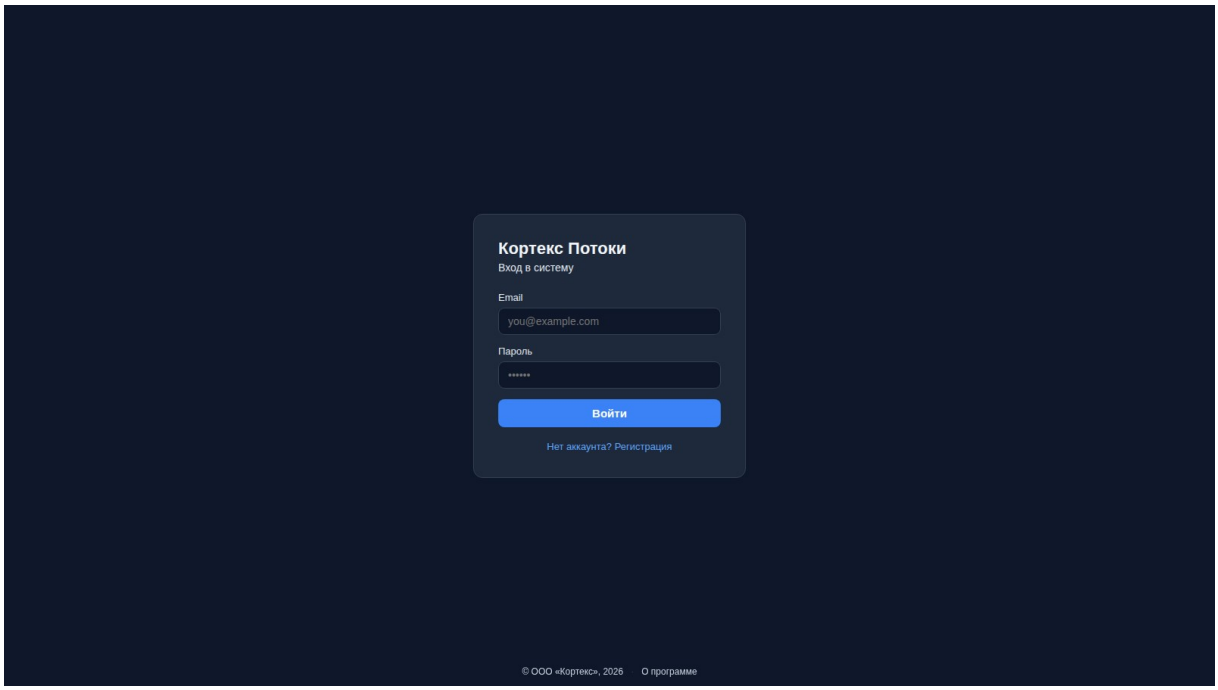


Рисунок 1 — Страница входа в систему

Поле / элемент	Назначение
Email	Адрес электронной почты, указанный при регистрации или при создании учётной записи администратором. Регистр букв не учитывается.
Пароль	Пароль учётной записи; вводимые символы

Поле / элемент	Назначение
	скрыты.
Войти	Выполняет вход. Вход также выполняется нажатием клавиши Enter. Во время проверки на кнопке отображается многоточие.
Нет аккаунта? Регистрация	Переключает форму в режим регистрации новой учётной записи (подраздел 2.2).

Заполните поля «Email» и «Пароль» и нажмите «Войти». При успешной проверке открывается раздел «Пайплайны» (раздел 5).

2.2. Регистрация

Пользователь может самостоятельно создать учётную запись. Для этого на странице входа нажмите «Нет аккаунта? Регистрация»: подпись формы сменится на «Регистрация», над полем «Email» появится поле «Имя» (необязательное). Укажите адрес электронной почты (он становится логином и должен быть уникальным), пароль длиной не менее 6 символов и нажмите «Зарегистрироваться». Учётная запись создаётся с ролью «Пользователь», вход выполняется сразу. Ссылка «Уже есть аккаунт? Войти» возвращает форму в режим входа. Роль «Администратор» назначается администратором в разделе «Админ» (подраздел 13.2).

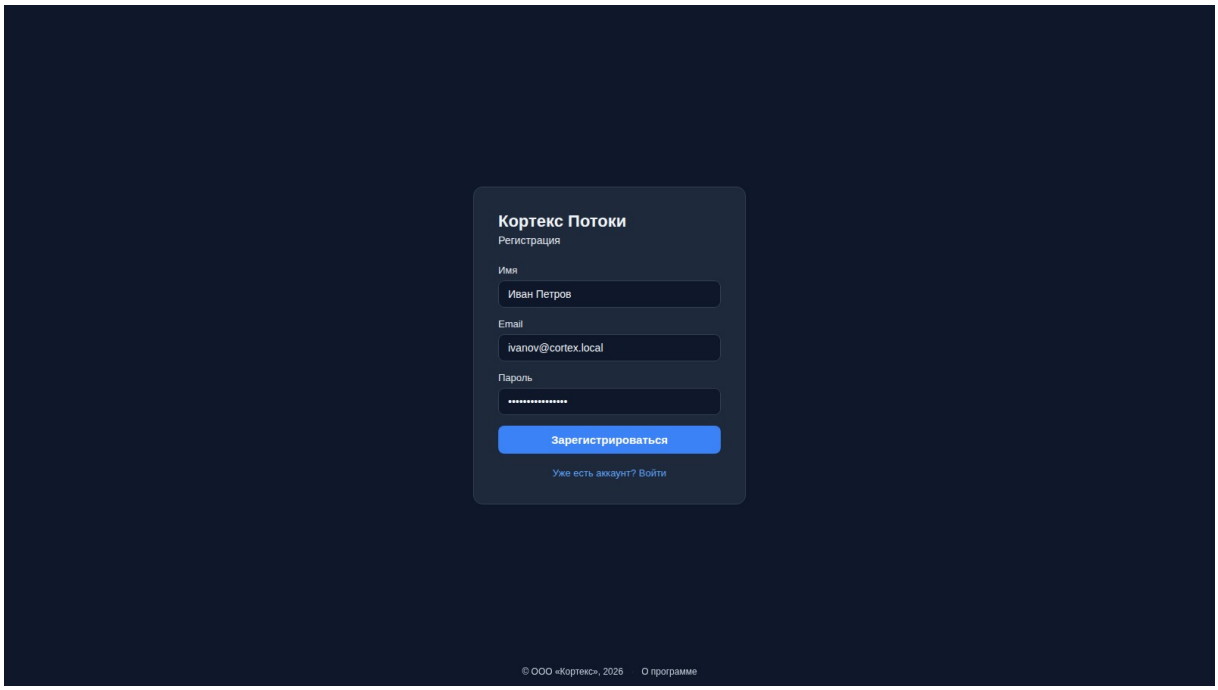


Рисунок 2 — Форма регистрации

2.3. Сообщения об ошибках входа

При ошибке входа или регистрации под полями формы выводится сообщение красного цвета.

Сообщение	Причина и действия пользователя
Неверный e-mail или пароль	Неверный адрес электронной почты или

Сообщение	Причина и действия пользователя
	пароль.
Учётная запись заблокирована	Учётная запись заблокирована администратором; обратитесь к администратору.
Этот e-mail уже зарегистрирован	Адрес уже зарегистрирован; выполните вход или укажите другой адрес.
Укажите e-mail и пароль (не менее 6 символов)	При регистрации не указан адрес или пароль короче 6 символов.
Не удалось войти	Сервер недоступен; повторите попытку позже.

Каждая попытка входа (успешная, неуспешная, попытка входа заблокированного пользователя) регистрируется в журнале событий (подраздел 13.4).

2.4. Выход и продолжительность сеанса

Для завершения работы нажмите кнопку «Выход» в верхней части раздела «Пайплайны». Сеанс будет закрыт, и Система вернётся на страницу входа.

Сеанс продлевается автоматически, пока пользователь работает в Системе. Если сеанс истёк или не может быть продлён (например, после блокировки учётной записи администратором), Система перенаправляет пользователя на страницу входа.

3. Общие принципы работы

3.1. Интерфейс

После входа открывается раздел «Пайплайны» — реестр процессов. В верхней части экрана расположены заголовок раздела и панель кнопок перехода к другим разделам; под ней — таблица процессов. Внизу экрана расположена строка с информацией о правообладателе и ссылкой «О программе».

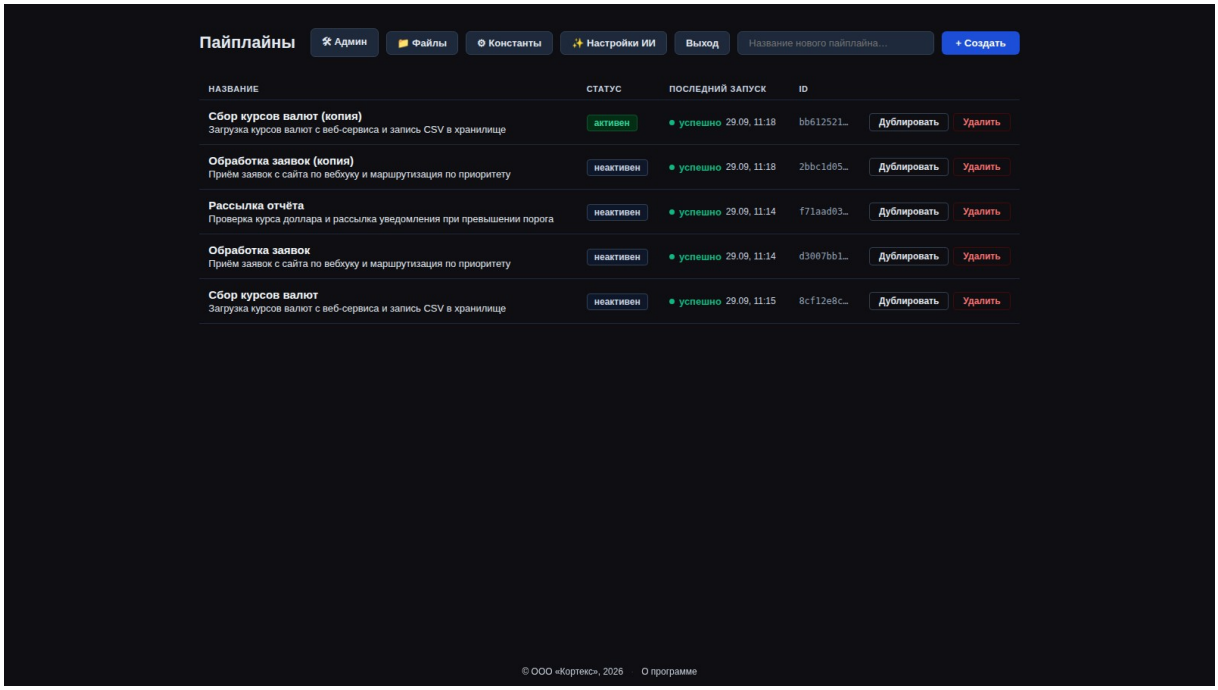


Рисунок 3 — Главный экран Системы — раздел «Пайплайны»

Кнопки в шапке раздела «Пайплайны» снабжены пиктограммами; в руководстве кнопки называются по тексту надписи.

Кнопка / элемент	**Назначение**
Админ	Переход в раздел администрирования «Управление системой» (раздел 13). Кнопка отображается только пользователям с ролью «Администратор».
Файлы	Переход в раздел «Файлы» — файловое хранилище (раздел 11).
Константы	Открывает окно «Глобальные константы» — значения, доступные во всех процессах (подраздел 13.6). Кнопка отображается только администратору.
Настройки ИИ	Открывает окно «Настройки ИИ» — параметры подключения ИИ-ассистента к сервису большой языковой модели (подраздел 13.7). Кнопка отображается только администратору.
Выход	Завершает сеанс работы.
Название нового пайплайна…	Поле ввода названия создаваемого процесса.
+ Создать	Создаёт процесс с введённым названием и открывает его в визуальном редакторе.

3.2. Списки и постраничный вывод

Табличные списки Системы (процессы, пользователи, журнал событий, файлы текущей папки) имеют единое устройство: строка заголовков столбцов и строки записей;

при наведении указателя строка подсвечивается; кнопки действий расположены в правой части строки. Длинные списки прокручиваются вертикально.

Данные запрашиваются у сервера порциями: реестр процессов — до 200 процессов (новые сверху), история запусков — 50 последних запусков процесса, журнал событий — 300 последних записей. Программный интерфейс Системы поддерживает постраничную выборку списков процессов и запусков параметрами `limit` (размер страницы) и `offset` (смещение). Реестр процессов и история запусков обновляются автоматически каждые 5 секунд, журнал событий — каждые 10 секунд.

3.3. Диалоговые окна и подтверждения

Операции над объектами выполняются в диалоговых окнах, открывающихся поверх экрана. Окно закрывается кнопкой «Закрыть», «Отмена», значком «×» в заголовке или щелчком за пределами окна; окно выбора узла и панель настроек узла закрываются также клавишей Esc.

Удаление процессов, пользователей, файлов и папок выполняется только после подтверждения в стандартном окне браузера с текстом вида «Удалить «Название»?». Ввод имени новой папки, нового имени файла и нового пароля пользователя выполняется в стандартном окне ввода браузера.

3.4. Подвал и раздел «О программе»

Внизу страниц входа, реестра процессов, файлового хранилища и раздела администрирования выводится строка «© ООО «Кортекс», 2026» и ссылка «О программе». Страница «О программе» доступна без входа в Систему.

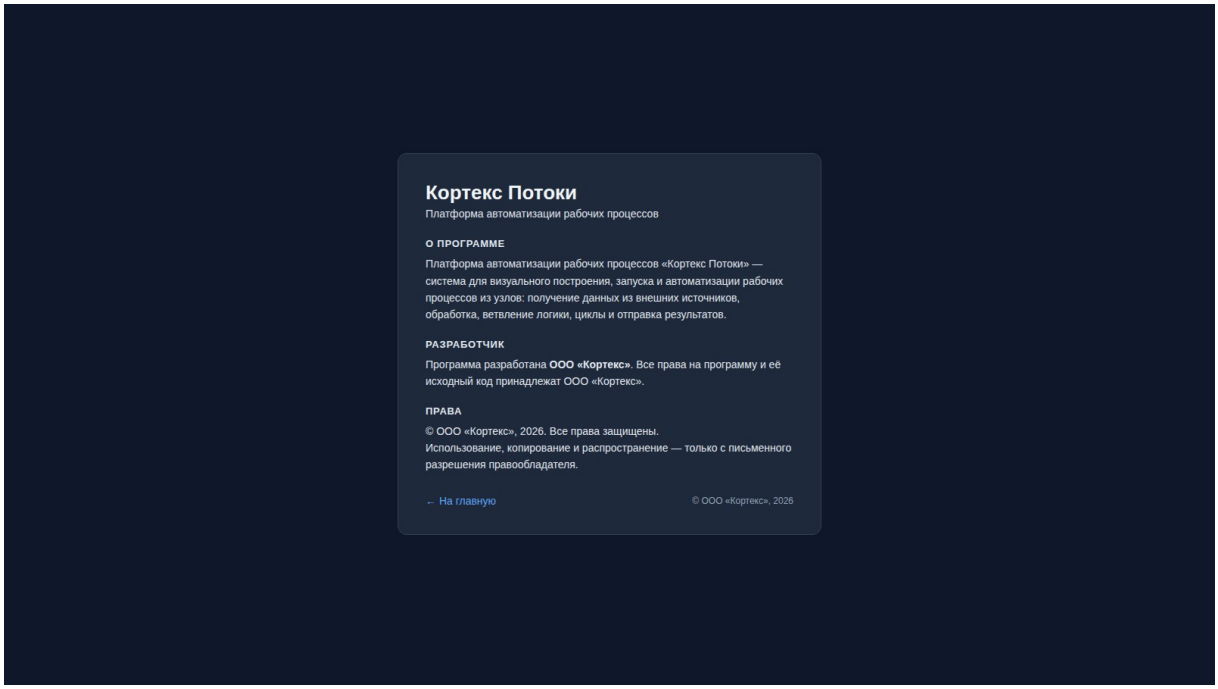


Рисунок 4 — Страница «О программе»

ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

Страница содержит полное наименование и краткое описание назначения Системы, сведения о разработчике и правообладателе — ООО «Кортекс», условия использования и ссылку «← На главную».

4. Роли пользователей

Роль	Доступные действия
Пользователь	Создание процессов и работа с собственными процессами (редактирование, запуск, просмотр журнала и истории запусков, восстановление версий, удаление); создание токенов вебхуков и настройка расписания в собственных процессах; предоставление другим пользователям доступа к собственным процессам на чтение или запись; работа с процессами, доступ к которым предоставлен другими пользователями, в пределах выданного уровня; использование ИИ-ассистента (после настройки администратором); работа с файловым хранилищем в веб-интерфейсе и по протоколу SFTP; просмотр глобальных констант.
Администратор	Все действия пользователя, а также: доступ ко всем процессам Системы с правами владельца; раздел «Админ» — управление пользователями (создание, назначение прав администратора, блокировка, сброс пароля, удаление с передачей процессов), системные настройки, журнал событий, диагностика; изменение глобальных констант; настройка подключения ИИ-ассистента к сервису большой языковой модели.

Первая учётная запись администратора создаётся при установке Системы. Роль «Администратор» назначается и снимается в разделе «Админ» → «Пользователи» (подраздел 13.2).

4.1. Уровни доступа к процессу

Каждый процесс имеет владельца — пользователя, создавшего процесс. Владелец может предоставить доступ к процессу другим пользователям (раздел 10). Уровень доступа определяет набор допустимых действий:

Действие	Владелец	Запись	Чтение
Просмотр процесса на холсте и настроек узлов	да	да	да
Просмотр истории запусков и выходных	да	да	да

Действие	Владелец	Запись	Чтение
данных			
Просмотр истории версий	да	да	да
Изменение и сохранение процесса	да	да	нет
Запуск процесса	да	да	нет
Восстановление версии	да	да	нет
Управление токенами вебхуков	да	нет	нет
Предоставление и отзыв доступа (кнопка «Доступ»)	да	нет	нет
Удаление процесса	да	нет	нет

Администратор имеет права владельца на все процессы Системы. Процессы, к которым у пользователя нет доступа, в его реестре не отображаются.

5. Реестр процессов

Раздел «Пайплайны» содержит реестр процессов, доступных пользователю: собственных процессов и процессов, к которым пользователю предоставлен доступ. Администратор видит все процессы Системы. Раздел открывается после входа и по кнопкам «← Пайплайны» и «← Назад» из других разделов.

5.1. Список процессов

Столбец / элемент	Назначение
Название	Название и описание процесса (если задано).
Статус	Признак активности для запуска по расписанию: «активен» или «неактивен» (подраздел 9.2).
Последний запуск	Состояние последнего запуска с цветовым индикатором — «успешно» (зелёный), «ошибка» (красный), «выполняется» (оранжевый), «ожидание» (серый) — и время запуска; «—», если запусков не было.
ID	Первые восемь символов идентификатора процесса (полностью — в адресной строке браузера при открытии процесса).
Дублировать / Удалить	Копирование (подраздел 5.3) и удаление (подраздел 5.4) процесса.

Щелчок по строке процесса открывает его в визуальном редакторе (раздел 6). Если процессов нет, выводится сообщение «Пайплайны ещё не созданы.».

5.2. Создание процесса

1. Введите название процесса в поле «Название нового пайплайна…» в шапке раздела «Пайплайны».
2. Нажмите «+ Создать» или клавишу Enter. Кнопка недоступна, пока поле пустое.
3. Процесс создаётся с пустым холстом, пользователь становится его владельцем, и процесс сразу открывается в визуальном редакторе.

5.3. Дублирование процесса

Кнопка «Дублировать» в строке процесса создаёт новый процесс с названием «<исходное название> (копия)», тем же описанием, графом узлов и связей и переменными процесса. Владельцем копии становится пользователь, выполнивший дублирование. История запусков, версии, токены вебхуков и выданные доступы в копию не переносятся. Дублирование удобно для создания процесса по образцу существующего.

5.4. Удаление процесса

Нажмите «Удалить» в строке процесса и подтвердите действие в окне «Удалить «Название»?». Вместе с процессом удаляются его история запусков, версии, токены вебхуков и выданные доступы; задания расписания процесса отменяются. Удалить процесс может только его владелец или администратор. Удаление процесса регистрируется в журнале событий.

6. Визуальный редактор процессов

6.1. Состав экрана

Визуальный редактор открывается при выборе процесса в реестре или сразу после создания процесса. Экран редактора состоит из панели инструментов (сверху), палитры узлов (слева), холста (в центре), панели «Лог запуска» (под холстом) и, при включении, панели ИИ-ассистента (справа).

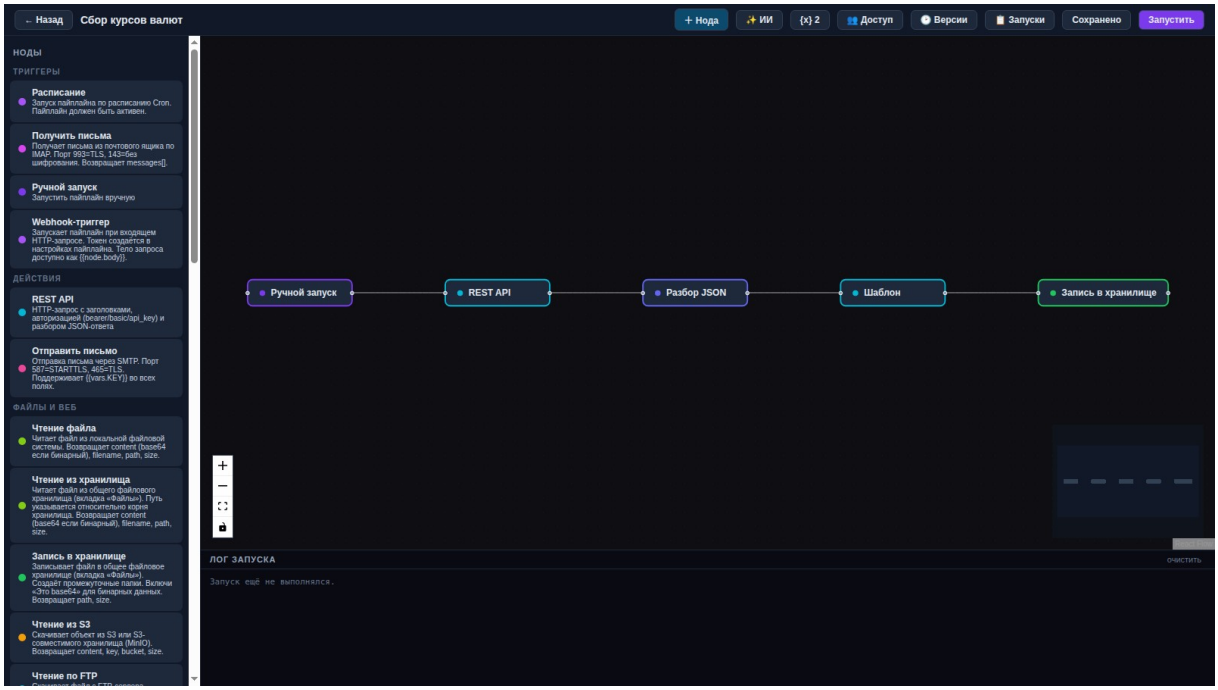


Рисунок 5 — Визуальный редактор процесса

6.2. Панель инструментов

Кнопка	Назначение
← Назад	Возврат в раздел «Пайплайны»; несохранённые изменения не сохраняются. Справа от кнопки — название процесса.
＋ Нода	Окно «Добавить ноду» (подраздел 6.5).
ИИ	Панель ИИ-ассистента (раздел 12); сочетание клавиш Ctrl+I (на компьютерах Apple — ⌘+I).
Переменные	Окно «Переменные пайплайна» (подраздел 6.9); если переменные заданы, вместо слова выводится их количество.
Доступ	Окно «Доступ к пайплайну» (раздел 10); отображается только владельцу процесса и администратору.
Версии / Запуски	Окна «История версий» (подраздел 10.3) и «История запусков» (подраздел 8.5).
Сохранено / Сохранить* / Сохранение…	Сохранение: «Сохранено» — изменений нет; «Сохранить*» — есть несохранённые изменения; «Сохранение…» — идёт сохранение.
Запустить / ⟳ Выполняется…	Проверка графа и запуск процесса (раздел 8); во время выполнения кнопка недоступна.

ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

6.3. Холст

На холсте отображаются узлы процесса и связи между ними. Каждый узел имеет цветной индикатор типа, название и точки подключения связей: вход — слева, выходы — справа.

Действие	Как выполнить
Переместить узел или холст	Перетащить узел или пустую область холста мышью.
Выделить узел или связь	Щелчок; несколько элементов — рамкой при нажатой клавише Shift.
Открыть настройки узла	Двойной щелчок по узлу (подраздел 6.7).
Удалить узел или связь	Выделить и нажать Delete; при удалении узла удаляются и его связи.
Изменить масштаб, показать весь процесс	Колесо мыши или кнопки «+», «−» и кнопка с изображением рамки в левом нижнем углу холста; мини-карта в правом нижнем углу показывает весь граф.

Любое изменение на холсте (добавление, перемещение, удаление узла или связи, изменение настроек) помечает процесс как изменённый: на кнопке сохранения появляется надпись «Сохранить*».

6.4. Палитра узлов

Палитра «Ноды» слева от холста содержит все доступные типы узлов, сгруппированные по категориям: «Триггеры», «Действия», «Файлы и веб», «Логика», «Данные», «AI», «Вывод». Для каждого узла выводятся цветной индикатор, название и краткое описание. При наведении указателя рамка элемента окрашивается в цвет узла.

Чтобы добавить узел, перетащите его из палитры на холст и отпустите в нужном месте. Узлу автоматически присваиваются идентификатор (по типу узла, например `rest_api`; при повторном добавлении — `rest_api_1`, `rest_api_2` и т. д.) и название (название типа; при повторе — с порядковым номером). Идентификатор используется в подстановках значений (подраздел 6.8), название можно изменить в панели настроек узла.

6.5. Окно выбора узла

Кнопка «＋ Нода» открывает окно «Добавить ноду» для быстрого поиска узла. Поле «Поиск по названию или описанию…» отбирает узлы по всем категориям (если совпадений нет, выводится «Ничего не найдено»); вкладки «Все», «Триггеры», «Действия», «Файлы и веб», «Логика», «Данные», «AI», «Вывод» ограничивают список категорией. Щелчок по карточке узла добавляет узел в центр видимой области холста и закрывает окно. Окно закрывается без добавления узла кнопкой «×», клавишей Esc или щелчком вне окна.

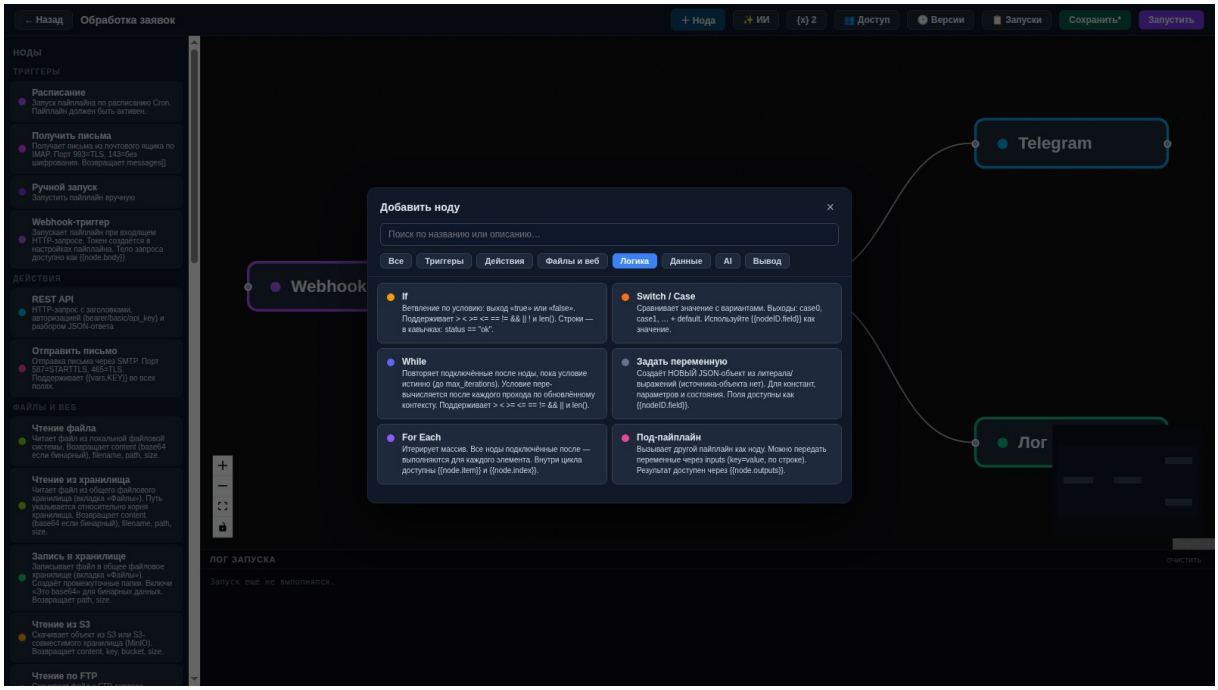


Рисунок 6 — Окно «Добавить ноду»

6.6. Связи между узлами

Связь задаёт порядок выполнения и передачу данных от узла к узлу. Чтобы создать связь, нажмите кнопку мыши на точке выхода узла (справа) и протяните линию к точке входа другого узла (слева). Узлы-триггеры не имеют входа — с них начинается процесс.

Большинство узлов имеет один выход. У узла «If» два выхода — «true» и «false»: выполняются узлы, подключённые к выходу, соответствующему результату условия. У узла «Switch / Case» три выхода — «case0» (первый вариант), «case1» (второй вариант) и «default» (совпадений нет). Узлы на невыбранном выходе и ниже по этой ветви не выполняются и отмечаются в журнале как пропущенные.

6.7. Панель настроек узла

Двойной щелчок по узлу открывает панель настроек узла в центре экрана. Изменения применяются к узлу сразу при вводе; для записи в базу данных нажмите на панели инструментов «Сохранить*».

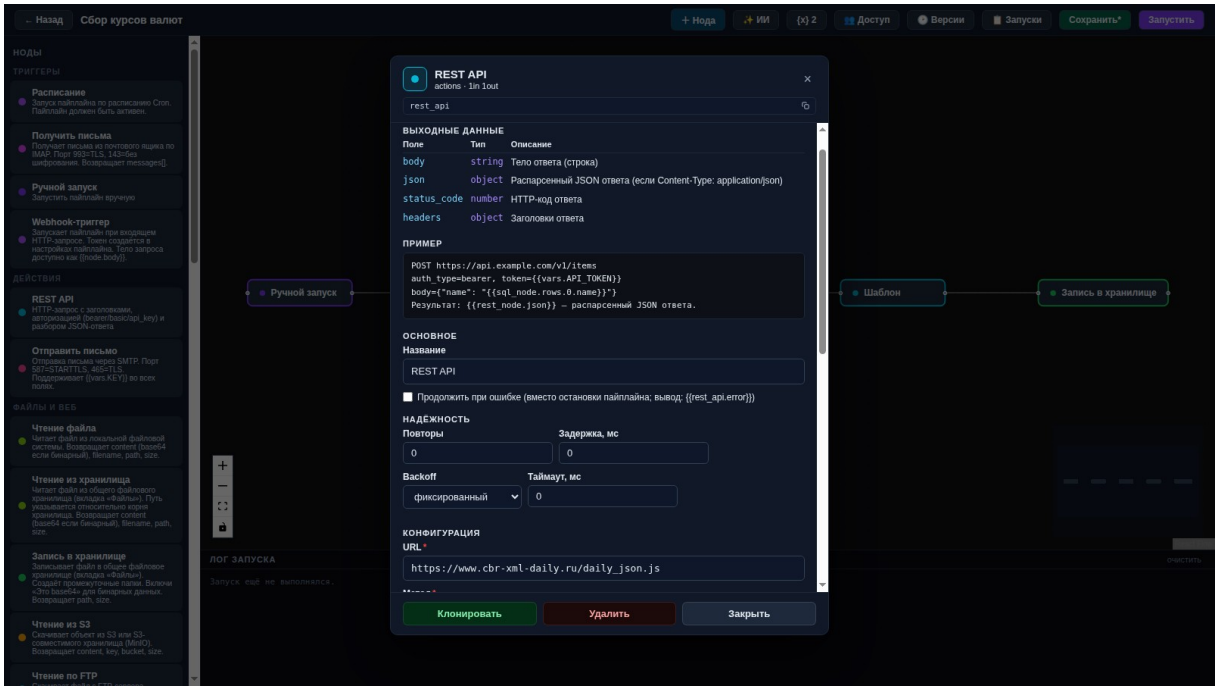


Рисунок 7 — Панель настроек узла «REST API»

Блок / элемент	**Назначение**
Заголовок	Название узла, обозначение категории, число входов и выходов, строка с идентификатором узла (например, `rest_api`); щелчок по идентификатору копирует его в буфер обмена.
Описание, Пример	Назначение узла и пример заполнения параметров.
Выходные данные	Таблица «Поле / Тип / Описание» — поля, которые узел передаёт следующим узлам; на них ссылаются подстановкой `{{идентификатор.поле}}`.
Основное	Поле «Название» — отображаемое название узла; флажок «Продолжить при ошибке» (кроме триггеров) — ошибка узла не останавливает процесс, её текст передаётся в поле `{{идентификатор.error}}`.
Надёжность	Параметры повторов и таймаута узла (кроме триггеров) — см. ниже.
Конфигурация	Параметры, специфичные для типа узла (раздел 7); обязательные отмечены «*».
Последний вывод	Выходные данные узла после запуска (подраздел 8.4).
Webhook URL	Только для узла «Webhook-триггер» — адреса вебхука (подраздел 9.1).
Клонировать / Удалить / Закрыть	Копирование узла со всеми настройками, удаление узла и его связей, закрытие панели (также клавиша Esc).

Параметры секции «Надёжность»: «Повторы» — число повторных попыток после ошибки (0 — без повторов); «Задержка, мс» — пауза перед повтором; «Backoff» — «фиксированный» (одинаковая пауза) или «экспоненциальный» (пауза удваивается с каждой попыткой); «Таймаут, мс» — предельное время одной попытки (0 — без ограничения).

6.8. Подстановка значений и подсказки

В текстовых полях настроек узла можно использовать ссылки на результаты других узлов и на переменные:

Запись	Значение
`{{идентификатор.поле}}`	Поле выходных данных узла, например `{{rest_api.status_code}}`; вложенные значения — через точку, номера элементов массивов с нуля: `{{sql.rows.0.name}}`.
`{{идентификатор}}`	Все выходные данные узла целиком.
`{{vars.ИМЯ}}`	Значение переменной процесса или глобальной константы с именем ИМЯ.

Объекты и массивы подставляются в формате JSON, числа и строки — как текст. Ссылка, для которой значение не найдено, остаётся в тексте без изменений.

При вводе символов `{{` под полем открывается список подсказок: переменные процесса (группа «⚙️ Переменные пайплайна») и поля выходных данных всех остальных узлов процесса, сгруппированные по названиям узлов. Список сужается по мере ввода. Выбор подсказки — клавишами ↑ и ↓ и Enter или Tab либо щелчком мыши; Esc закрывает список. В поле ссылки на узлы подсвечиваются голубым цветом и отображаются с названием узла вместо идентификатора.

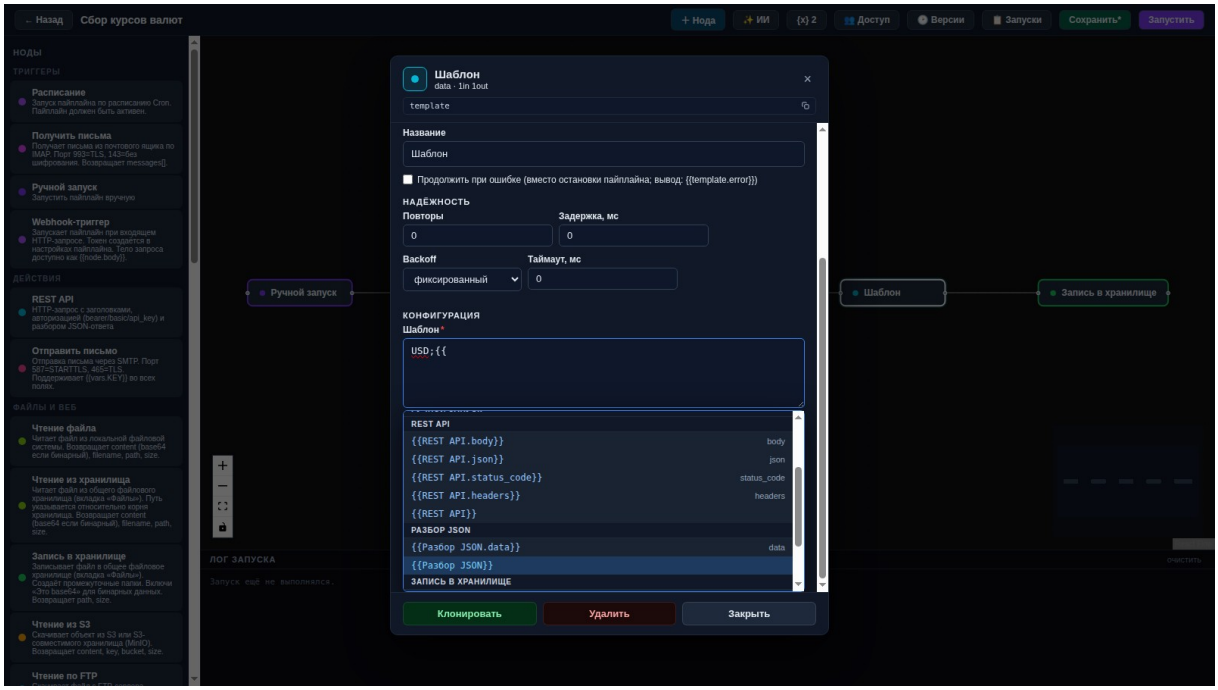


Рисунок 8 — Подсказки подстановки значений

6.9. Переменные процесса

Переменные процесса — именованные значения (адреса, логины, параметры), которые используются в узлах записью `{{vars.ИМЯ}}`. Переменные хранятся вместе с процессом и копируются при его дублировании. Если имя переменной процесса совпадает с именем глобальной константы (подраздел 13.6), используется значение переменной процесса.

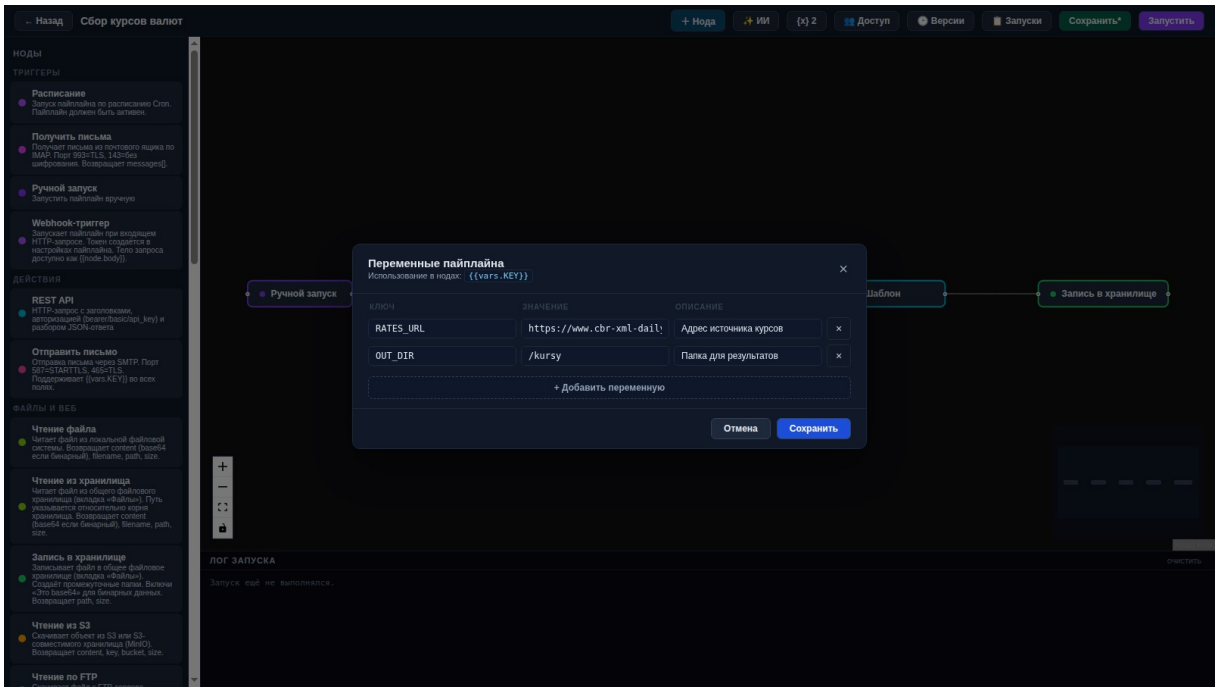


Рисунок 9 — Окно «Переменные пайплайна»

Окно открывается кнопкой «Переменные». Для каждой переменной задаются «Ключ» (имя латинскими буквами, цифрами и знаком подчёркивания, например `API_TOKEN`), «Значение» и необязательное «Описание». Кнопка «+ Добавить переменную» добавляет строку, «×» — удаляет её; строки с пустым ключом отбрасываются. «Сохранить» применяет изменения к процессу (для записи в базу данных затем нажмите «Сохранить*» на панели инструментов), «Отмена» закрывает окно без изменений.

6.10. Проверка графа перед запуском

При нажатии «Запустить» Система проверяет граф процесса. Если обнаружены ошибки, под панелью инструментов выводится красная панель «⚠️ Пайплайн не готов к запуску:» со списком ошибок, запуск не выполняется.

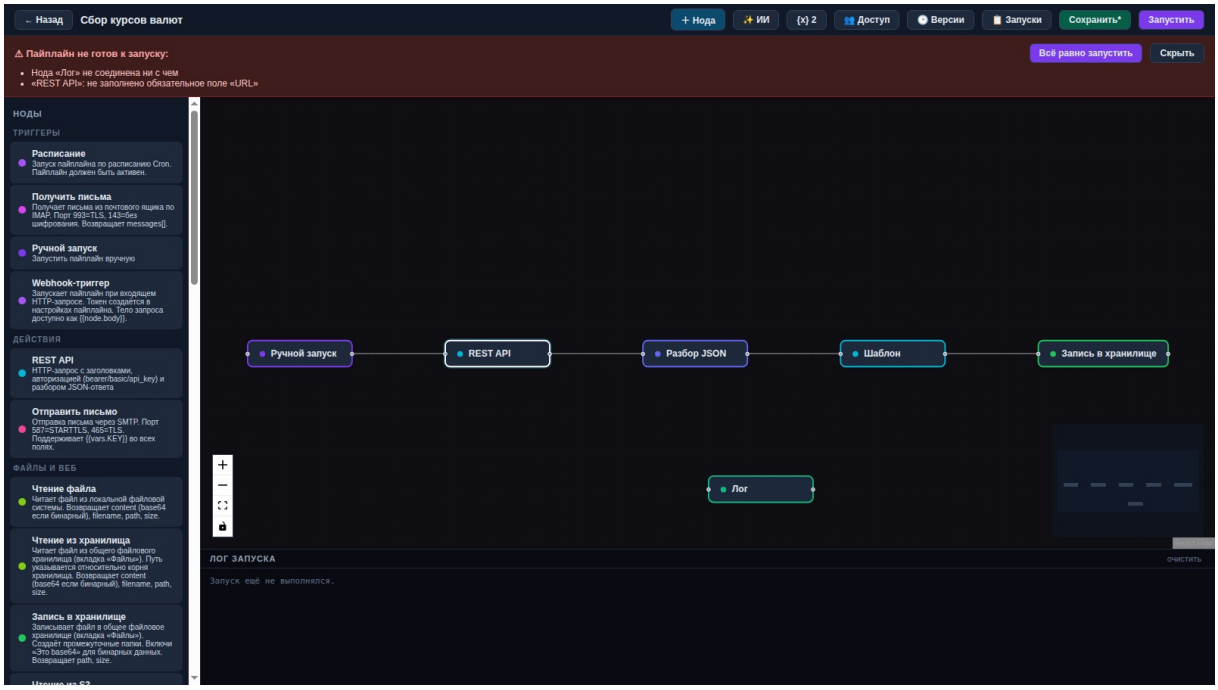


Рисунок 10 — Панель ошибок проверки графа

Сообщение	Причина и действия
Граф пуст — добавьте ноды	На холсте нет ни одного узла.
Нет триггера — добавьте стартовую ноду (например, «Запуск вручную»)	Нет узла из категории «Триггеры».
Нода «Название» не соединена ни с чем	Узел без связей; соедините или удалите его.
«Название»: не заполнено обязательное поле «Поле»	Не заполнен обязательный параметр узла (отмечен «*»).

Кнопка «Всё равно запустить» запускает процесс без исправления (несвязанные узлы при выполнении пропускаются), «Скрыть» закрывает панель.

6.11. Сохранение процесса

Нажмите «Сохранить*» на панели инструментов. Сохраняются граф процесса (узлы, их настройки и расположение, связи) и переменные процесса. Если граф

изменился по сравнению с сохранённым ранее, Система автоматически создаёт новую версию процесса (подраздел 10.3). После успешного сохранения кнопка принимает вид «Сохранено».

Запуск процесса выполняется по сохранённому на сервере состоянию. Перед запуском сохраните изменения.

7. Библиотека узлов

7.1. Общие сведения

Библиотека узлов содержит готовые функциональные блоки, из которых строятся процессы. Узлы сгруппированы по категориям палитры. Для каждого узла ниже приведены назначение, основные параметры секции «Конфигурация» (обязательные отмечены «*») и выходные данные, доступные следующим узлам. Во всех текстовых параметрах допускается подстановка `{{идентификатор.поле}}` и `{{vars.ИМЯ}}`.

Идентификатор узла в примерах совпадает с идентификатором, который Система присваивает первому узлу данного типа (например, `rest_api`, `parse_json`, `for_each`).

7.2. Триггеры

Узел	Назначение	Основные параметры	Выходные данные
Ручной запуск	Начало процесса, запускаемого кнопкой «Запустить» в редакторе.	—	triggered_at — время запуска.
Расписание	Запуск по расписанию cron (подраздел 9.2); процесс должен быть активен.	Cron-выражение*, Описание.	triggered_at — время срабатывания.
Webhook-триггер	Запуск процесса HTTP-запросом POST на адрес с токеном (подраздел 9.1).	Блок «Webhook URL».	Поля JSON-тела запроса — как `{{webhook_trigger.имя_поля}}`; triggered_at — время срабатывания.
Получить письма	Получение писем из почтового ящика по протоколу IMAP при каждом выполнении узла (подраздел 9.3).	IMAP Host*, Порт* (993 — TLS, 143 — без шифрования), Имя пользователя*, Пароль*, Использовать TLS, Почтовый ящик, Только непрочитанные, Макс. писем.	messages — массив писем (subject, from, body, date, uid); count — количество писем.

7.3. Действия

Узел	Назначение	Основные параметры	Выходные данные
REST API	HTTP-запрос к веб-сервису с заголовками, авторизацией и разбором JSON-ответа.	URL*, Метод* (GET, POST, PUT, PATCH, DELETE), Тип авторизации (none, bearer, basic, api_key), Token / API Key, API Key Header, Basic Auth: пользователь, Basic Auth: пароль, Headers (JSON), Тело запроса, Таймаут (сек), по умолчанию 30.	body — тело ответа; json — ответ, разобранный как JSON (при ответе в формате JSON); status_code — код ответа; headers — заголовки ответа.
Отправить письмо	Отправка письма по протоколу SMTP.	SMTP Host*, Порт* (587 — STARTTLS, 465 — TLS), Имя пользователя, Пароль, SSL/TLS (порт 465), От, Кому (через запятую)*, Тема*, Текст*, HTML-тело.	sent — признак успешной отправки.

7.4. Файлы и веб

Узел	Назначение	Основные параметры	Выходные данные
Чтение файла	Чтение файла из файловой системы сервера.	Путь к файлу*.	content (для двоичных файлов — base64); filename; path; size; is_binary.
Чтение из хранилища	Чтение файла из общего файлового хранилища (раздел «Файлы»).	Путь в хранилище* от корня, например `/reports/export.csv`.	content; filename; path; size; is_binary.
Запись в хранилище	Запись файла в хранилище; папки создаются автоматически, существующий файл перезаписывается.	Путь в хранилище*, Содержимое, Это base64 (бинарный файл).	path; size.
Чтение из S3	Получение объекта из S3-совместимого хранилища объектов.	Бакет*, Ключ объекта*, Регион, Access Key ID, Secret	content; key; bucket; size; content_type; is_binary.

Узел	Назначение	Основные параметры	Выходные данные
		Access Key, Endpoint — адрес хранилища в инфраструктуре заказчика.	
Чтение по FTP	Получение файла с FTP-сервера.	Host*, Порт (по умолчанию 21), Имя пользователя, Пароль, Путь на сервере*.	content; path; size; is_binary.
Загрузка HTML	Загрузка веб-страницы по адресу.	URL*.	html; url; title; text — текст без разметки; links — ссылки (href, text); status_code.

7.5. Логика

Узел	Назначение	Основные параметры	Выходные данные
If	Ветвление по условию: выход «true» или «false» (подраздел 7.10).	Выражение*, например `rest_api.status_code == 200`.	result — результат условия.
Switch / Case	Выбор ветви по значению: значение сравнивается с вариантами по порядку.	Значение*, например `{{webhook_trigger.priority}}`; Варианты (через запятую)*.	matched — имя сработавшего выхода (case0, case1 или default); value — сравниваемое значение.
While	Цикл «пока»: узлы после узла выполняются повторно, пока условие истинно (проверяется после каждого прохода).	Условие*, Макс. итераций (по умолчанию 100).	iterations; timed_out — достигнут предел; completed — завершён по условию.
Задать переменную	Создание объекта с полями из JSON — для параметров и промежуточного состояния.	Данные (JSON-объект)*, например `{"page": 1, "limit": 100}`.	Каждое поле объекта — как `{{set_variable.имя_поля}}`.
For Each	Цикл «для каждого»: вся цепочка узлов после узла выполняется для каждого элемента массива.	Массив элементов (JSON)*, например `{{set_variable.requests}}`; Параллельность (1 = последовательно), не более 32.	item, index — текущий элемент и его номер; results; count.

Узел	Назначение	Основные параметры	Выходные данные
Под-пайплайн	Вызов другого процесса как узла с возвратом его результатов.	Пайплайн* — выбор из списка; Входные переменные (key=value) — по строке, в вызываемом процессе доступны как `{{vars.key}}`.	outputs — выходные данные узлов вызванного процесса; success.

Глубина вложенности вызовов ограничена десятью уровнями: при взаимном вызове процессов (А → Б → А) выполнение прерывается с ошибкой.

7.6. Данные

Узел	Назначение	Основные параметры	Выходные данные
SQL-запрос	Выполнение запроса к внешней базе данных PostgreSQL или MySQL.	Драйвер* (postgres, mysql), DSN подключения*, Запрос*, Параметры (JSON-массив или CSV), JSON Param ($1).	rows — строки SELECT; row_count; rows_affected — для INSERT, UPDATE, DELETE; columns.
Выборка из HTML	Извлечение элементов HTML-страницы по селектору CSS или XPath.	HTML-содержимое*, CSS / XPath селектор*, Base URL (для относительных ссылок).	items (text, href); items_json — JSON-строка; count.
Разбор HTML	Разбор всей HTML-страницы: заголовок, текст, ссылки.	HTML-содержимое*.	title; text; links; html.
Разбор JSON	Разбор строки в формате JSON с необязательным извлечением вложенного значения.	JSON-содержимое*, Path (необязательно), например `.items.0.name`.	data.
Разбор CSV	Разбор текста в формате CSV или TSV в массив записей.	CSV-содержимое*, Разделитель (по умолчанию запятая), Первая строка — заголовок, Пропустить строк.	rows — объекты «заголовок: значение»; headers; row_count.
Разбор XML	Разбор текста в формате XML во вложенный объект.	XML-содержимое*.	data; атрибуты — ключ `-attr`, текст — `#text`.
Разбор Excel	Разбор файла электронной таблицы	Путь к файлу или Base64-содержимое;	rows; headers; row_count; sheet;

Узел	Назначение	Основные параметры	Выходные данные
	формата .xlsx.	Имя листа; Первая строка — заголовок.	sheets.
Шаблон	Формирование текстовой строки по шаблону с подстановкой значений других узлов.	Шаблон*.	text — результат подстановки.
Изменить поля	Преобразование существующего объекта: добавление, переименование, удаление полей.	Источник (JSON объект); Поля (key=value, по строке)*: `ключ=значение` — добавить или заменить, `-ключ` — удалить.	data; json.

7.7. Запрос к большой языковой модели (категория «AI»)

Узел	Назначение	Основные параметры	Выходные данные
Вызов LLM	Отправка текстового запроса сервису большой языковой модели (внешнему или размещённому в инфраструктуре заказчика) с совместимым программным интерфейсом и получение текста ответа — например, для составления краткого изложения, классификации или извлечения сведений из текста.	Провайдер* — тип программного интерфейса сервиса (выбирается из списка); Модель* — имя модели, предоставляемой сервисом; API Key — ключ доступа (рекомендуется указывать через `{{vars.ИМЯ}}`); API URL (необязательно) — адрес сервиса; Системный промпт; Промпт* — текст запроса с подстановками; Max Tokens; Температура (0–2).	text — текст ответа; content — то же; model — имя ответившей модели; usage — количество обработанных токенов.

7.8. Вывод

Узел	Назначение	Основные	Выходные данные

Узел	Назначение	Основные параметры	Выходные данные
		параметры	
Лог	Запись сообщения в журнал запуска; при включённом дампе — вывод всех данных процесса в данной точке для отладки.	Сообщение, Дамп контекста.	logged — записанное сообщение; context_dump — снимок данных в формате JSON (при включённом дампе).
Telegram	Отправка сообщения в мессенджер Telegram от имени бота.	Bot Token*, Chat ID*, Текст сообщения*, Parse Mode (HTML, Markdown, MarkdownV2).	ok — признак успешной отправки; message_id — идентификатор сообщения.

7.9. Правила подстановки значений

- ссылка на узел действует только на узлы, выполненные ранее в том же запуске (расположенные выше по связям);
- внутри цикла «For Each» текущий элемент доступен как `{{for_each.item}}`, его поля — как `{{for_each.item.имя_поля}}`, номер — как `{{for_each.index}}`;
- переменные `{{vars.ИМЯ}}` берутся из переменных процесса, а при их отсутствии — из глобальных констант;
- объекты и массивы подставляются в формате JSON, поэтому результат одного узла можно передать в поле «Массив элементов (JSON)», «JSON-содержимое» или «Данные (JSON-объект)» другого узла без преобразования;
- секретные значения (пароли, ключи, токены) рекомендуется хранить в переменных процесса или глобальных константах и указывать в узлах ссылкой `{{vars.ИМЯ}}`.

7.10. Выражения условий

В узлах «If» и «While» условие записывается выражением над выходными данными узлов и переменными без фигурных скобок: `идентификатор.поле` или `vars.ИМЯ`.

Элемент	Запись	Пример
Сравнение	`==`, `!=`, `>`, `<`, `>=`, `<=`	`rest_api.status_code == 200`
Логическое «и», «не»	`&&`, `!`	`sql.row_count > 0 && vars.MODE == "prod"`
Длина массива или строки	`len()`	`len(scrape_html.items) > 0`
Строковые значения	в двойных кавычках	`webhook_trigger.priority == "high"`

ООО «КОРТЕКС»
350063, г. Краснодар, ул. Северная, д. 324, офис 12/2
www.cortex.ooo

Логическое «или» записывается двумя вертикальными чертами: `rest_api.status_code == 200 || rest_api.status_code == 201`.

8. Выполнение и мониторинг

8.1. Запуск процесса

Процесс запускается:

- вручную — кнопкой «Запустить» в редакторе (уровень доступа «владелец» или «запись»); выполнение идёт асинхронно, ход выполнения отображается на холсте и в панели «Лог запуска»;
- по расписанию — узлом «Расписание» в активном процессе (подраздел 9.2);
- по вебхуку — HTTP-запросом POST на адрес вебхука (подраздел 9.1);
- из другого процесса — узлом «Под-пайплайн»; вызываемый процесс выполняется синхронно и возвращает результаты;
- через программный интерфейс — асинхронно запросом POST `/api/v1/pipelines/{идентификатор}/run` или синхронно запросом POST `/api/v1/pipelines/{идентификатор}/run-sync`, ответ которого содержит выходные данные всех узлов.

Узел выполняется после всех узлов, от которых к нему ведут связи. Каждый запуск регистрируется в истории запусков процесса.

8.2. Отображение хода выполнения

Во время выполнения рамка и индикатор каждого узла окрашиваются в цвет его состояния, справа от названия выводится знак: выполняется — оранжевый, «⟳»; выполнен успешно — зелёный, «✓»; завершился с ошибкой — красный, «✗»; пропущен (невыбранная ветвь условия) — серый, «↷».

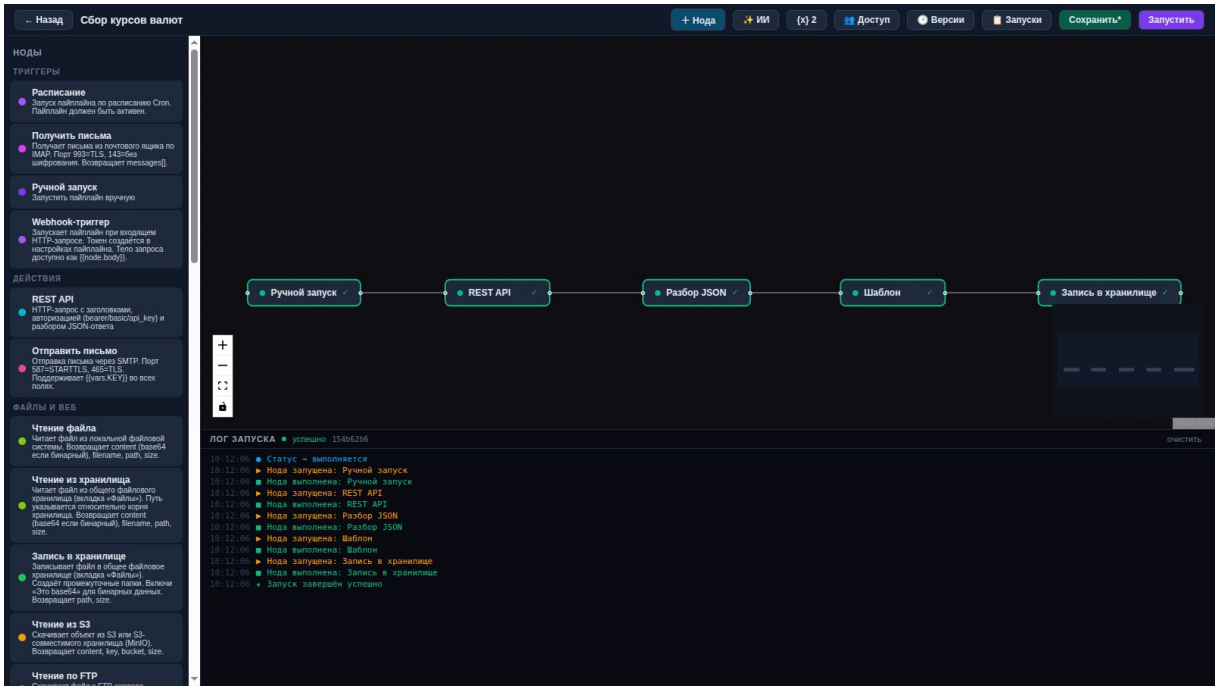


Рисунок 11 — Ход выполнения процесса

8.3. Журнал запуска

Панель «Лог запуска» под холстом отображает события текущего запуска в реальном времени с указанием времени каждого события. Высота панели изменяется перетаскиванием верхней границы, кнопка «очистить» очищает панель и сбрасывает окраску узлов. До первого запуска выводится «Запуск ещё не выполнялся.».

Запись журнала	Значение
Статус → выполняется; Нода запущена: <название>	Запуск начался; начато выполнение узла.
Нода выполнена: <название> / Нода пропущена: <название>	Узел выполнен успешно / не выполнялся (невыбранная ветвь).
Нода завершилась с ошибкой: <название> — <текст>	Ошибка выполнения узла.
повтор 2/3 после ошибки: <текст>; итерация 1 / 5	Повторная попытка выполнения узла; очередная итерация цикла.
<сообщение>; Debug [<название>]	Сообщение узла «Лог»; дамп данных процесса в формате JSON (параметр «Дамп контекста»).
Запуск завершён успешно / Запуск завершился с ошибкой: <текст>	Итог запуска.

8.4. Выходные данные узла

После запуска откройте панель настроек узла двойным щелчком. В нижней части панели появится раскрывающийся блок «Последний вывод» с числом полей — результат выполнения узла в последнем запуске: имя поля, тип значения и значение. Для массивов показываются первые три элемента, для длинных строк — первые 300 символов; ссылка «развернуть» у поля показывает значение полностью, «свернуть» —

сворачивает. Щелчок по заголовку «Последний вывод» сворачивает и разворачивает блок.

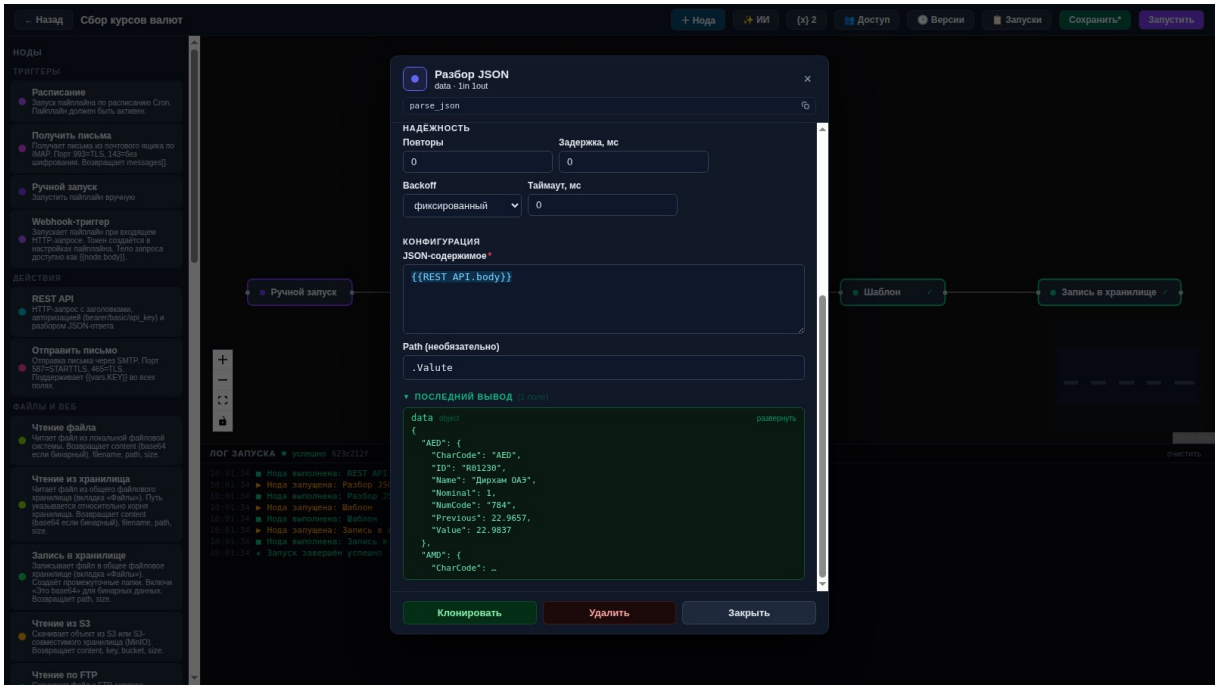


Рисунок 12 — Выходные данные узла после запуска

Выходные данные узлов используются для проверки правильности настройки: по ним определяются имена полей для подстановки в следующие узлы.

8.5. История запусков

Кнопка «Запуски» открывает окно «История запусков» с подписью «Выберите запуск, чтобы загрузить его выходные данные на холст». В списке — 50 последних запусков процесса (в том числе по расписанию и по вебхуку): статус с цветовым индикатором («успешно», «ошибка», «выполняется», «ожидание»), для запусков с ошибкой — текст ошибки, дата и время, длительность. Список обновляется автоматически; если запусков нет, выводится «Запусков пока нет».

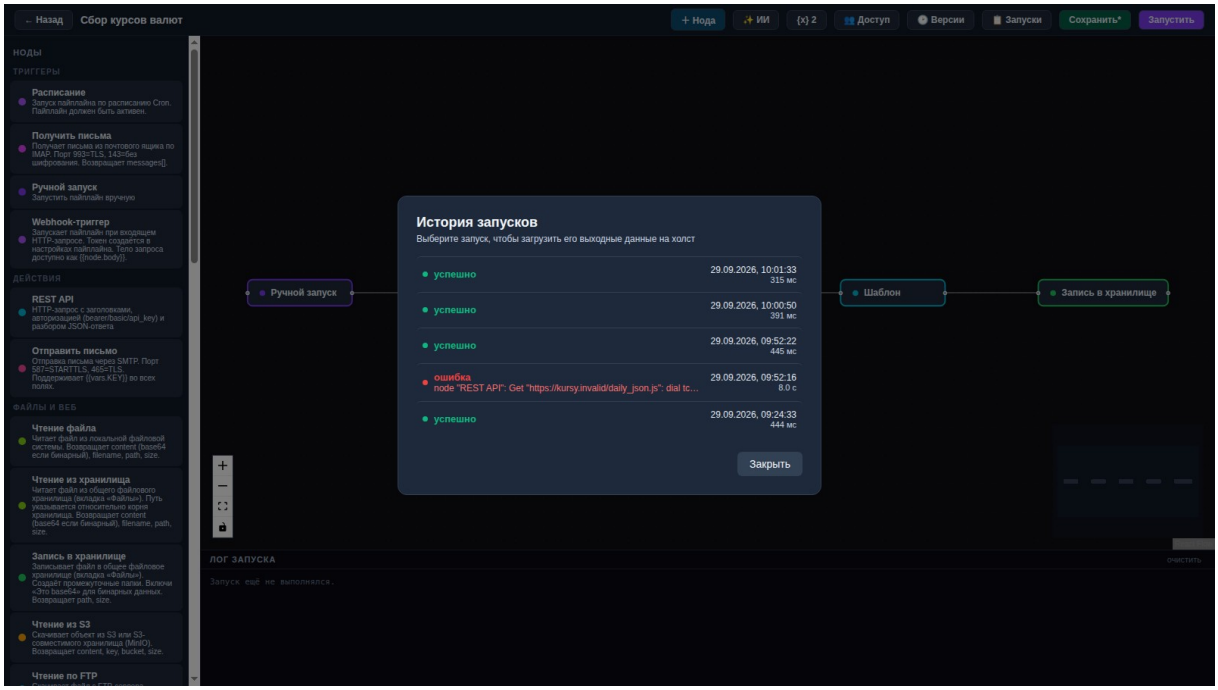


Рисунок 13 — Окно «История запусков»

Щелчок по строке запуска загружает сохранённые выходные данные узлов: узлы с результатами отмечаются на холсте зелёным, их данные доступны в блоке «Последний вывод» панели настроек узла. Выходные данные сохраняются для успешно завершённых запусков.

8.6. Надёжность выполнения

Для каждого узла задаются повторы с задержкой, таймаут и продолжение при ошибке (подраздел 6.7). Общая продолжительность запуска ограничена параметром «Таймаут запуска (сек)» (по умолчанию 300 секунд, раздел «Админ» → «Настройки»): запуск, превысивший предел, прерывается со статусом «ошибка». Вложенные вызовы процессов узлом «Под-пайплайн» ограничены десятью уровнями, цикл «While» — параметром «Макс. итераций».

8.7. Срок хранения истории

История запусков и версии процессов очищаются автоматически по правилам, установленным администратором (подраздел 13.3): по возрасту запусков (по умолчанию 30 дней), по числу хранимых запусков на процесс (по умолчанию 200) и по числу хранимых версий на процесс (по умолчанию 50). Выполняющиеся запуски не удаляются.

9. Внешний запуск и расписание

9.1. Запуск по вебхуку

Вебхук позволяет внешней информационной системе запускать процесс HTTP-запросом и передавать в него данные.

1. Добавьте в процесс узел «Webhook-триггер», соедините его со следующими узлами и сохраните процесс.
2. Откройте панель настроек узла «Webhook-триггер».
3. В блоке «Webhook URL» нажмите «+ Новый токен». Появится адрес вида `https://<адрес Системы>/api/v1/webhooks/<токен>` (пока токенов нет, выводится «Нет токенов. Создайте для получения URL.»).
4. Нажмите «copy» — адрес будет скопирован в буфер обмена. Кнопка «×» удаляет токен, после чего запросы на этот адрес не принимаются. У процесса может быть несколько адресов, например для разных внешних систем.

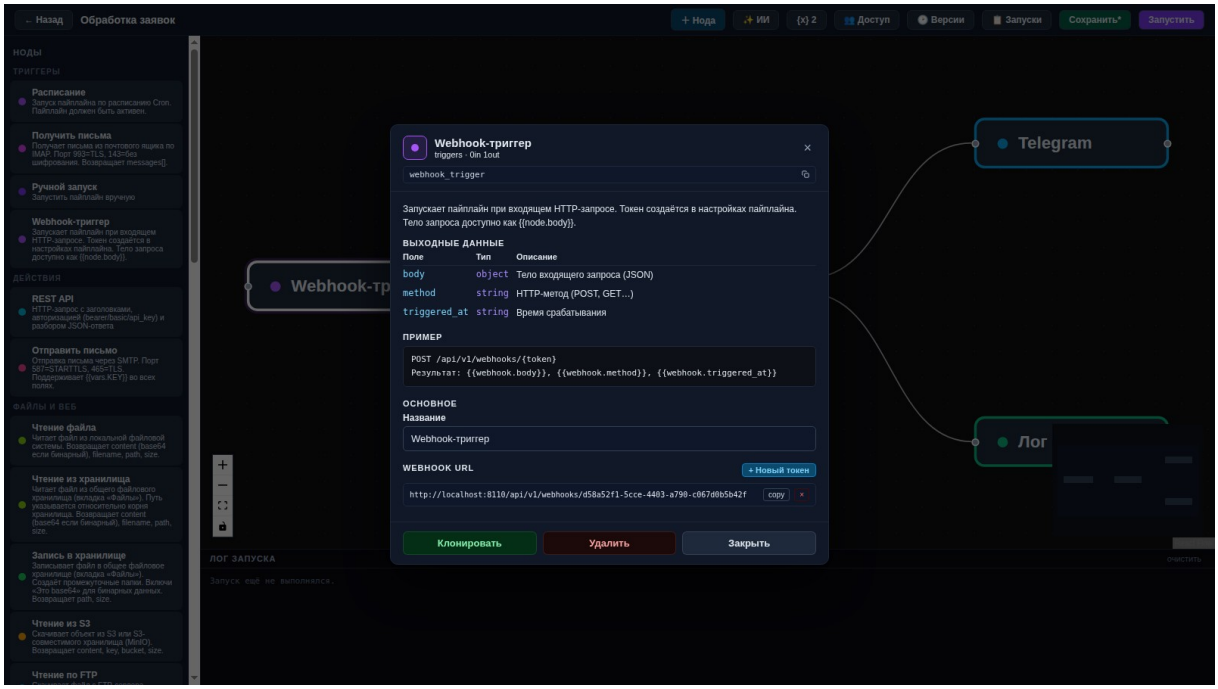


Рисунок 14 — Блок «Webhook URL» узла «Webhook-триггер»

Управлять токенами вебхука может владелец процесса и администратор.

Параметр запроса	**Значение**
Метод	POST; авторизация не требуется — доступ определяется секретным токеном в адресе.
Тело	Объект JSON (заголовок `Content-Type: application/json`); поля объекта доступны в процессе как `{{webhook_trigger.имя_поля}}`.
Ответ	Код 201 и описание созданного запуска; процесс выполняется асинхронно, результат — в истории запусков.
Ошибки	404 — неизвестный токен; 429 и сообщение «Превышена частота запросов к вебхуку» — превышена частота (по умолчанию не более 60 запросов в минуту на токен; значение задаётся при установке).

Пример запроса:

```
curl -X POST "https://<адрес Системы>/api/v1/webhooks/<токен>" \
  -H "Content-Type: application/json" \
  -d '{"number": 1024, "priority": "high", "text": "Не работает принтер в каб.
12"}'
```

9.2. Запуск по расписанию

Для периодического запуска добавьте в процесс узел «Расписание» и укажите в поле «Cron-выражение» расписание в стандартном формате cron из пяти полей: минута, час, день месяца, месяц, день недели. Время расписания — системное время сервера.

Примеры: `0 9 * * 1-5` — каждый будний день в 9:00; `*/15 * * * *` — каждые 15 минут; `30 7 1 * *` — первого числа месяца в 7:30; `@daily` — раз в сутки в 00:00.

Запуск по расписанию выполняется только для процессов со статусом «активен» (столбец «Статус» реестра процессов). Чтобы включить запуск по расписанию, щёлкните в строке процесса по отметке «неактивен» — она сменится на «активен». Повторный щелчок выключает запуск по расписанию. Изменять статус может пользователь с уровнем доступа «владелец» или «запись».

После изменения статуса или сохранения процесса расписание обновляется автоматически, перезапуск Системы не требуется. Ближайшие запланированные запуски отображаются администратору в разделе «Админ» → «Диагностика», карточка «Планировщик». Запуски по расписанию отображаются в истории запусков процесса.

9.3. Получение почты

Узел «Получить письма» при каждом выполнении подключается к почтовому ящику по протоколу IMAP и возвращает массив писем. Для регулярной обработки входящей почты постройте процесс «Расписание» → «Получить письма» → «For Each» → обработка каждого письма, включите параметр «Только непрочитанные» и сделайте процесс активным. Логин и пароль почтового ящика рекомендуется хранить в переменных процесса и указывать как `{{vars.EMAIL_USER}}` и `{{vars.EMAIL_PASS}}`.

10. Совместный доступ и версии

10.1. Предоставление доступа

Владелец процесса может предоставить доступ к процессу другим зарегистрированным пользователям.

1. Откройте процесс и нажмите «Доступ» на панели инструментов.
2. В окне «Доступ к пайплайну» введите адрес коллеги в поле «email пользователя».
3. Выберите уровень «Чтение» или «Запись» (подраздел 4.1) и нажмите «Добавить». Для пользователя, уже имеющего доступ, уровень изменяется.

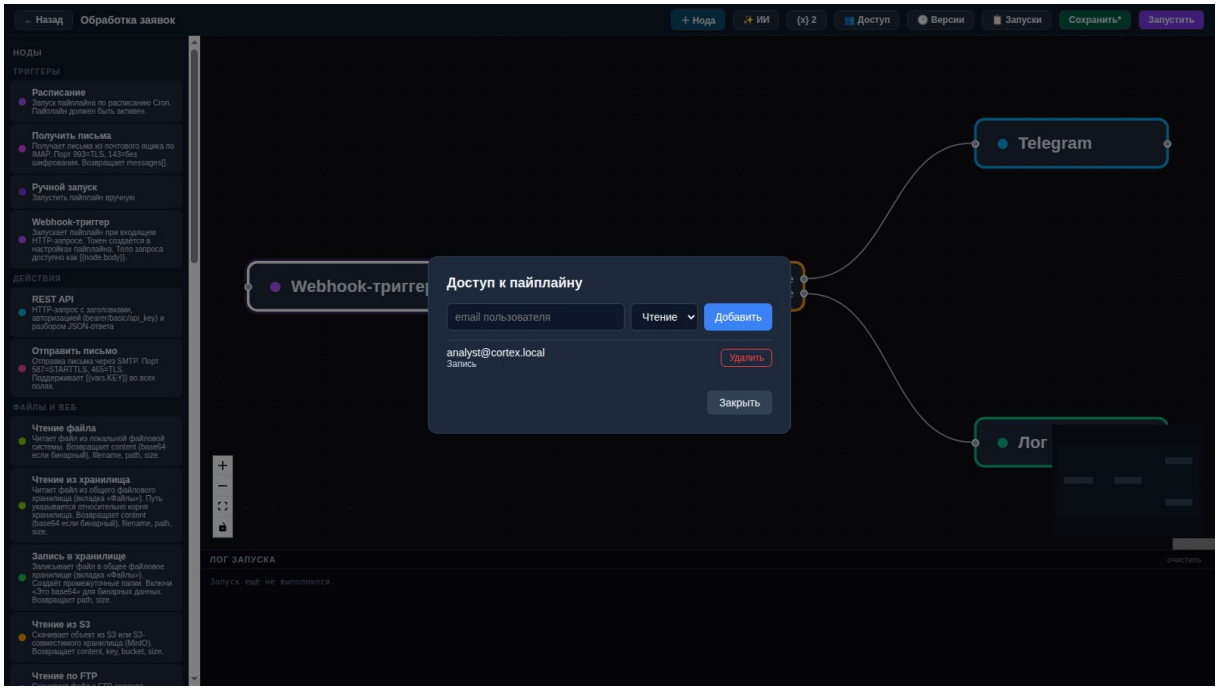


Рисунок 15 — Окно «Доступ к пайплайну»

В списке окна выводятся адреса пользователей с доступом и уровень доступа («Пока никому не предоставлен доступ», если список пуст); кнопка «Удалить» отзывает доступ. Под полем ввода выводятся сообщения об ошибках: «user not found» — пользователь не зарегистрирован, «cannot share with yourself» — указан собственный адрес.

Пользователь, получивший доступ, видит процесс в своём реестре и работает с ним в пределах уровня доступа; действия сверх уровня (например, сохранение или запуск с уровнем «Чтение») Система отклоняет.

10.2. Версии процесса

Версия процесса — сохранённый снимок графа (узлов, связей, настроек) и переменных процесса. Версия создаётся автоматически при каждом сохранении, изменяющем граф процесса, а также при восстановлении версии. Номера версий возрастают: v1, v2, v3 и т. д.

10.3. Просмотр истории версий

Кнопка «Версии» открывает окно «История версий» со списком версий от новых к старым: номер версии (vN), дата и время создания, адрес пользователя, сохранившего процесс. Для версии, созданной восстановлением, рядом с номером выводится оранжевая пометка «восстановлено из vN». Кнопка «Восстановить» отображается пользователям с уровнем доступа «владелец» или «запись». Если версий нет, выводится «Версий пока нет — они создаются при сохранении изменений графа».

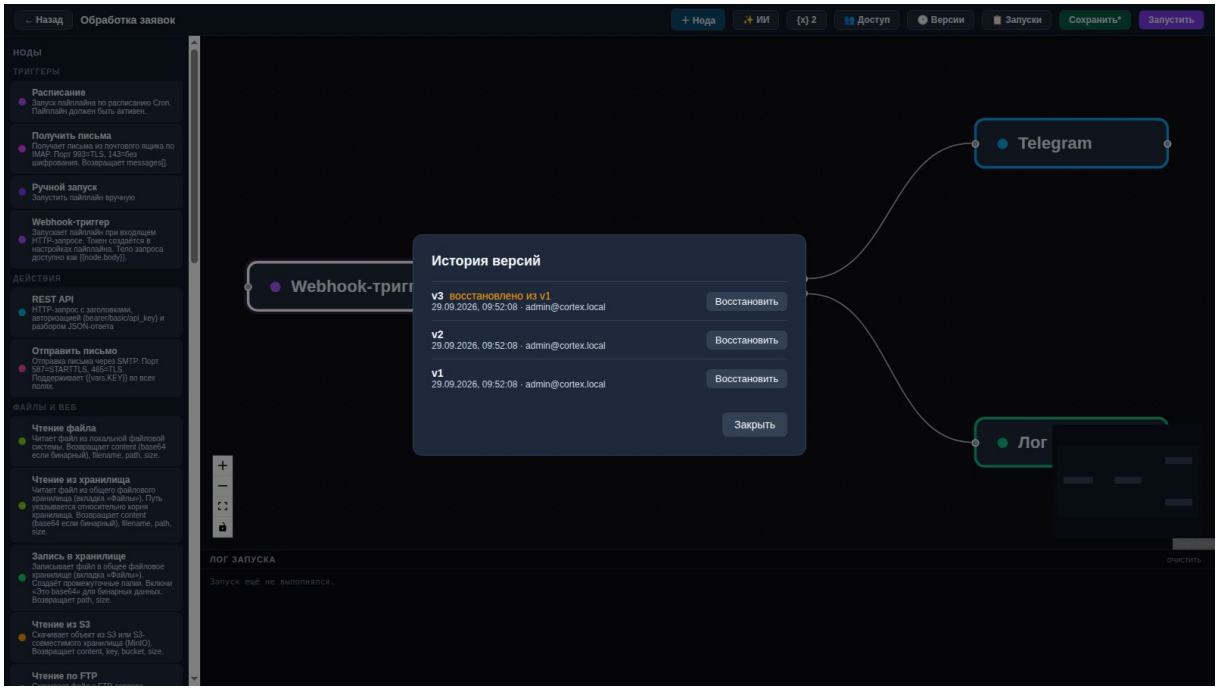


Рисунок 16 — Окно «История версий»

10.4. Восстановление версии

Нажмите «Восстановить» в строке нужной версии. Граф и переменные процесса заменяются содержимым выбранной версии, изменения сразу записываются в базу данных, на холсте отображается восстановленный процесс, окно закрывается. Восстановление создаёт новую версию с пометкой «восстановлено из vN», поэтому состояние процесса до восстановления остаётся в истории и может быть возвращено.

11. Файловое хранилище

11.1. Раздел «Файлы»

Раздел «Файлы» — общее файловое хранилище Системы: единое дерево папок и файлов на сервере, доступное всем пользователям через веб-интерфейс, по протоколу SFTP и из процессов (узлы «Чтение из хранилища» и «Запись в хранилище»). Раздел открывается кнопкой «Файлы» в реестре процессов.

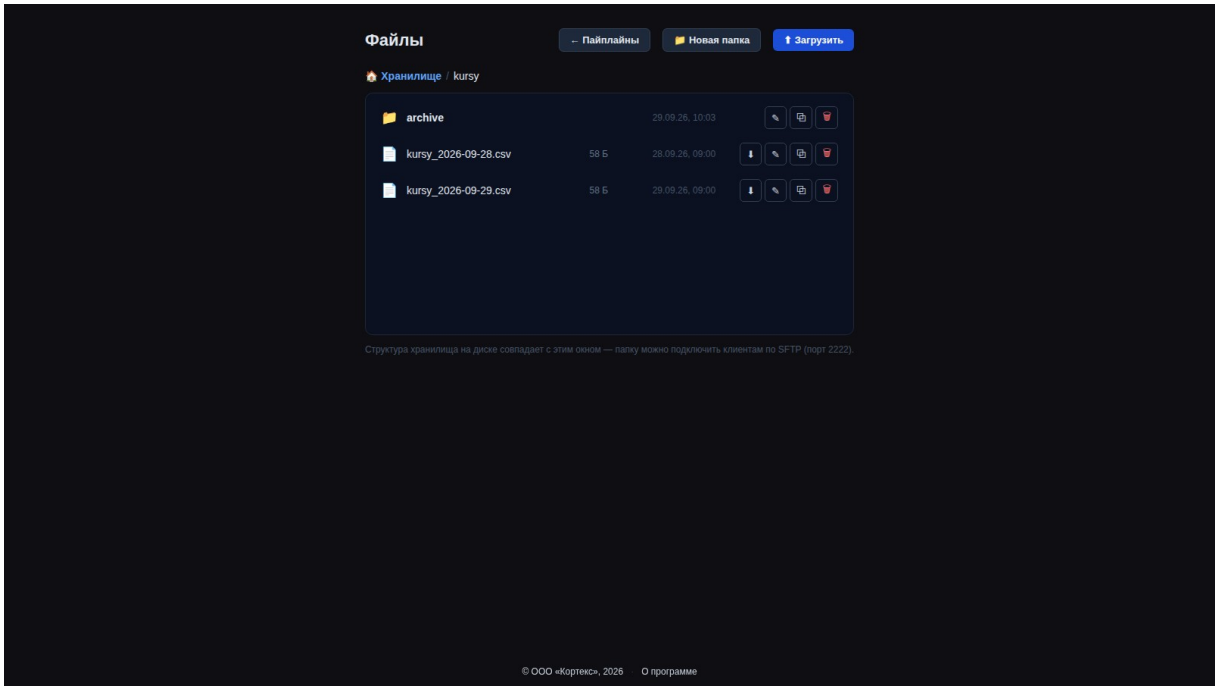


Рисунок 17 — Раздел «Файлы»

Элемент	Назначение
Новая папка	Создаёт папку в текущей папке (имя вводится в окне «Имя новой папки:»).
Загрузить	Загружает выбранные на компьютере файлы в текущую папку.
Строка пути	Путь от корня «Хранилище»; щелчок по элементу открывает папку. Двойной щелчок по строке папки в списке открывает её.
Скачать	Скачивает файл на компьютер (только для файлов).
Переименовать / Копировать	Изменяет имя (окно «Новое имя:») или создаёт копию в текущей папке (окно «Имя копии:», по умолчанию «<имя> (копия)»).
Удалить	Удаляет файл или папку со всем содержимым после подтверждения.

Кнопка «← Пайплайны» возвращает в реестр процессов. Для каждого файла выводятся размер и дата изменения. Если папка пуста, выводится сообщение «Папка пуста. Перетащите файлы сюда или нажмите «Загрузить».».

11.2. Загрузка и перемещение перетаскиванием

- чтобы загрузить файлы в текущую папку, перетащите их из окна файлового менеджера компьютера в область списка файлов — область подсвечивается;
- чтобы загрузить файлы в конкретную папку, отпустите их над строкой этой папки;
- чтобы переместить файл или папку внутри хранилища, перетащите строку файла на строку папки назначения.

Максимальный размер загружаемых за один раз файлов задаётся администратором при установке (по умолчанию 100 МБ) и отображается в разделе «Админ» → «Настройки» (параметр «Лимит загрузки (байт)»).

11.3. Использование хранилища в процессах

Узел «Запись в хранилище» сохраняет результат процесса в файл хранилища, узел «Чтение из хранилища» — читает файл хранилища для обработки. Пути в обоих узлах указываются относительно корня хранилища, например `/kursy/kursy.csv`. Файлы, записанные процессами, сразу появляются в разделе «Файлы» и доступны по протоколу SFTP; файлы, загруженные пользователями или внешними системами по SFTP, сразу доступны процессам.

11.4. Подключение по протоколу SFTP

Система содержит встроенный SFTP-сервер, открывающий то же дерево файлов, что и раздел «Файлы». Подключение выполняется любым SFTP-клиентом — графическим или консольным.

Параметры подключения: адрес сервера — тот же, что у веб-интерфейса (без https://); порт — 2222 (или иной, заданный при установке); имя пользователя — адрес электронной почты учётной записи Системы; пароль — пароль учётной записи. Корневая папка подключения — корень общего хранилища, выход за его пределы невозможен.

При первом подключении SFTP-клиент показывает отпечаток ключа сервера и запрашивает подтверждение; подтвердите подключение. Отпечаток сохраняется клиентом, при последующих подключениях запрос не выводится.

Пример подключения консольным клиентом sftp и работы с файлами:

```
sftp -P 2222 "ivanov@cortex.local"@<адрес сервера>
sftp> ls
kursy  zayavki
sftp> cd kursy
sftp> get kursy_2026-09-29.csv
sftp> put report.xlsx
sftp> bye
```

Все операции SFTP-клиента (просмотр, скачивание, загрузка, переименование, удаление, создание папок) выполняются в том же хранилище и сразу отражаются в разделе «Файлы».

12. ИИ-ассистент

12.1. Назначение и условия работы

ИИ-ассистент формирует новый процесс или дорабатывает открытый процесс по текстовому описанию задачи на русском языке. Ассистент использует сервис большой языковой модели (внешний или размещённый в инфраструктуре заказчика) с

совместимым программным интерфейсом и учитывает перечень узлов библиотеки Системы и их параметры.

ИИ-ассистент работает только после того, как администратор настроил подключение к сервису большой языковой модели (подраздел 13.7). Если подключение не настроено, на запрос выводится сообщение «Интеграция с ИИ не настроена. Откройте «Настройки ИИ» и укажите провайдера, модель и API-ключ.» и кнопка «Открыть «Настройки ИИ»».

12.2. Открытие панели ассистента

Панель ИИ-ассистента открывается в визуальном редакторе процесса кнопкой «ИИ» на панели инструментов или сочетанием клавиш Ctrl+I (на компьютерах Apple — ⌘+I). Повторное нажатие кнопки или сочетания клавиш закрывает панель. Панель располагается справа от холста.

В заголовке панели «ИИ-ассистент» расположены кнопки «⚙️» (открывает окно «Настройки ИИ», подраздел 13.7) и «×» (закрывает панель). Под заголовком — область сообщений: запросы пользователя и ответы ассистента; до первого запроса в ней выводится подсказка с примером описания задачи. Внизу — поле ввода с подсказкой «Опишите пайплайн… (Enter — отправить, Shift+Enter — перенос)» и кнопка «Отправить»; во время обработки запроса на кнопке выводится «Генерация…», в области сообщений — «ИИ думает…».

12.3. Описание задачи

Опишите задачу обычным текстом: откуда получить данные, как их обработать, куда передать результат и когда запускать процесс. Чем точнее описание (адреса, имена полей, условия, формат результата), тем точнее предложенный процесс. Пример: «Каждый день в 9 утра получай курсы валют с веб-сервиса, выбирай курсы доллара и евро и сохраняй их в файл CSV в папку kursy хранилища».

Если на холсте уже есть узлы, ассистент дорабатывает текущий процесс: в описании укажите, что нужно изменить, например «добавь отправку результата письмом на адрес otchet@cortex.local».

12.4. Предпросмотр и применение

Ответ ассистента содержит пояснение, схему предложенного процесса (предпросмотр графа), при необходимости — блок «⚠️ Предупреждения» с замечаниями (например, о параметрах, которые нужно заполнить вручную), кнопку «Применить» и число узлов предложенного процесса.

место для снимка экрана: В новом пустом процессе «Курсы валют (ИИ)» нажать Ctrl+I, ввести описание «Каждый день в 9 утра получай курсы валют с веб-сервиса и сохраняй курсы доллара и евро в CSV-файл в папку kursy хранилища», нажать «Отправить» и дождаться ответа. Видны запрос пользователя, пояснение ассистента, схема из 4–5 узлов, кнопка «Применить» и надпись «N нод».

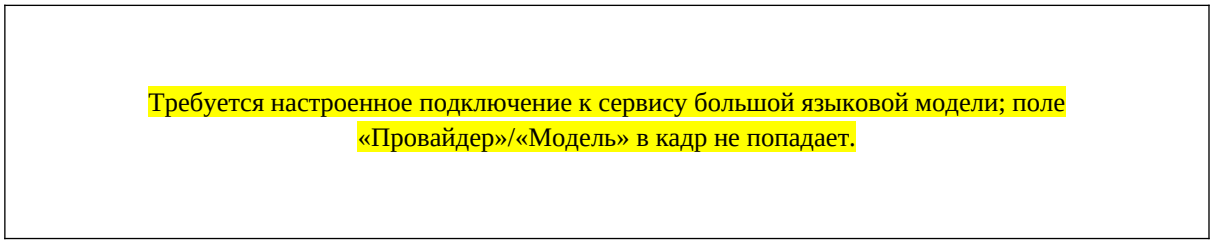


Рисунок 18 — Панель ИИ-ассистента с предложенным процессом

1. Просмотрите схему предложенного процесса и предупреждения.
2. Нажмите «Применить» — граф на холсте будет заменён предложенным, процесс отметится как изменённый («Сохранить*»).
3. Проверьте и при необходимости дополните параметры узлов (адреса, учётные данные, пути) в панели настроек узлов.
4. Сохраните процесс кнопкой «Сохранить*» (для отказа от изменений вернитесь в реестр кнопкой «← Назад» без сохранения).

Если предложенный процесс не подходит, уточните описание и отправьте новый запрос. Если корректный граф получить не удалось, выводится «ИИ не смог сгенерировать валидный граф: …», при недоступности сервиса — «Ошибка вызова ИИ: …».

13. Администрирование

13.1. Раздел «Управление системой»

Раздел открывается кнопкой «Админ» в реестре процессов и доступен только пользователям с ролью «Администратор». Раздел содержит заголовок «Управление системой», кнопку «← Пайплайны» и вкладки «Пользователи», «Настройки», «Журнал», «Диагностика». Кроме того, администратор управляет глобальными константами (кнопка «Константы») и настройками ИИ-ассистента (кнопка «Настройки ИИ») в реестре процессов.

13.2. Пользователи

Вкладка «Пользователи» содержит список учётных записей Системы.

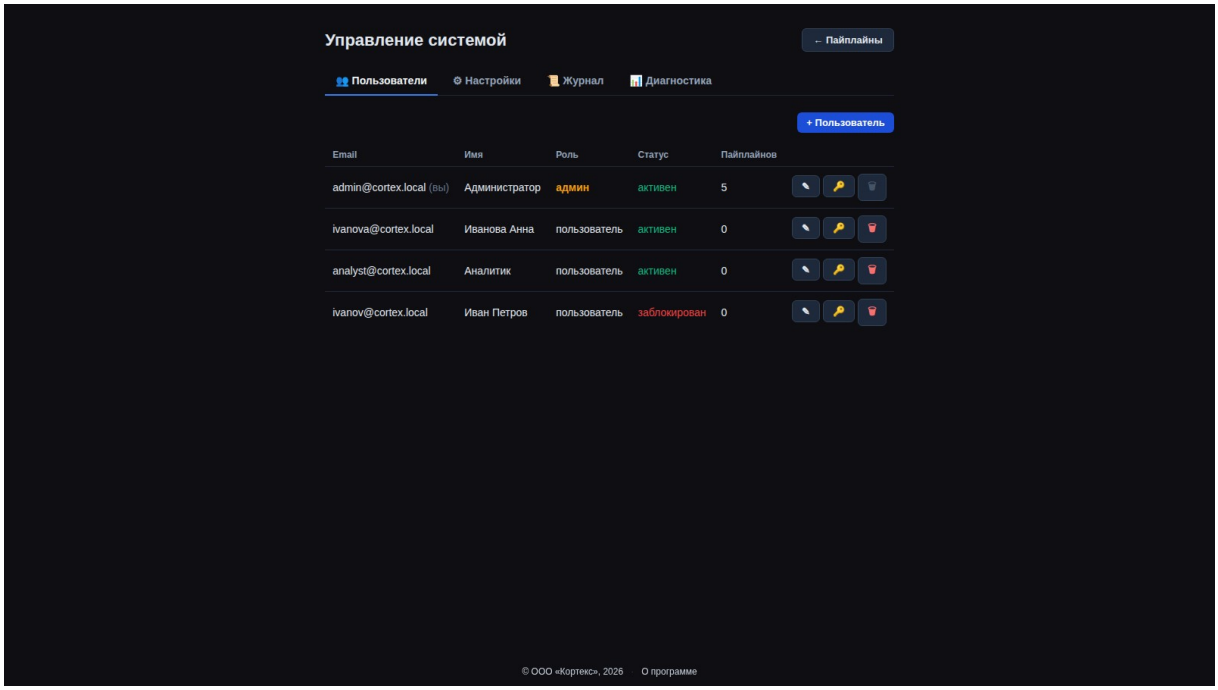


Рисунок 19 — Вкладка «Пользователи»

Столбцы таблицы: «Email» (строка текущего администратора помечена «(вы)»), «Имя», «Роль» («админ» или «пользователь»), «Статус» («активен» или «заблокирован»), «Пайплайнов» — число процессов, владельцем которых является пользователь. Кнопки в строке: с изображением карандаша — редактирование, с изображением ключа (подсказка «Сбросить пароль») — сброс пароля, с изображением корзины — удаление (для собственной учётной записи недоступна).

Создание пользователя: нажмите «+ Пользователь», в окне «Новый пользователь» заполните «Email», «Имя», «Пароль» (не менее 6 символов), при необходимости установите флажок «Администратор» и нажмите «Сохранить». Если адрес не заполнен или пароль короче 6 символов, выводится «Email и пароль (мин. 6) обязательны».

Изменение пользователя: нажмите кнопку редактирования, в окне «Редактировать» измените «Email», «Имя», флажки «Администратор» (назначение и снятие роли) и «Активен» (снятие флажка блокирует учётную запись — вход заблокированного пользователя отклоняется) и нажмите «Сохранить». Собственную роль и статус администратор изменить не может — выводится подсказка «Нельзя изменить свою роль или статус.».

Сброс пароля: нажмите кнопку «Сбросить пароль», введите новый пароль в окне «Новый пароль для <email> (мин. 6 символов):» и нажмите «ОК». При успешной смене выводится «Пароль обновлён», при слишком коротком пароле — «Минимум 6 символов».

Удаление пользователя: нажмите кнопку удаления и подтвердите действие в окне «Удалить пользователя <email>? Его пайплайны перейдут к вам.». Процессы удаляемого

пользователя передаются администратору, выполнившему удаление. Удалить единственного активного администратора нельзя.

Все операции с пользователями регистрируются в журнале событий.

13.3. Настройки

Вкладка «Настройки» содержит два блока: «Редактируемые параметры» и «Конфигурация (только чтение)».

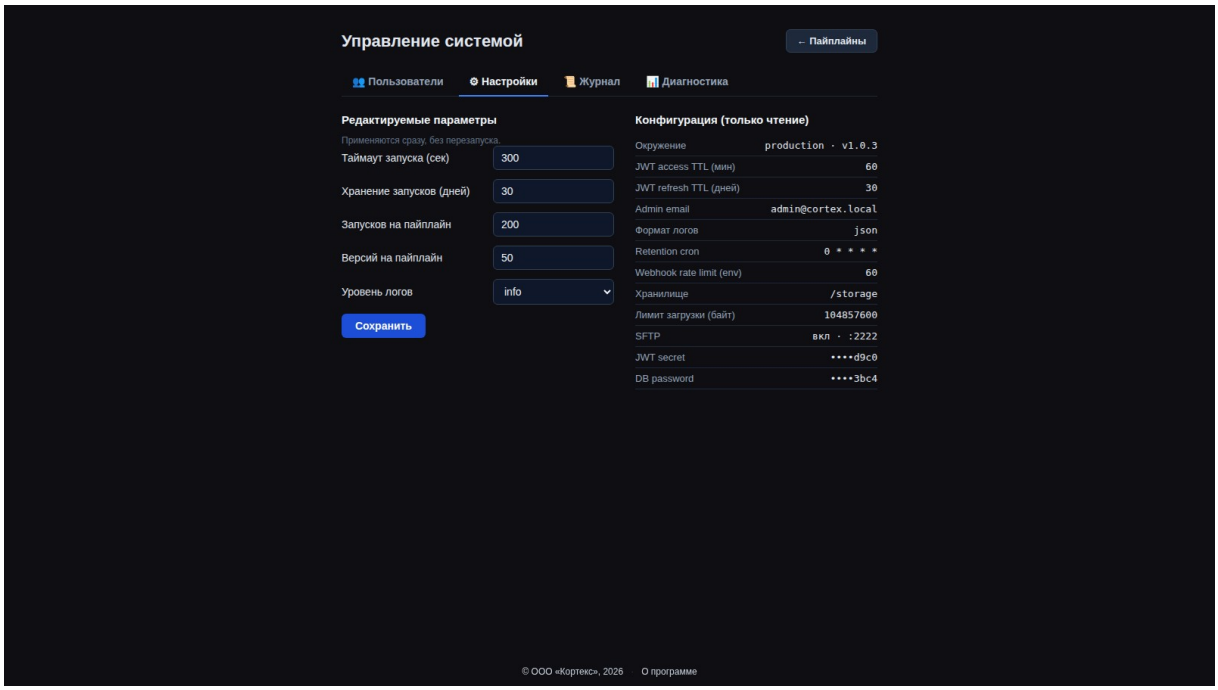


Рисунок 20 — Вкладка «Настройки»

Редактируемые параметры применяются сразу после нажатия «Сохранить», без перезапуска Системы; при успешном сохранении выводится «Сохранено».

Параметр	**Назначение**
Таймаут запуска (сек)	Предельная продолжительность одного запуска процесса. По умолчанию 300.
Хранение запусков (дней) / Запусков на пайплайн / Версий на пайплайн	Срок хранения истории запусков и число хранимых запусков и версий каждого процесса; сверх лимита удаляются старейшие; 0 — без ограничения.
Уровень логов	Детализация технического журнала серверной части: debug, info, warn, error.

Блок «Конфигурация (только чтение)» отображает параметры, заданные при установке (окружение и версия, время действия ключей доступа, формат логов, расписание очистки, ограничение частоты вебхуков, путь и лимит загрузки хранилища, состояние SFTP); секретные значения замаскированы.

13.4. Журнал

Вкладка «Журнал» отображает журнал событий — действия пользователей, значимые для безопасности и администрирования. Выводятся 300 последних записей, журнал обновляется автоматически каждые 10 секунд.

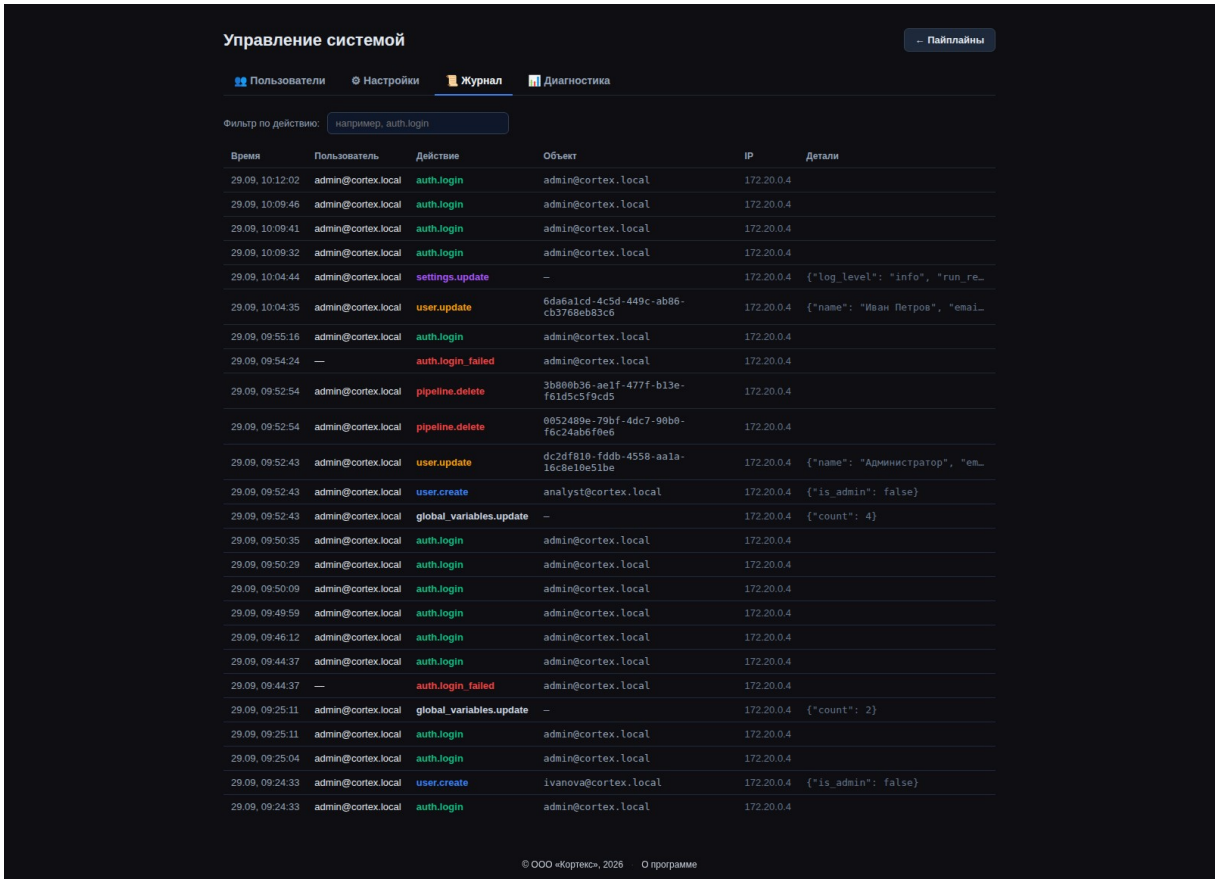


Рисунок 21 — Вкладка «Журнал»

Столбцы таблицы: «Время», «Пользователь» (адрес электронной почты), «Действие» (код действия, цвет соответствует характеру события), «Объект» (адрес или идентификатор пользователя либо процесса), «IP» (адрес, с которого выполнено действие), «Детали» (сведения в формате JSON). Поле «Фильтр по действию:» отбирает записи по полному коду действия, например «auth.login».

Код действия	Событие
auth.login / auth.login_failed / auth.login_blocked	Успешный вход / неверный адрес или пароль / попытка входа заблокированного пользователя.
user.create / user.update / user.reset_password / user.delete	Создание, изменение, сброс пароля, удаление пользователя.
settings.update / global_variables.update / ai_settings.update	Изменение системных настроек / глобальных констант / настроек ИИ-ассистента.
pipeline.delete	Удаление процесса.

13.5. Диагностика

Вкладка «Диагностика» отображает состояние компонентов Системы. Данные обновляются автоматически каждые 5 секунд и по кнопке «⟳ Обновить».

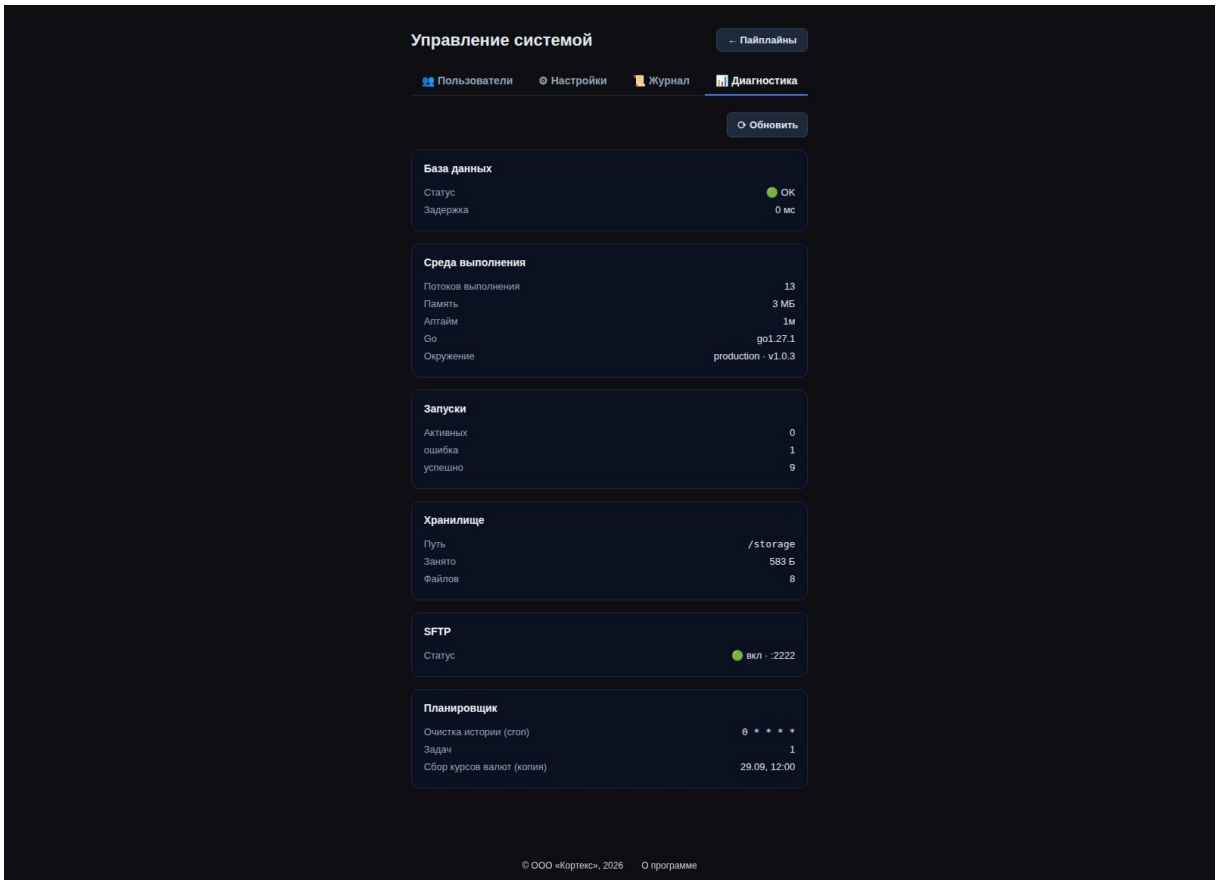


Рисунок 22 — Вкладка «Диагностика»

Карточки вкладки: «База данных» — статус подключения к СУБД («OK» с зелёным индикатором или текст ошибки с красным) и задержка ответа; «Среда выполнения» — число параллельных потоков выполнения (Goroutines), занятая память, время непрерывной работы (Аптайм), версия среды исполнения, окружение и версия Системы; «Запуски» — число выполняющихся запусков (Активных) и число запусков в каждом статусе; «Хранилище» — путь к каталогу хранилища, занятый объём и число файлов; «SFTP» — состояние встроенного SFTP-сервера и порт; «Планировщик» — расписание очистки истории (Retention cron), число заданий расписания и время ближайшего запуска для первых шести заданий.

Для внешнего контроля работоспособности серверная часть предоставляет адрес проверки состояния `/health` (ответ `{"status":"ok"}`) и адрес `/metrics` с метриками в формате Prometheus (число и длительность запросов, число выполненных узлов по типам и статусам). Адреса предназначены для систем мониторинга и настраиваются администратором сервера.

13.6. Глобальные константы

Глобальные константы — именованные значения, доступные во всех процессах записью `{{vars.ИМЯ}}`: общие адреса сервисов, учётные данные, параметры. Окно «Глобальные константы» открывается кнопкой «Константы» в реестре процессов. Просматривать константы могут все пользователи, изменять — только администратор. Если переменная процесса имеет то же имя, в этом процессе используется значение переменной процесса.

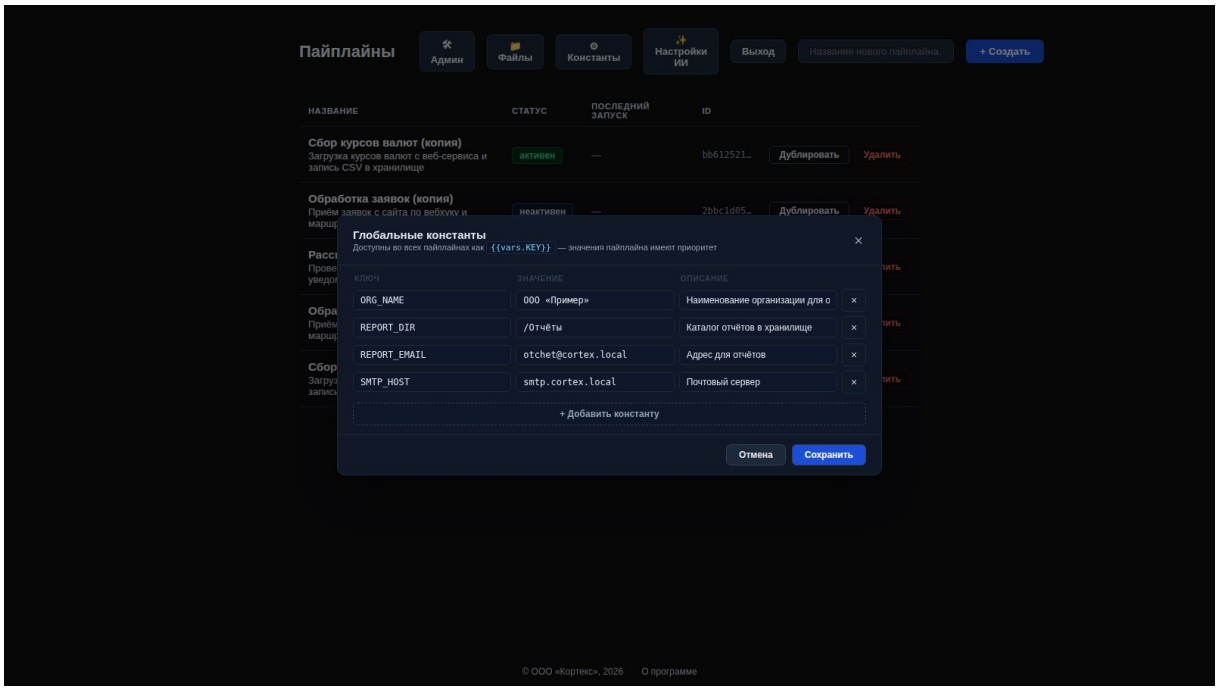


Рисунок 23 — Окно «Глобальные константы»

Для каждой константы задаются «Ключ» (например, `SMTP_HOST`), «Значение» и необязательное «Описание». Кнопка «+ Добавить константу» добавляет строку, «×» удаляет её, строки с пустым ключом отбрасываются. «Сохранить» записывает константы (изменение регистрируется в журнале событий), «Отмена» закрывает окно без изменений.

13.7. Настройки ИИ-ассистента

Окно «Настройки ИИ» открывается администратором кнопкой «Настройки ИИ» в реестре процессов или кнопкой «⚙️» панели ИИ-ассистента. Настройки действуют для всех процессов и всех пользователей (подпись окна «Глобально для всех пайплайнов — используется ИИ-ассистентом»). Сохранять настройки может только администратор.

Поле / элемент	Назначение
Провайдер	Тип программного интерфейса сервиса большой языковой модели (из списка). Для сервиса в инфраструктуре заказчика без ключа доступа выбирается соответствующий тип — поле «API-ключ» скрывается, в поле адреса указывается адрес сервиса.

Поле / элемент	Назначение
Модель	Имя модели, предоставляемой сервисом.
API-ключ	Ключ доступа к сервису. Сохранённый ключ не отображается: в поле выводится его замаскированный вид; чтобы не менять ключ, оставьте поле пустым. Флажок «Удалить сохранённый ключ» удаляет ключ.
API URL (необязательно)	Адрес программного интерфейса сервиса; если не заполнен, используется адрес по умолчанию для выбранного типа интерфейса.
Дополнительно	Раскрывает поля «Температура» (вариативность ответов, по умолчанию 0.2), «Макс. токенов» (предельный объём ответа, по умолчанию 4000) и «Системный промпт (переопределение)» (собственная инструкция для модели вместо формируемой автоматически по библиотеке узлов).
Сохранить / Отмена	Сохраняет настройки или закрывает окно без изменений. Изменение регистрируется в журнале событий.

14. Выгрузки и выходные данные

Тип	Описание
Файлы хранилища	Результаты процессов, записанные узлом «Запись в хранилище» (CSV, JSON, текст, двоичные файлы); скачиваются кнопкой «Скачать» в разделе «Файлы» или по протоколу SFTP.
Выходные данные узла	Блок «Последний вывод» панели настроек узла — результат текущего или выбранного в истории запуска.
Журнал запуска	Хронология выполнения узлов, сообщения и дампы данных узлов «Лог».
Результат синхронного запуска	Выходные данные всех узлов в формате JSON в ответе программного интерфейса (подраздел 8.1).
Сообщения и письма	Результаты, отправляемые узлами «Отправить письмо» и «Telegram».
Журнал событий и диагностика	Административные сведения о действиях пользователей и состоянии Системы (раздел «Админ»).

Чтобы получить результат процесса в виде файла, завершите процесс узлом «Запись в хранилище»: сформируйте содержимое узлом «Шаблон» (текст, CSV) или передайте результат узла в формате JSON, укажите путь в хранилище и после запуска скачайте файл в разделе «Файлы».

15. Типовые сценарии

Сценарии выполняются на демонстрационных процессах «Сбор курсов валют» и «Обработка заявок». Демонстрационные данные не содержат персональных сведений.

15.1. Получить данные веб-сервиса, разобрать JSON и записать CSV в хранилище

1. В разделе «Пайплайны» введите название «Сбор курсов валют» и нажмите «+ Создать».
2. Перетащите из палитры на холст узлы «Ручной запуск» (категория «Триггеры»), «REST API» («Действия»), «Разбор JSON» («Данные»), «Шаблон» («Данные») и «Запись в хранилище» («Файлы и веб»). Соедините их последовательно: выход каждого узла — со входом следующего.
3. Дважды щёлкните узел «REST API». В поле «URL» укажите адрес веб-сервиса, возвращающего курсы валют в формате JSON (например, `https://www.cbr-xml-daily.ru/daily_json.js`), в поле «Метод» выберите GET. Закройте панель.
4. Откройте узел «Разбор JSON». В поле «JSON-содержимое» введите `{{` и выберите в подсказках `body` узла «REST API» — получится `{{rest_api.body}}`. В поле «Path (необязательно)» введите `.Valute`.
5. Откройте узел «Шаблон» и введите в поле «Шаблон» текст файла CSV:

```
Валюта;Курс
USD;{{parse_json.data.USD.Value}}
EUR;{{parse_json.data.EUR.Value}}
CNY;{{parse_json.data.CNY.Value}}
```

6. Откройте узел «Запись в хранилище». В поле «Путь в хранилище» введите `/kursy/kursy.csv`, в поле «Содержимое» — `{{template.text}}`.
7. Нажмите «Сохранить*», затем «Запустить». Дождитесь записи «Запуск завершён успешно» в панели «Лог запуска».
8. Откройте узел «Разбор JSON» и проверьте в блоке «Последний вывод» разобранные курсы.
9. Перейдите в раздел «Файлы», откройте папку «kursy» и скачайте файл kursy.csv кнопкой «Скачать».

15.2. Настроить запуск процесса по расписанию

1. В разделе «Пайплайны» нажмите «Дублировать» в строке процесса «Сбор курсов валют» и откройте созданную копию «Сбор курсов валют (копия)».
2. Выделите узел «Ручной запуск» и удалите его клавишей Delete.
3. Добавьте узел «Расписание» и соедините его выход со входом узла «REST API».
4. Откройте узел «Расписание», в поле «Cron-выражение» введите `0 9 * * 1-5`, в поле «Описание» — «По будням в 9:00».

5. Чтобы каждый запуск сохранял отдельный файл, в узле «Запись в хранилище» замените путь на `/kursy/kursy_{{cron_trigger.triggered_at}}.csv`.
6. Нажмите «Сохранить*».
7. В реестре процессов щёлкните по отметке «неактивен» в строке процесса — в столбце «Статус» появится «активен» (подраздел 9.2).
8. Администратор может проверить задание в разделе «Админ» → «Диагностика», карточка «Планировщик»: указано время следующего запуска. После срабатывания расписания запуск появится в окне «История запусков» процесса, а файл — в папке «kursy» раздела «Файлы».

15.3. Создать процесс, запускаемый по вебхуку

1. Создайте процесс «Обработка заявок».
2. Добавьте узлы «Webhook-триггер», «If», «Telegram» и «Лог». Соедините «Webhook-триггер» → «If»; выход «true» узла «If» — с узлом «Telegram», выход «false» — с узлом «Лог».
3. В узле «If» в поле «Выражение» введите `webhook_trigger.priority == "high"`.
4. В узле «Telegram» заполните «Bot Token» и «Chat ID» (рекомендуется через переменные процесса `{{vars.TG_TOKEN}}`, `{{vars.TG_CHAT}}`) и «Текст сообщения»: `Срочная заявка №{{webhook_trigger.number}}: {{webhook_trigger.text}}`.
5. В узле «Лог» в поле «Сообщение» введите `Заявка № {{webhook_trigger.number}} принята в обычном порядке`.
6. Сохраните процесс, откройте узел «Webhook-триггер», в блоке «Webhook URL» нажмите «+ Новый токен» и скопируйте адрес кнопкой «copy».
7. Отправьте тестовый запрос POST на скопированный адрес с телом `{"number": 1024, "priority": "high", "text": "Не работает принтер в каб. 12"}` (подраздел 9.1) либо передайте адрес администратору внешней системы.
8. Нажмите «Запуски» и выберите запуск — на холсте отобразятся выполненные узлы; в узле «Webhook-триггер» в блоке «Последний вывод» видны переданные поля number, priority, text.

15.4. Обработать список заявок циклом For Each

1. Создайте процесс «Обработка заявок (список)» и добавьте узлы «Ручной запуск», «Задать переменную», «For Each», «Шаблон», «Лог». Соедините их последовательно.
2. В узле «Задать переменную» в поле «Данные (JSON-объект)» введите:

```
{"requests": [
  {"number": 1021, "text": "Замена картриджа"},
  {"number": 1022, "text": "Доступ к общей папке"},
  {"number": 1023, "text": "Настройка почты"}
```

```
]}
```

3. В узле «For Each» в поле «Массив элементов (JSON)» введите `{{set_variable.requests}}`; в поле «Параллельность (1 = последовательно)» оставьте 1 (для одновременной обработки нескольких элементов укажите большее число).
4. В узле «Шаблон» введите `Заявка №{{for_each.item.number}} ({{for_each.index}}): {{for_each.item.text}}`.
5. В узле «Лог» в поле «Сообщение» введите `{{template.text}}`.
6. Сохраните и запустите процесс. В панели «Лог запуска» для каждой заявки выводятся записи «итерация 1 / 3», «итерация 2 / 3», «итерация 3 / 3» и сформированные сообщения.

Для обработки реальных данных вместо узла «Задать переменную» используйте результат другого узла, возвращающий массив: `{{sql.rows}}`, `{{parse_csv.rows}}`, `{{imap_receive.messages}}`, `{{scrape_html.items}}`.

15.5. Предоставить доступ коллеге и восстановить версию

1. Откройте процесс «Обработка заявок» под учётной записью владельца и нажмите «Доступ».
2. Введите адрес коллеги (например, analyst@cortex.local), выберите «Запись» и нажмите «Добавить». Закройте окно.
3. Коллега входит в Систему под своей учётной записью, видит процесс «Обработка заявок» в реестре, изменяет, например, текст сообщения узла «Лог» и сохраняет процесс — создаётся новая версия.
4. Чтобы вернуть предыдущее состояние, владелец (или коллега с уровнем «Запись») открывает процесс и нажимает «Версии».
5. В окне «История версий» по дате, времени и адресу автора определите версию до изменения и нажмите «Восстановить».
6. Процесс на холсте примет вид выбранной версии, в истории появится новая версия с пометкой «восстановлено из vN».
7. Чтобы прекратить доступ коллеги, откройте «Доступ» и нажмите «Удалить» в строке его адреса.