

Пользовательская документация

Mimic



Mimic - это современный инструмент для аналитиков, разработчиков и QA-инженеров, который позволяет быстро и удобно имитировать работу систем или сервисов.

С помощью Mimic вы можете воспроизводить поведение реальных систем, отдельных сервисов или отдельных методов с возможностью проксирования всех остальных запросов, что ускоряет разработку, облегчает тестирование и снижает риски на этапе внедрения.

Возможности

Система предоставляет следующие возможности:

- **Пользовательский интерфейс** - многопользовательский веб-интерфейс с поддержкой авторизации и настройки ролей;
- **HTTP/HTTPS** - имитация взаимодействия с внешними сервисами через HTTP/HTTPS-запросы;
- **gRPC** - симуляция сервисов, использующих протокол gRPC;
- **gRPC-контракты** - поддержка загрузки gRPC-контрактов: автоматическое извлечение описаний сервисов через рефлексию или импорт .proto-файлов;
- **Топики Kafka** - эмуляция поведения сторонних сервисов через взаимодействие с реальными Kafka-топиками, включая - отправку и получение сообщений в существующей Kafka-инфраструктуре;
- **Комбинирование** - возможность комбинировать различные типы симуляций (HTTP, gRPC, Kafka) в рамках одного проекта;
- **Логика ответов** - поддержка условной логики: возврат различных ответов на основе содержимого запроса (параметров, заголовков, тела и т.д.);

- **Проксирирование** - частичное или полное проксирирование запросов к реальным сервисам;
- **Журнал событий** - встроенный журнал событий для удобного мониторинга и отладки;

Основные сценарии использования Mimic

1. Разработка без зависимых сервисов.

- **Проблема:** Команда разрабатывает сервис **A**, который зависит от сервиса **B**, но тот ещё не готов, работает нестабильно или отвечает слишком медленно.
- **Решение:** С помощью Mimic разработчики создают имитацию сервиса **B**, симуляцию его через HTTP/HTTPS, gRPC или Kafka. Благодаря поддержке условной логики ответов можно моделировать разные сценарии: успешные ответы, ошибки, таймауты. При этом уже реализованные методы сервиса **B** могут проксирироваться напрямую - без полной замены его функциональности.
- **Результат:** Разработка и локальное тестирование сервиса **A** продолжают без блокировок, даже если сервис **B** частично или полностью недоступен.

2. Надёжное тестирование сложных сценариев (включая отказы)

- **Проблема:** QA-инженеры не могут протестировать поведение системы при сбоях внешних зависимостей - например, при HTTP-ошибке 500, недоступности gRPC-сервиса или проблемах с сообщениями в Kafka (отсутствие ответа, некорректный формат, задержка).
- **Решение:** В Mimic настраивается динамический мок, который возвращает нужный ответ в зависимости от заголовков, URL-пути, тела запроса или других атрибутов. Для Kafka - можно эмулировать отправку битых сообщений, искусственные задержки или полное отсутствие ответа в топике.
- **Результат:** Удаётся покрыть тестами отказоустойчивые и крайние сценарии, что значительно повышает надёжность и предсказуемость системы в production-условиях.

3. Интеграционное тестирование гибридных архитектур

- **Проблема:** Система использует гибридную архитектуру, в которой одновременно задействованы REST, gRPC и Kafka. Для сквозного тестирования необходимо взаимодействие между сервисами, но часть из них

уже развернута в staging-среде, а другая - ещё в разработке или работает нестабильно.

- **Решение:** Mimic позволяет объединить все три типа симуляций (HTTP/HTTPS, gRPC, Kafka) в рамках одного проекта. При этом:
 - Kafka-топики используются как реальные - система может как получать события из них, так и отправлять мокированные ответы с учётом заданной логики;
 - REST- и gRPC-эндпоинты имитируются с поддержкой условной логики (например, разные ответы в зависимости от тела запроса или заголовков);
 - Рабочие методы проксируются напрямую к соответствующим staging-сервисам, минуя мок.
- **Результат:** Команда может проводить реалистичное сквозное тестирование даже при неполной готовности зависимостей - без настройки сложных тестовых окружений или ожидания завершения всех компонентов.

4. Совместная работа над контрактами и моками

- **Проблема:** Аналитики, разработчики и QA-команды используют разрозненные моки или локальные заглушки, что приводит к рассогласованности, багам на стыках и постоянному уточнению «а как на самом деле должен отвечать сервис?»
- **Решение:** Mimic предоставляет многопользовательский веб-интерфейс с поддержкой ролей и совместной работы. Аналитик или разработчик могут загрузить .proto-файл (вручную или через рефлексию) и настроить HTTP-моки под бизнес-сценарии, а QA - добавить тестовые кейсы с ошибками, таймаутами и проверками валидации. Все участники работают с единой согласованной версией моков, доступной по постоянной ссылке.
- **Результат:** Исчезает путаница в контрактах - команда получает единую «истину» о поведении зависимостей, сокращаются ошибки на этапе интеграции и ускоряется общий цикл разработки.

5. Демонстрация и демо-среды без внешних зависимостей

- **Проблема:** Нужно показать продукт клиенту, инвестору или на хакатоне, но подключение к реальным внешним API (платёжные системы, CRM, внешние сервисы) невозможно, небезопасно или нарушает условия использования.
- **Решение:** В Mimic настраивается полный набор моков, включая HTTP-эндпоинты, gRPC-вызовы и взаимодействие с Kafka-топиками. Благодаря

встроенному журналу событий, спикер может в реальном времени демонстрировать, как система обрабатывает действия пользователя - от запроса до реакции на событие.

- **Результат:** Создаётся стабильная, предсказуемая и полностью автономная демо-среда, не зависящая от сторонних систем, сетевых условий или ограничений внешних API.

6. Быстрая отладка и аудит поведения моков

- **Проблема:** Разработчики и QA тратят время на настройку внешнего логирования (например, через Kibana или файлы), чтобы понять, какие запросы пришли в мок, какие сценарии сработали и почему был возвращён тот или иной ответ. Часто логи неполные, разрозненные или требуют парсинга.
- **Решение:** Mimic предоставляет встроенный журнал событий, в котором по умолчанию отображаются все входящие и исходящие вызовы - включая тело, заголовки, параметры и сработавший сценарий имитации. Никакой дополнительной настройки не требуется: всё видно сразу в веб-интерфейсе.
- **Результат:** Инженеры мгновенно понимают, как мок обработал запрос, быстро находят ошибки в логике и подтверждают корректность поведения — без переключения между системами или анализа сырых логов.

Требования к инфраструктуре

Ресурсы	Минимальный	Рекомендуемый
vCPU	1	3
RAM	4 Гб	8 Гб
Системный диск	не менее 10 Гб	20 Гб
Диск с данными	не менее 3 Гб	10 Гб

Структура системы Mimic

Система Mimic поддерживает два режима развёртывания, адаптированных под потребности различных команд: **1. Единая система (Docker Compose)**

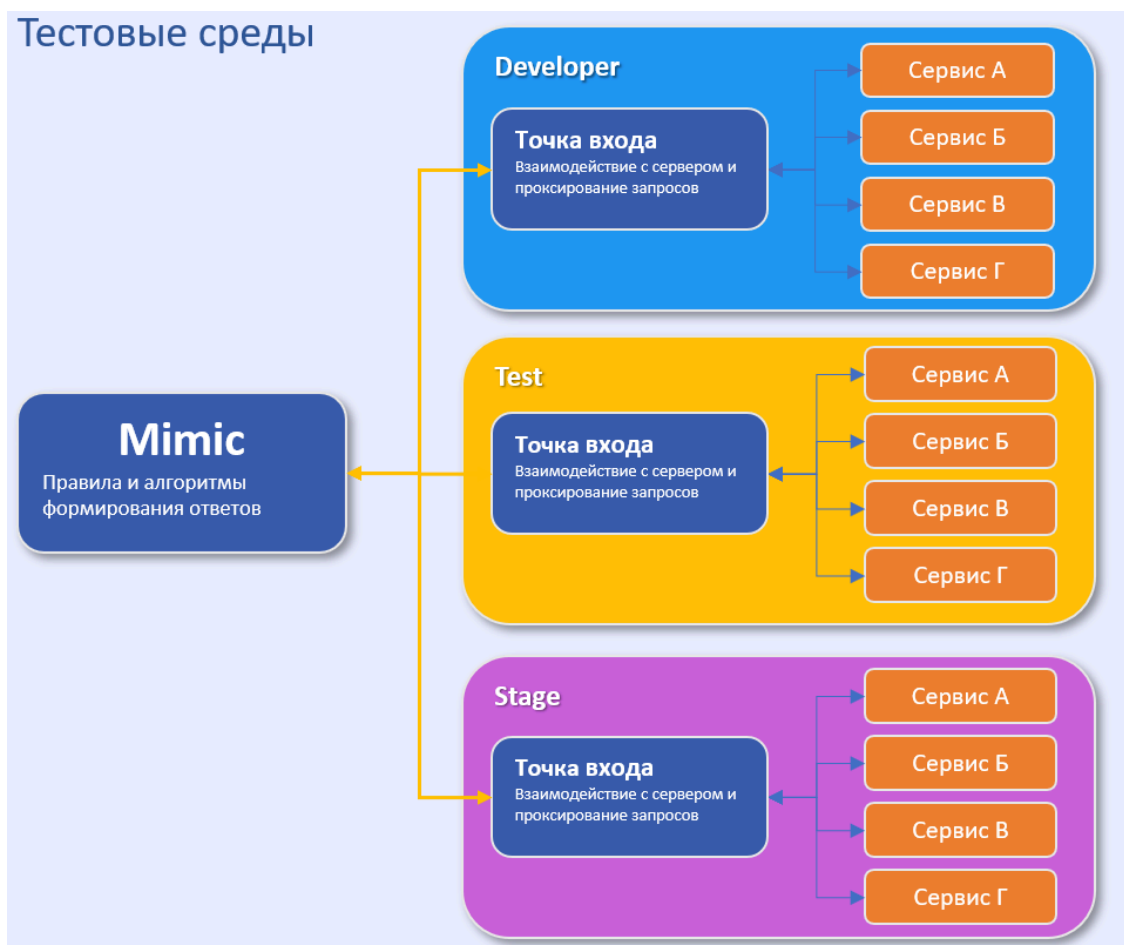
Рекомендуется для небольших команд, локальной разработки и демо-сред. Все компоненты — ядро Mimic, база данных, веб-интерфейс и точки входа —

запускаются в одном контейнерном окружении через `docker-compose.yml`.

Преимущества:

- Минимальные требования к инфраструктуре,
- Быстрое развёртывание «из коробки»,
- Удобство для отладки и обучения.

2. Распределённая архитектура (много-контурная) Разворачивается в изолированных контурах (например, Kubernetes-кластерах). Подходит для крупных команд с повышенными требованиями к масштабируемости, безопасности и отказоустойчивости.



Ключевые особенности:

- Ядро Mimic размещается централизованно (вне контуров взаимодействия)
- В каждом изолированном контуре (например, Developer, Test, Stage) разворачивается локальная точка входа прокси/адаптер, который:
 - принимает запросы от сервисов
 - взаимодействует с центральным ядром Mimic
 - возвращает сформированный ответ (мок или прокси)

Преимущества: - позволяет изолировать среды разработки, тестирования и приёмки - минимизировать сетевые задержки внутри контура - обеспечить отказоустойчивость и масштабируемость ядра

Принцип работы

Для каждого проекта можно определить собственный набор маршрутов (endpoint/method). В рамках маршрута настраиваются сценарии, которые определяют, как будет формироваться ответ на входящий запрос.

Когда Mimic получает запрос, он:

- Определяет, к какому проекту и контуру (стенду) относится запрос
- Ищет подходящий Endpoint и применяет логику обработки

Обработка запроса

- Если в Mimic найден маршрут (endpoint) со сценариями: Система последовательно проверяет сценарии в порядке приоритета до первого подходящего
 - Каждый сценарий может содержать фильтры (по заголовкам, параметрам URL, телу запроса и др.)
 - Для срабатывания сценария все его фильтры должны совпасть с данными входящего запроса (логическое И)
 - При совпадении:
 - извлекаются переменные (если заданы),
 - формируется ответ на основе шаблона (мока)
 - Сценарии без фильтров имеют самый низкий приоритет и считаются сценариями по умолчанию — они подходят всегда, если ни один другой сценарий не сработал
- Если ни один сценарий не подошёл: Mimic проверяет настройки проксирования для текущего контура:
 - Если проксирование включено — запрос перенаправляется на целевой URL реального сервиса, указанный в настройках проекта для этого контура
 - Если проксирование отключено — возвращается ошибка **404 Not Found**

Пример ответа при отсутствии подходящего сценария и отключённом проксировании:

```
{
  "code": 5,
  "message": "Mimic не может сгенерировать ответ или проксировать
    запрос, проверьте настройки",
  "details": [
    "В проекте {{Id}} для маршрута {{Endpoint}} не найдено ни
    одного подходящего сценария для мокирования и проксирования"
  ]
}
```

Примечание: Контур (или стенд) — это изолированная среда развертывания (например: dev, test, stage, prod; по умолчанию - Main). Для каждого контура можно задать отдельный целевой адрес для проксирования. Наличие и количество контуров зависит от инфраструктуры, в которой развёрнут Mimic

Стандартный процесс создания имитации

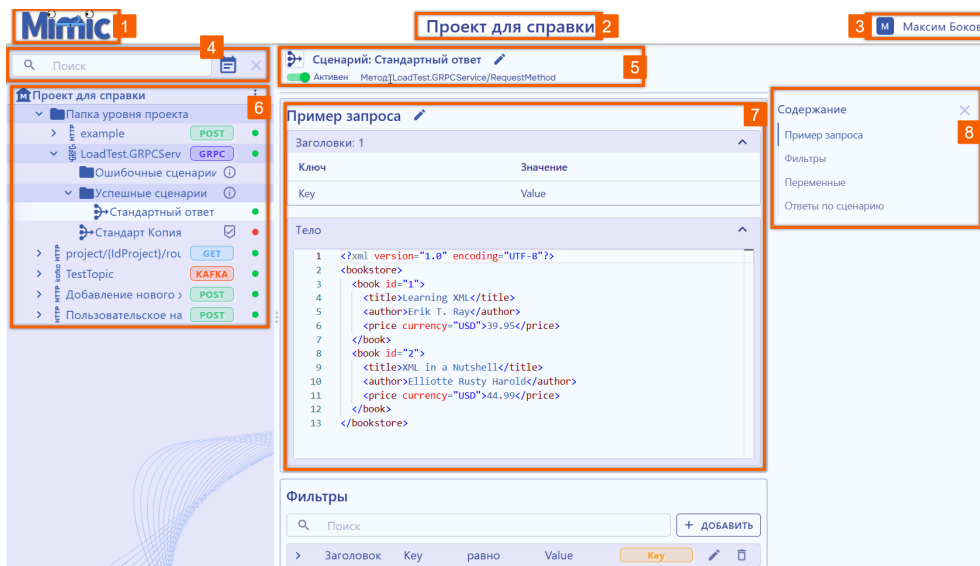
- Создание проекта
- Добавление маршрута
- Создание сценария
- Формирование ответа

Интерфейс системы

Все основные рабочие страницы — Настройки проекта, Endpoint, Сценарий — построены по единому шаблону, обеспечивающему:

- единообразный пользовательский опыт (UX)
- интуитивную навигацию
- гибкость при работе с большими проектами

Ниже — изображение с пронумерованными компонентами интерфейса и их детальным описанием



Интерфейс

1. Логотип Mimic

- Расположен в левом верхнем углу.
- Является ссылкой на главную страницу со списком всех проектов системы.

2. Название текущего проекта



- Отображает имя открытого проекта (например, “Проект для справки”)
- Не является кликабельным, но даёт контекст текущей работы

3. Блок пользователя Содержит аватар и имя (например, “Иванов Иван”). При нажатии открывается контекстное меню:

- Мой Профиль
- Администрирование (если доступно)
- Журнал событий
- Справка
- Версия
- Выход

4. Панель управления

- Поиск - фильтрация объектов в дереве по имени
- Кнопка “Журнал событий” открывает Журнал событий для текущего проекта
- Кнопка закрепления дерева («гвоздик» / «крестик»):
 - Отображается **✕ Закреплено** — дерево остаётся видимым постоянно

- Отображается  **Откреплено** — дерево сворачивается влево при потере фокуса и раскрывается при наведении на кнопку 

- **Регулировка ширины** — перетаскивание за тройную точку : по центру правой границы всей панели с деревом.

5. **Шапка выбранного объекта** Отображается в верхней части страницы объекта и содержит:

- Тип и название объекта (например, “Сценарий: Стандартный ответ”)
- Статус активности (переключатель “Активен” / “Неактивен”)
- Кнопку редактирования объекта (Сценарий, Endpoint, Папка)

6. **Дерево проекта** Иерархическая структура с поддержкой вложенности и перетаскивания:




Уровень	Тип объекта	Возможности
Уровень проекта	Папка проекта	Может содержать другие папки проекта или endpoint’ы. Поддерживает перетаскивание между папками и в корень.
Уровень endpoint’а	Endpoint	Может находиться в корне проекта или в папке проекта.
	Папка уровня endpoint’а	Создаётся внутри endpoint’а; может содержать сценарии и вложенные папки endpoint’а.
Уровень сценария	Сценарий	Привязан к одному endpoint’у; может располагаться: — напрямую в endpoint’е, — в папке уровня

Уровень	Тип объекта	Возможности
		endpoint'a. Перетаскивание в папку проекта не поддерживается.

Перетаскивание: осуществляется за маркер :: в начале строки объекта

7. **Основной контент** (центральная область) Отображает данные выбранного объекта. Все разделы расположены на одной странице с прокруткой

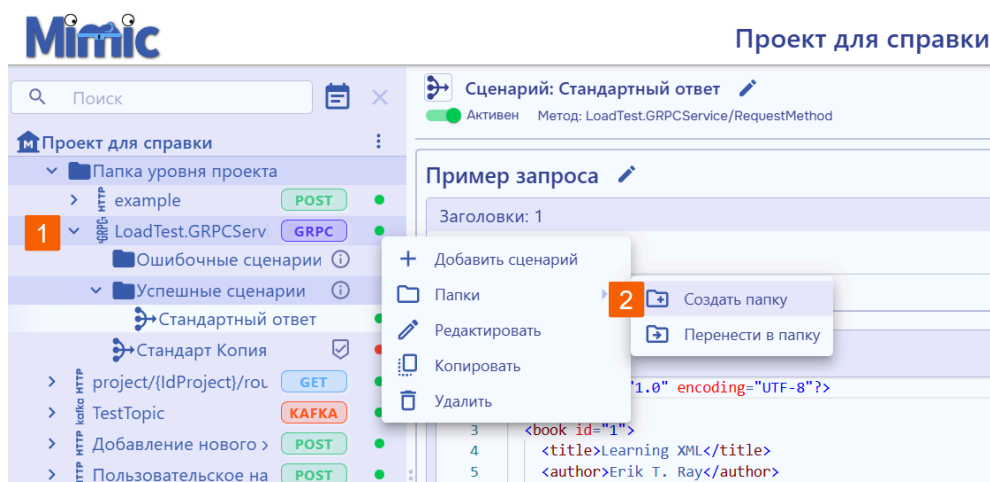
8. **Панель “Содержание”** (справа) Оглавление текущего объекта Можно открепить/закрепить содержание:

- Отображается  **Закреплено** — панель всегда видна
- Отображается  **Откреплено** — панель сворачивается в область по центру с права и раскрывается при наведении на неё 

Создание папки

В Mimic папки используются для логической группировки объектов и поддерживаются на двух уровнях:

- **Уровень проекта** — папки для группировки endpoint'ов и других папок
- **Уровень endpoint'a** — папки внутри конкретного endpoint'a для группировки сценариев



Вызов_создания_папки

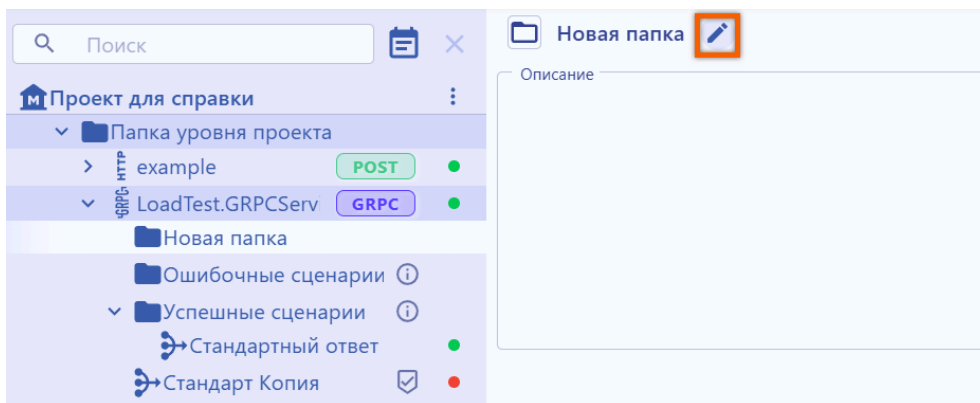
Описание действий

1. Вызовите контекстное меню (ПКМ или значок :) по уже существующему объекту в дереве:
 - по имени проекта
 - по папке проекта/endpoint'a
 - по endpoint'у
2. В меню выберите **“Создать папку”**

Результат: новая папка появляется в дереве непосредственно под тем объектом, для которого было вызвано меню с именеи **“Новая папка”**

Переименование и редактирование папки

Чтобы переименовать папку или изменить её описание, выберите нужную папку в дереве и нажмите кнопку карандаш.



Вызов_редактирование_папки.png

Форма редактирования папки


Редактирование папки


1 Название
Имя папки

2 Описание
Любое пользовательское описание папки

3 Сохранить ОТМЕНА


Редактирование_папки

1. **Наименование** - впишите желаемое имя папки
2. **Описание** - необязательно, но полезно для документирования назначения папки, этот текст можно будет просматривать при наведении на иконку  в дереве.
3. Нажмите **“Сохранить”**, чтобы применить изменения

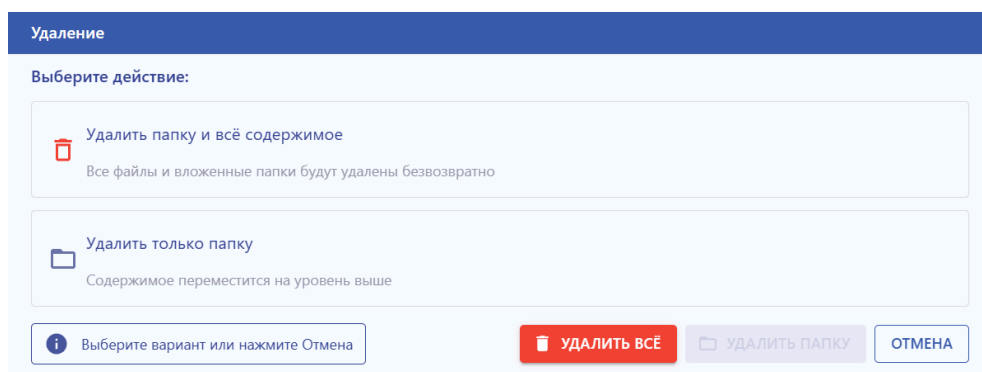
 **Советы по использованию** Используйте папки проекта для разделения по доменам (например, *Авторизация, Платежи, Уведомления*). Используйте папки endpoint'а для сценариев по статусам: *200 OK, 400 Validation, 500 Errors*. Имена папок должны быть понятными и без спецсимволов (допускаются буквы, цифры, дефисы и подчёркивания).

Удаление папки

Чтобы удалить папку:

1. Вызовите контекстное меню, нажав ПКМ или значок  : рядом с названием папки в дереве объектов.
2. В открывшемся меню выберите пункт «Удалить».

Если папка содержит объекты, откроется диалог подтверждения с выбором действия:



Удаление

Варианты удаления:

- **“Удалить папку и всё содержимое”** — безвозвратно удаляет папку вместе со всеми вложенными endpoint'ами, сценариями и подпапками.
- **“Удалить только папку”** — папка удаляется, а её содержимое перемещается на уровень выше (в родительскую папку или корень объекта).

Операция недоступна, если у вас недостаточно прав (роль «Читатель»)

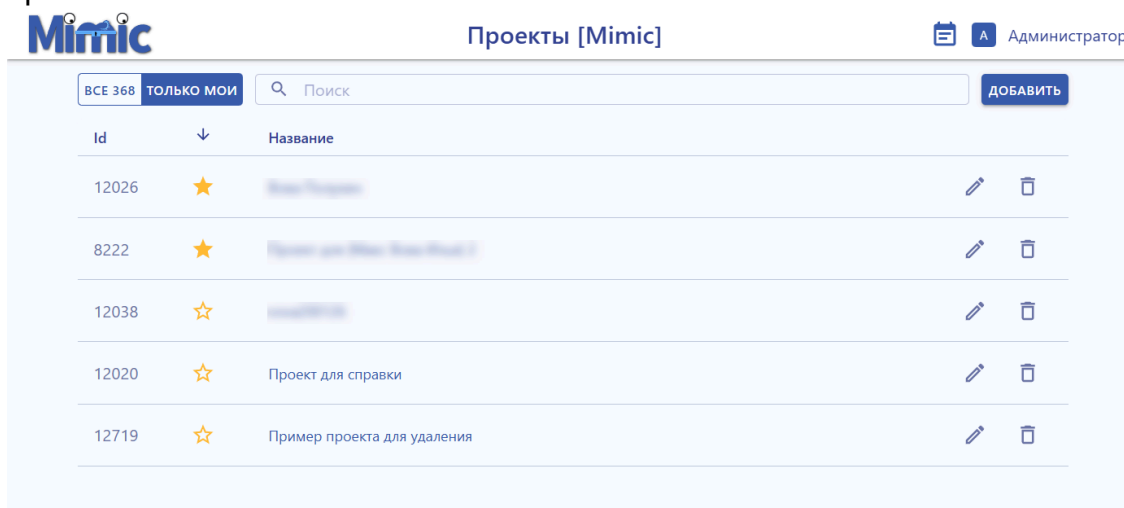
Проект

Проект - это изолированная рабочая область, в которой создаётся имитация поведения одного или нескольких внешних API, сервисов или систем. Каждый проект полностью независим от других: его настройки, Endpoint'ы, сценарии и данные не пересекаются с другими проектами. Это позволяет, например, создать несколько проектов для одного и того же мокируемого сервиса — чтобы изолировать конфигурации разных команд или сценарии тестирования.

⚠ Важно В текущей версии изоляция **gRPC-соединений** между проектами ограничена. Если в разных проектах используются сервисы с одинаковыми **package** и **service** (указанными в **.proto-файлах**), возможна некорректная маршрутизация запросов. В таких случаях система будет направлять вызовы к последнему созданному соединению, соответствующему данному **package.service**.

Просмотр проектов

После входа в систему вы автоматически попадаете на страницу со списком проектов.



The screenshot shows the Mimic web interface for project management. At the top, there is a header with the Mimic logo, the title "Проекты [Mimic]", and the user role "Администратор". Below the header, there is a search bar with a search icon and the text "Поиск", and a "ДОБАВИТЬ" button. The main content area displays a table of projects with columns for "Id", "Название", and action icons (edit and delete). The table contains five rows of project data.

Id	Название	Действия
12026	Project 1	✎ 🗑
8222	Project 2	✎ 🗑
12038	Project 3	✎ 🗑
12020	Проект для справки	✎ 🗑
12719	Пример проекта для удаления	✎ 🗑

На этой странице доступны:

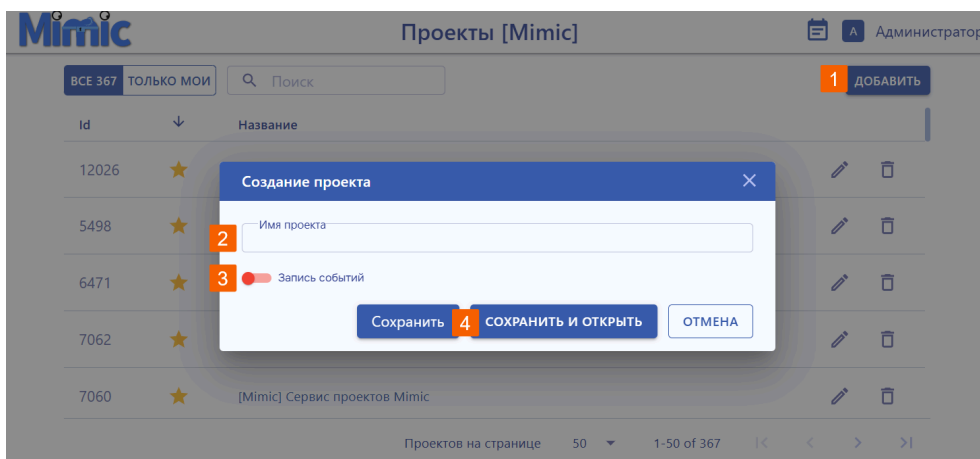
- “Поиск” по названию проекта
- Два режима отображения списка:

- “**Все**” — все проекты, к которым у вас есть доступ
- “**Только мои**” — проекты, где вам назначена роль “Владелец”
- “**Создание нового проекта**” — нажмите кнопку “**Добавить**”
- “**Редактирование имени проекта**” — нажмите значок карандаша рядом с названием (доступно при наличии соответствующих прав)
- “**Удаление проекта**” — нажмите значок корзины рядом с проектом (доступно при наличии соответствующих прав)

Создание проекта

Создать проект можно с главной страницы системы Mimic, где отображается список всех существующих проектов.


Пример добавление проекта:




Добавление проекта

Описание действий

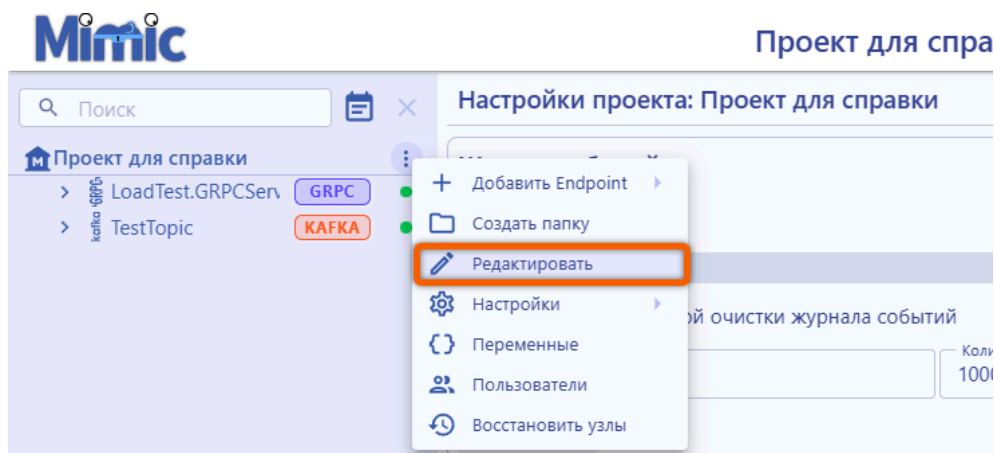
1. Нажмите кнопку “**Добавить**” в правом верхнем углу экрана
2. В поле “**Имя проекта**” укажите название имитации
3. При необходимости включите опцию “**Запись событий**” (подробнее см. в разделе [Журнал событий](#))
4. После заполнения формы:
 - Нажмите “**Сохранить и открыть**”, чтобы сразу перейти к настройке проекта
 - Или “**Сохранить**”, если хотите создать проект без немедленного открытия

В шапке каждой страницы Mimic в имени текущего пользователя расположено меню пользователя, из которой можно мгновенно перейти в **Журнал событий** — значок  Подробнее см. в разделе Журнал событий

 **Права пользователей** Создать проект может любой пользователь системы. После создания он становится **“Владельцем”** этого проекта.

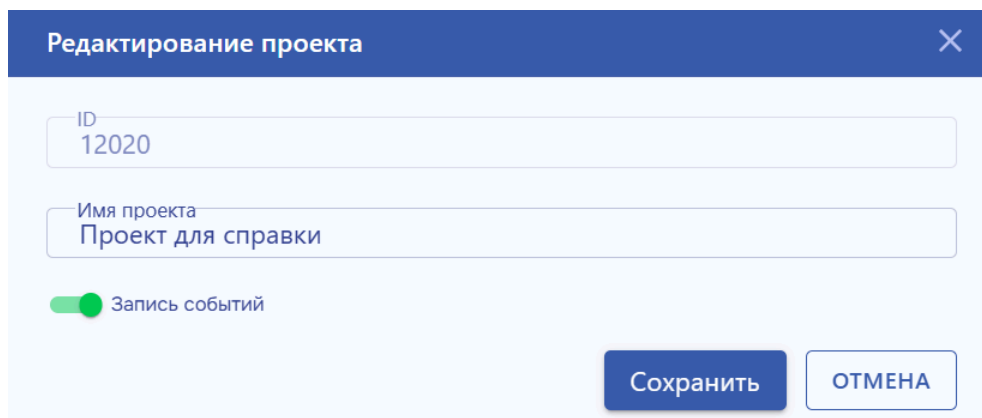
Редактирование проекта

Форму редактирования проекта можно открыть через контекстное меню проекта в левой панели (дереве) — выберите пункт **“Редактировать”**



Проект_меню_редактирование

Форма содержит три параметра:

The screenshot shows a modal window titled 'Редактирование проекта'. It contains three input fields: 'ID' with the value '12020', 'Имя проекта' with the value 'Проект для справки', and a toggle switch for 'Запись событий' which is currently turned on. At the bottom right, there are two buttons: 'Сохранить' and 'ОТМЕНА'.

Редактирование проекта

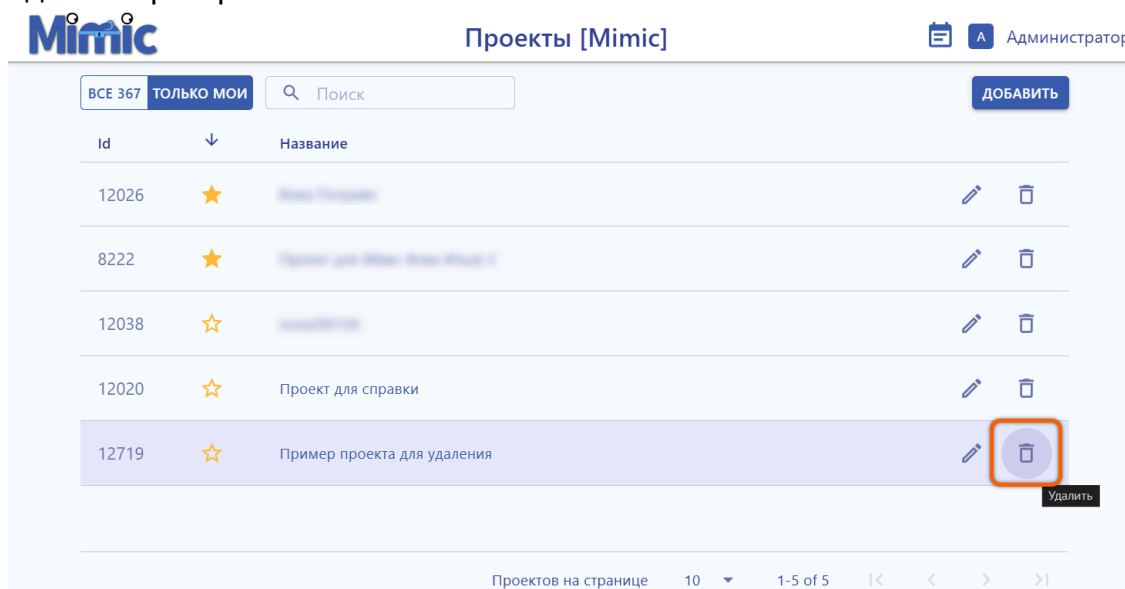
- **ID проекта** Уникальный идентификатор, присваиваемый при создании проекта. Используется для внутренней ссылки на проект в базе данных и

внешних интеграциях. Изменить ID невозможно — он гарантирует однозначную идентификацию проекта в системе.

- **Имя проекта** Понятное человеку название, отображаемое в интерфейсе. Может изменяться в любой момент и используется для удобной навигации и поиска среди множества проектов.
- **Запись событий** Опция, включающая или отключающая логирование всех запросов и ответов в рамках проекта. При активации данные сохраняются в Журнал событий

Удаление проекта

Удалить проект может только владелец проекта или пользователь с ролью администратора.



Порядок удаления:

1. Перейдите на главную страницу, где отображается список всех проектов
2. Нажмите значок корзины напротив проекта, который требуется удалить
3. Подтвердите удаление во всплывающем диалоге


⚠️ Внимание После удаления все связанные объекты (Endpoint'ы, сценарии, ответы и т.д.) будут безвозвратно удалены. Восстановление невозможно.

Переход в настройки проекта

Страница “**Настройки проекта**” позволяет управлять глобальными параметрами проекта:

- поведением журнала событий
- переменными проекта
- ролями и доступом пользователей
- интеграциями (HTTP/gRPC/Kafka)

Для перехода в настройки проекта выполните одно из следующих действий:

- кликните по имени проекта в левой панели (дерево проекта)
- или откройте контекстное меню проекта (через значок , расположенный рядом с названием проекта) и выберите в выпадающем списке “**Настройки**” и нужный раздел (например, “Переменные проекта”, “Пользователи”, “Журнал событий” и др.)

Настройка Журнал событий

В этом разделе можно:

- Включить или отключить запись событий
- Настроить правила автоматической очистки журнала
- Выполнить мгновенную очистку журнала вручную
- Посмотреть объём дискового пространства, занимаемого событиями

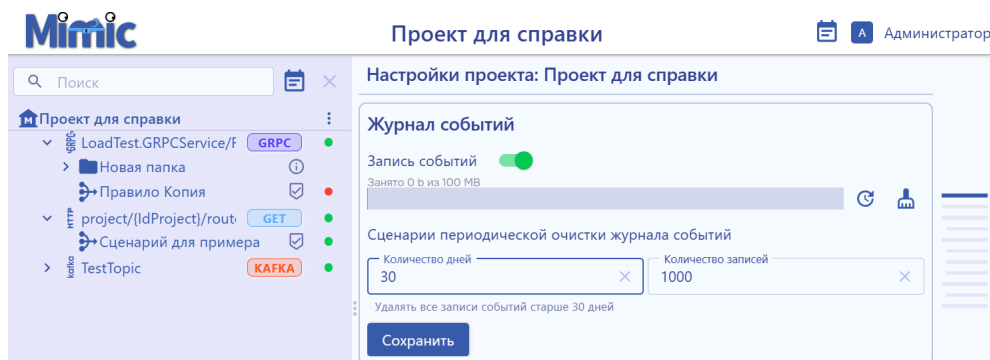
Подробнее о просмотре событий и работе с журналом можно ознакомиться в разделе Журнал событий

Со временем журнал может накопить большой объём данных, что затруднит поиск нужной информации и снизит производительность фильтрации

⚠️ Рекомендация Чтобы избежать перегрузки журнала, рекомендуется либо настроить автоматическую очистку старых записей, либо отключить запись событий, если логирование не требуется для текущего проекта.

Правила автоматической очистки

Настройка правил доступна только при включённой записи событий на проекте.



Настройка_журнала_событий

Доступны два независимых правила, которые применяются в строгом порядке:

1. Очистка по возрасту записей (в днях)


Удаляются все события, старше указанного количества дней. *Пример: хранить только записи за последние 30 дней.*

2. Очистка по максимальному количеству событий


В журнале сохраняется только заданное число самых свежих записей. *Пример: ограничить журнал 1000 последними событиями.*



шкала_памяти

Над настройками отображается шкала использования дискового пространства, показывающая объём данных, занимаемый журналом событий для данного проекта. Данные обновляются периодически. Чтобы получить актуальную информацию, нажмите кнопку 

Мгновенная очистка журнала

Для немедленного удаления записей используйте кнопку . Откроется форма очистки памяти

Очистка памяти

Количество дней

Количество записей
1000

ОЧИСТИТЬ ЗАПИСИ ЖУРНАЛА ПО ПРАВИЛУ

УДАЛИТЬ ВСЕ ЗАПИСИ

ОТМЕНА

Форма очистки

В данной форме можно:

- скорректировать параметры очистки и принудительно запустить очистку по нему
- удалить все записи событий по проекту

Подробнее про просмотр событий можно узнать в разделе Журнал событий.

Переменные проекта

В этом разделе отображаются переменные, действующие на уровне всего проекта. Здесь можно создавать, редактировать и удалять переменные проекта. Подробнее о работе с переменными см. в разделе Переменные

Настройки HTTP, HTTPS

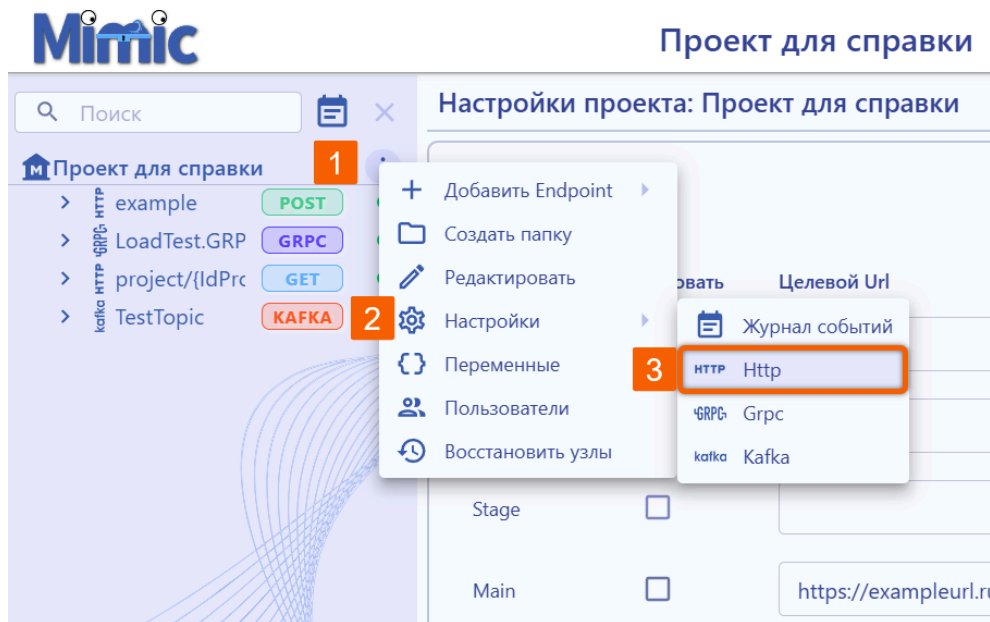
Mimic поддерживает мокирование как HTTP, так и HTTPS-запросов — это зависит от конфигурации инфраструктуры, в которой развернуто приложение.

На странице настроек HTTPS можно настроить проксирование запросов на реальный сервис. Также доступна функция импорта маршрутов с примерами запросов и ответов на основе спецификации OpenAPI

Проксирование HTTP/HTTPS - это механизм переадресации входящего запроса от клиента на целевой сервер (или сервис) с последующей передачей ответа от этого сервера обратно клиенту. Такой подход позволяет комбинировать моки и реальные вызовы, упрощает интеграционное тестирование и обеспечивает гибкость при разработке и отладке API.

Настройка проксирования HTTP,gRPC

Раздел “Проксирование” позволяет управлять режимом перенаправления запросов на реальные сервисы. Для выполнения настроек проксирования HTTP или GRPC сервисов перейдите в настройки проекта. Проксирование для топиков Kafka настраивается на уровне Endpoint



Открытие настроек

Описание действий:

1. Вызовите контекстное меню у проекта (например, правой кнопкой мыши или через значок ” : “)
2. Выберите пункт “**Настройки**”
3. В выпадающем меню выберите нужный протокол: “**Http**” или “**gRPC**”

Откроется соответствующая страница настроек проксирования Пример для HTTP/GRPC проксирования

Настройки проекта: Проект для справки

Настройки HTTP

Проксирование

Стенды	Проксировать	Целевой Url
Dev	<input type="checkbox"/>	<input type="text"/>
Test	<input checked="" type="checkbox"/>	<input type="text" value="https://exampleurl.ru"/>
Stage	<input type="checkbox"/>	<input type="text"/>
Main	<input checked="" type="checkbox"/>	<input type="text" value="https://exampleurl.ru"/>

Проксирование HTTP

Описание действий:

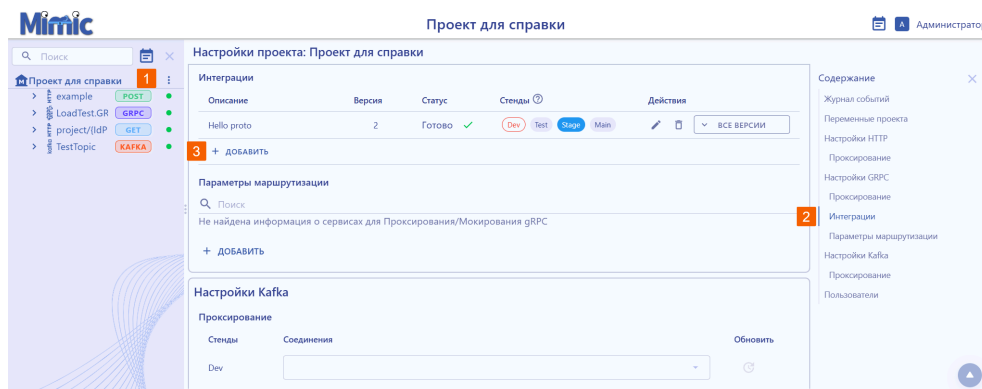
1. В поле **“Целевой URL”** укажите адрес реального сервиса, на который будут перенаправляться запросы (например, `https://api.example.com`)
2. Активируйте переключатель **“Проксировать”** для нужного контура (среды).
Переключатель недоступен, если поле **“Целевой URL”** не заполнено
3. Настройки сохраняются автоматически

⚠ Важно: Если контуры изолированы, проксирование запросов между разными контурами (например, из dev в prod) недоступно.

Настройки gRPC Интеграции

Раздел **“Интеграции”** позволяет создать gRPC-контракт — описание интерфейса внешнего сервиса на основе .proto-файлов. На его основе Mimic сможет сгенерировать клиентский код для мокирования и проксирования gRPC-вызовов.

Пример создания интеграции

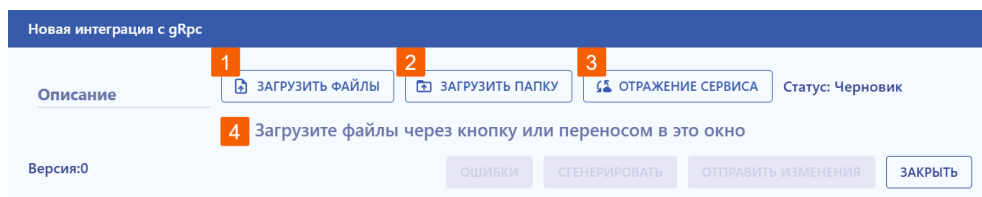


Интеграции gRPC

Описание действий:

1. Нажмите на имя проекта, чтобы открыть его настройки
2. Перейдите в раздел **“Настройки gRPC/Интеграции”**
3. Нажимаем кнопку **“+ Добавить”**

Откроится форма создания интеграции

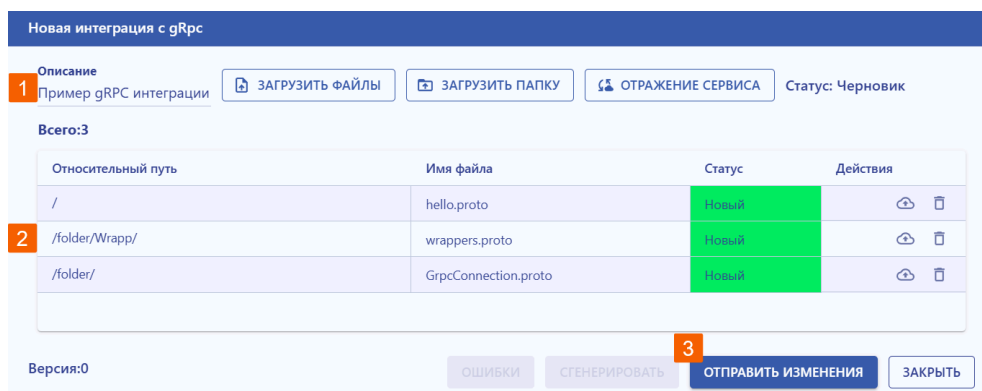


Методы загрузки

Форма поддерживает четыре способа загрузки данных в gRPC-контракт:

1. **Загрузка файлов** — нажмите кнопку **“Загрузить файлы”**, чтобы выбрать один или несколько .proto-файлов
2. **Загрузка папки** — если файлы организованы в иерархию каталогов, используйте **“Загрузить папку”** — это позволит сохранить структуру для последующих связей
3. **Рефлексия сервиса** — нажмите кнопку **“Отражение сервиса”** и укажите адрес работающего gRPC-сервиса, поддерживающего Server Reflection. Файлы контракта будут сформированы и загружены автоматически
4. **Drag & Drop** — перетащите файлы или папку прямо на форму создания соединения

После загрузки файлов в форму они отобразятся в списке

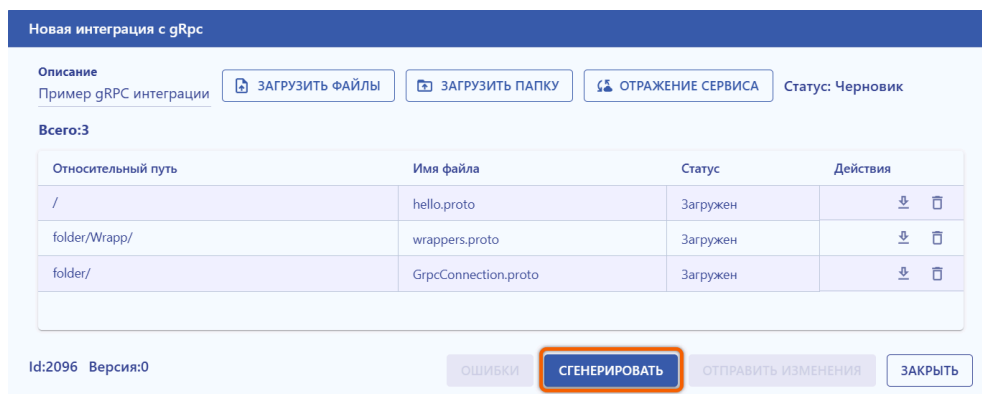


Загрузка GRPC

Для загрузки файлов в систему необходимо выполнить следующие действия:

1. **Укажите описание интеграции** — кратко опишите назначение или контекст использования (например, “Платёжный шлюз”)
2. При необходимости скорректируйте относительные пути для файлов, чтобы они соответствовали структуре, указанной в директивах `import` внутри `.proto`. Это необходимо для корректного разрешения зависимостей при генерации
3. Нажмите **“Отправить изменения”**, чтобы сохранить файлы на сервере

После отправки интеграция получает статус **“Черновик”**.



Генерация

Чтобы завершить создание, нажмите кнопку **“Сгенерировать”**. Система сгенерирует клиентскую библиотеку для взаимодействия и интеграция перейдёт в статус **“Готово”** для использования в маршрутах и сценариях.

Управление интеграциями GRPC

Все созданные интеграции отображаются в едином списке, включая предыдущие версии. Чтобы просмотреть историю версий, нажмите кнопку ▼ ВСЕ ВЕРСИИ Такой

подход обеспечивает:

- Гибкость: разные стенды могут использовать разные версии одной интеграции
- Изоляцию: локальные правки не требуют повторной загрузки .proto-файлов — достаточно обновить нужную версию

Описание	Версия	Статус	Стенды ?	Действия
Hello proto 1	3*	Готово ✓	Dev Test Stage Main	✎ 🗑 ^ СКРЫТЬ
Hello proto	1	Готово ✓	Dev Test Stage Main	✎ 🗑 11.02.2026 14:36:42
Hello proto	2	Готово ✓	Dev Test Stage Main	✎ 🗑 22.01.2026 12:24:24
Hello proto 2	3	Готово ✓	Dev Test Stage Main	✎ 🗑 26.03.2026 21:53:32
Hello proto	0	Черновик	Dev Test Stage Main	✎ 🗑 26.03.2026 21:54:46
Пример gRPC интеграции	1	Готово ✓	Dev Test Stage Main	✎ 🗑 ВСЕ ВЕРСИИ
+ ДОБАВИТЬ				

gRPC Соединения

Описание элементов интерфейса

1. **Агрегирующая запись интеграции** Отображает актуальное состояние интеграций и содержит следующие поля:

- **Описание** — произвольное имя, заданное пользователем для идентификации (например, “Платёжный шлюз v2”).
- **Версия** — номер последней завершённой версии (присваивается при переходе в статус “Готово”) Если рядом с номером отображается звёздочка (*), это означает, что существует черновик (интеграция в статусе “Черновик”).
- **Статус** — текущее состояние:
 - “Черновик” — изменения сохранены, но код не сгенерирован
 - “Готово” — клиентская библиотека успешно сгенерирована и готов к использованию
 - “Ошибка” — не удалось сгенерировать библиотеку (проверьте корректность .proto-файлов)
- **Стенды** — отображает, как используется интеграция на каждом контуре.

Обозначения:

- **Main** - стенд автоматически использует последнюю версию в статусе “Готово”. При публикации новой версии она применяется мгновенно
- **Main** - стенд использует конкретную версию, выбранную вручную. Новые версии не применяются автоматически. Чтобы узнать, какая

именно версия используется, раскройте список версий

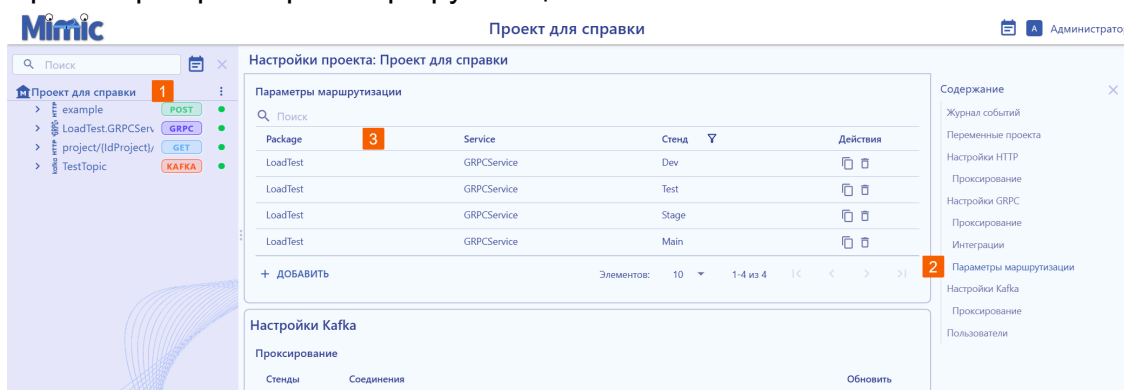
- o **Main** - интеграция не используется на этом стенде

2. **Раскрытый список версий** При раскрытии записи отображаются все сохранённые версии интеграции. В колонке **“Стенды”** для каждой версии указано, на каких стендах она используется.

Автоматическая регистрация в параметрах маршрутизации

После выбора среды использования для интеграции gRPC в таблице параметров маршрутизации автоматически появляется запись с указанием Package, Service и стенда. При отключении контракта, эта запись удаляется автоматически — синхронизация происходит без участия пользователя.

Просмотр параметров маршрутизации



Описание действий:

1. Нажмите на имя проекта, чтобы открыть его настройки
2. Перейдите в раздел **“Настройки GRPC маршрутизации”**
3. Отобразится список активных записей маршрутизации

i Примечание Если вы хотите использовать проксирование gRPC без полного разбора сообщений (например, для простого транзитного прокси), вы можете вручную добавить запись с указанием Package и Service. Это позволит системе сопоставить входящий вызов с текущим проектом и применить его настройки для проксирования, даже если контракт не загружен

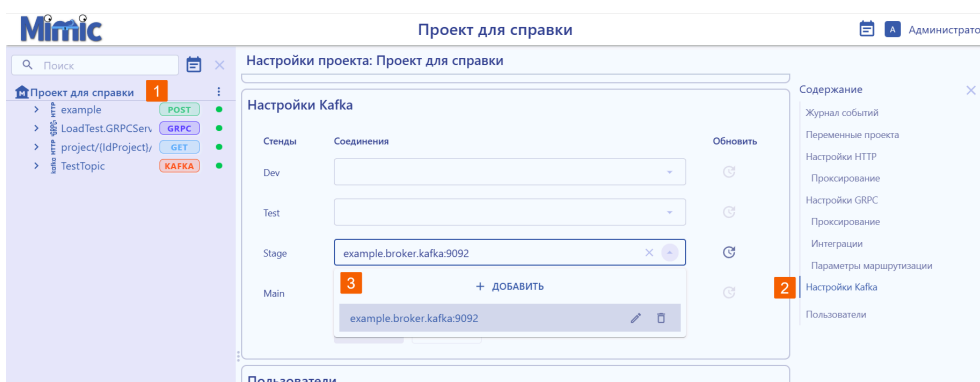
Настройки Kafka

В разделе “Настройки Kafka” можно настроить подключение к брокеру для каждого контура (стенда) Вы можете:

1. создать новое соединение, указав хост и порт Kafka-брокера
2. Или выбрать одно из ранее сохранённых соединений из выпадающего списка

Добавление Kafka-соединения

Пример добавления соединения



Kafka соединение1

Описание действий:

1. Нажмите на имя проекта, чтобы открыть его настройки
2. Перейдите в раздел “**Настройки Kafka/Интеграции**”
3. Для нужной среды раскройте список соединений
4. Нажмите кнопку “**+ Добавить**”

Откроется форма добавления соединения:

Kafka соединение2

Заполните поля:

1. “**Хост**” — адрес Kafka-брокера (например, kafka.example.com или IP)


2. “Порт” — номер порта (по умолчанию 9092)
3. Нажмите “Сохранить”, чтобы создать соединение

После сохранения новое соединение появится в списке и станет доступно для выбора на любом стенде.

Kafka соединение3

После настройки всех необходимых соединений нажмите кнопку “Сохранить” внизу блока, чтобы применить изменения.

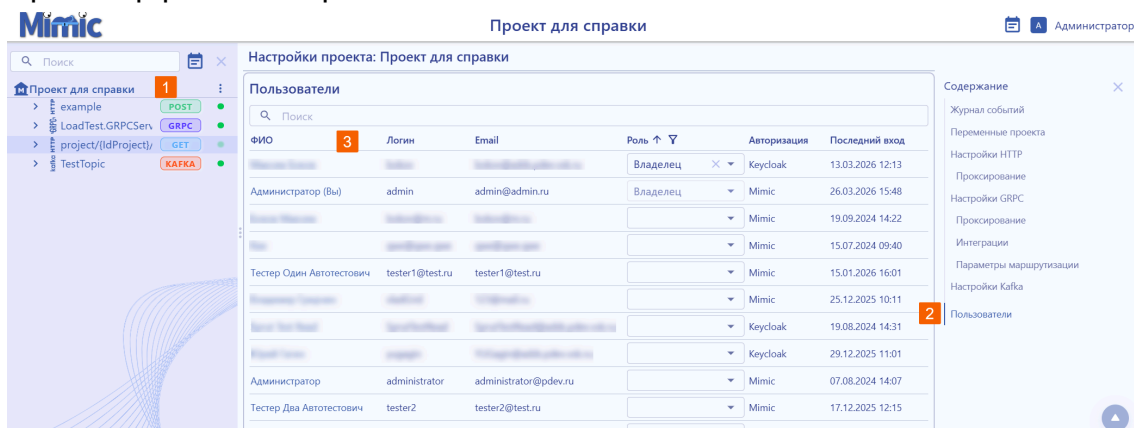
⚠ Важно При сохранении соединения Mimic инициирует автоматическое создание требуемых топиков — если в настройках брокера Kafka включена опция `auto.create.topics.enable=true`. В противном случае топикам необходимо создать вручную через административные инструменты Kafka.

i Примечание Если возникают проблемы с чтением данных из топиков, вы можете принудительно обновить состояние Kafka-соединения на агенте, нажав кнопку 

Пользователи

Доступ к изменению раздела “Пользователи” предоставляется только **Владельцу проекта** или **Администратору** системы. Этот раздел позволяет назначать пользователям роли на уровне проекта.

Просмотр ролей на проекте:



Описание действий:

1. Нажмите на имя проекта, чтобы открыть его настройки
2. Перейдите в раздел **“Пользователи”**
3. Отобразится список всех участников проекта с указанием их ролей. У пользователей без роли нет никаких прав, за исключением выданных глобальной ролью.

Роли проекта

1. Владелец проекта

- Полный доступ ко всем функциям проекта
- Может изменять пользователям роли на проекте
- Имеет полномочия на удаление проекта

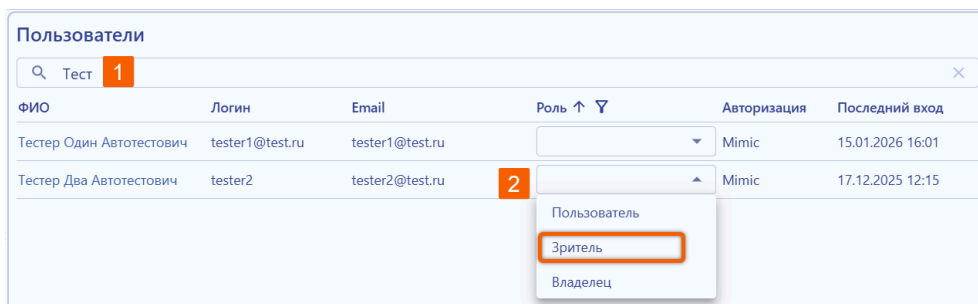
2. Пользователь проекта

- Полный доступ ко всем функциям кроме удаления проекта
- Не может назначать роли пользователям

3. Зритель проекта

- Только просмотр: маршруты, сценарии, ответы, журнал событий и др.
- Нет прав на редактирование, создание или удаление объектов
- Подходит для аудита, согласования или предоставления доступа внешним заинтересованным сторонам

Пример: добавление пользователя и назначение роли на проекте:



Добавление пользователя

Описание действий:

1. В строке поиска введите имя, email или логин пользователя, которого нужно добавить или изменить
2. В списке найдите нужного пользователя и:
 - выберите роль из выпадающего меню
 - или удалите роль (иконка крестика возле роли), чтобы исключить пользователя из проекта

i Примечание Если у пользователя назначена глобальная роль (например, «Администратор»), она имеет приоритет над ролями на уровне проекта. Такой пользователь получает соответствующий доступ ко всем проектам, даже если ему не назначена роль внутри конкретного проекта.

См. также

- [Про gRPC](#)
- [Введение в gRPC](#)
- Работа с Endpoint
- Журнал событий

Endpoint

Endpoint - это сущность имитируемого проекта, представляющая собой точку взаимодействия с проверяемым сервисом. Каждый endpoint соответствует конкретному адресу или каналу, по которому поступают входящие запросы. В рамках endpoint пользователь может настраивать сценарии обработки запросов и определять, какой ответ должен быть возвращён — в зависимости от заголовков,

тела запроса, параметров и других условий. Это позволяет точно имитировать поведение реального внешнего сервиса.

⚠ Важно Добавление метода gRPC становится доступным только после создания gRPC-интеграции. (Подробнее см. в разделе Настройки проекта -> gRPC)

Типы Endpoint'ов

Mimic поддерживает три типа endpoint'ов, соответствующих основным протоколам современных систем:

- **HTTP-маршрут** — для REST/HTTP(S)-взаимодействий
- **gRPC-метод** — для вызовов gRPC-сервисов
- **Kafka-топик** — для работы с событиями через Apache Kafka

Структура страницы Endpoint`а

Независимо от типа, страница endpoint`а состоит из следующих разделов:

- **Пример запроса** — пример входящего запроса со всеми необходимыми атрибутами (заголовки, параметры, тело и т.д.)
- **Сценарии** — правила обработки запросов с поддержкой условной логики (например: «если заголовок X = Y, вернуть ответ Z»)
- **Библиотека ответов** — хранилище готовых ответов, которые можно использовать в нескольких сценариях

Дополнительные разделы, доступные в зависимости от типа endpoint`а:

- **Ссылки для подключения** — отображаются только для HTTP-маршрутов и gRPC-методов; содержат готовые URL/gRPC-адреса чтобы указать в проверяемом сервисе для вызова
- **Проксирование** — доступно только для Kafka-топиков. Позволяет настроить перенаправление сообщений в реальные топики внешних систем

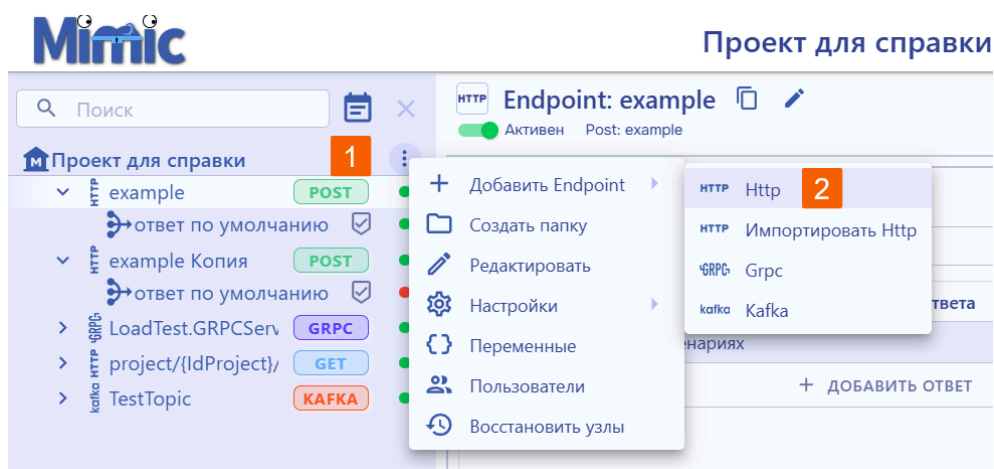
Добавление Endpoint

Добавление маршрута может быть выполнено **тремя способами**:

- **Вручную** — заполнив все необходимые поля формы (имя, путь, тип и т.д.)
- **Копированием существующего** — позволяет быстро создать похожий Endpoint, унаследовав настройки, пример запросов, все созданные сценарии и переменные исходного
- **Импортом из OpenAPI-спецификации** — автоматически создаются HTTP-маршруты на основе описанных в файле эндпоинтов, включая методы, пути и примеры запросов/ответов (доступно только для HTTP)

Добавление Endpoint - Вручную

Переход к форме создания Endpoint



Добавление маршрута

Описание действий

1. Вызовите контекстное меню проекта или папке на уровне проекта (например, правой кнопкой мыши по имени в дереве слева или через значок : рядом с его названием)
2. В открывшемся меню выберите **“+ Добавить Endpoint”**, затем подпункт нужного типа (**“HTTP”**, **“gRPC”** или **“Kafka”**)

Форма Добавления/Редактирование Endpoint для HTTP-маршрута

Добавление маршрута

В режиме создания/редактирования Endpoint необходимо указать следующие параметры:

- **Активность маршрута** - по умолчанию маршрут активен. Вы можете отключить его, если необходимо временно отключить логику обработки адреса
- **Наименование** - пользовательское название маршрута, отображаемое в навигационном дереве интерфейса. Если поле не заполнено, наименование будет сформировано автоматически на основе значения поля Endpoint
- **Тип маршрута** - HTTP-метод, определяющий логику взаимодействия с маршрутом. Доступные варианты: GET, POST, PUT, PATCH, DELETE
- **Адрес (Endpoint)** — относительный путь, по которому будет доступен маршрут. Указывается без базового URL (BaseUrl). Переменные значения нужно именовать и брать в двойные скобки, например `{{Id}}`, значения этих переменных можно будет использовать в ответных сообщениях

Заполнив все поля нажмите кнопку “**Сохранить**”.

После создания маршрута, имитация будет доступна по полному URL, как показано на рисунке ниже:



<https://mimic-entypoint.example.ru/123/example/object/{{Id}}/edit>

Alt text

1. Базовый адрес точки доступа
2. ProjectId (ID проекта в Mimic)
3. **Endpoint** указанный пользователем

Указанная строка отображается в разделе “**Ссылки для подключения**” на странице Endpoint

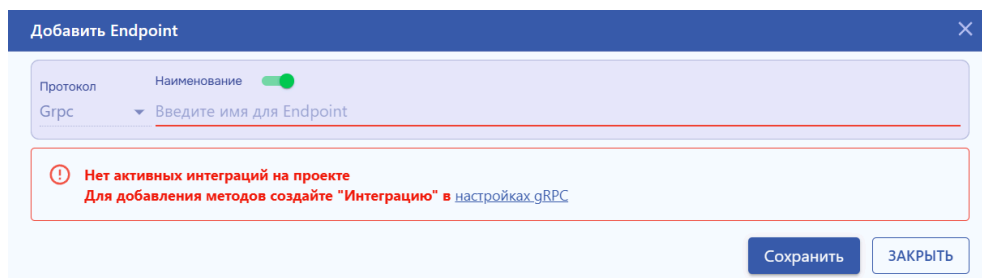
Форма Добавления/Редактирование Endpoint для gRPC-метода

Требования для работы имитаций gRPC Для полноценной работы механизма мокирования gRPC-вызовов необходимо:

- создать интеграции - позволяет выбирать методы из списка при создании нового Endpoint.
- активировать интеграцию нужной версии для соответствующей среды

- должны быть указаны **Package, Service, Стенд** (заполняются автоматически после активации интеграции)

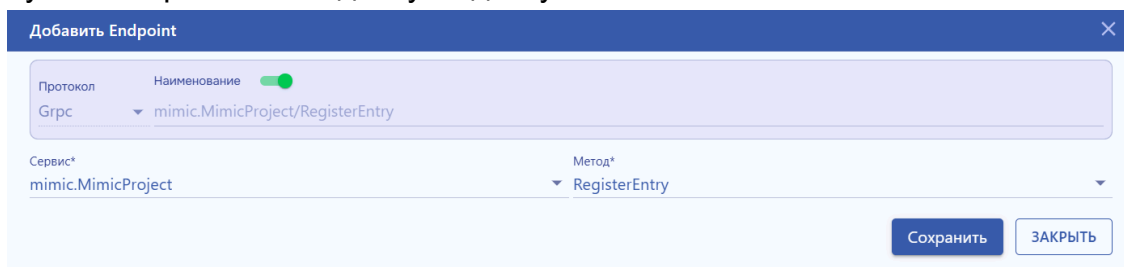
Первый вызов формы создания Endpoint для GRPC:



Добавление метода

Для проекта, в котором ещё не создано gRPC-интеграция, система покажет уведомление с возможностью перейти в настройки для его создания на основе proto файлов. Подробнее с данной операцией можно ознакомиться в разделе настройках проекта

Если соединение уже создано, откроется форма, в которой можно выбрать нужный сервис и метод из уже доступных:



Описание действий

1. В поле **Сервис** выберите нужный gRPC-сервис из списка
2. В поле **Метод** выберите метод этого сервиса

i Примечание Если в списке отсутствует необходимый сервис или метод создайте новое или измените имеющееся соединение в Настройка gRPC

3. При необходимости укажите пользовательское имя endpoint в поле **Наименование**. По умолчанию оно формируется автоматически из названий сервиса и метода
4. Нажмите кнопку **Сохранить**, чтобы применить изменения

Форма Добавления/Редактирование Endpoint для Kafka-топика

Пример добавления Kafka-топика:

The screenshot shows a form titled "Добавить Endpoint" with the following fields and controls:

- 1** **Активность endpoint'a**: A toggle switch is turned on.
- 2** **Наименование**: A dropdown menu is set to "Топик для проверки события".
- 3** **Топик**: A text input field contains "testevent".
- 4** **Топик для подключения**: A text input field contains "Mimic12020_testevent".
- 5** **Проксирование**: A radio button is selected.
- 6** **Сохранить**: A blue button at the bottom right.
- ЗАКРЫТЬ**: A button at the bottom right.

Добавление топика

При создании или редактировании endpoint'a типа Kafka-топик необходимо указать следующие параметры:

- 1. Активность endpoint'a** - по умолчанию endpoint активен. Вы можете отключить его, если нужно временно приостановить обработку сообщений из этого топика
- 2. Наименование** - пользовательское название маршрута, отображаемое в навигационном дереве интерфейса. Если поле не заполнено, наименование будет сформировано автоматически на основе значения поля "Топик"
- 3. Топик** — Имя входящего Kafka-топика, поведение которого вы хотите имитировать
- 4. Топик для подключения** - заполняется автоматически. Имя генерируется по шаблону: *Mimic{ProjectId}_{TopicIn}*. При активации подключения к Kafka в выбранной среде Mimic попытается автоматически создать топик с таким именем

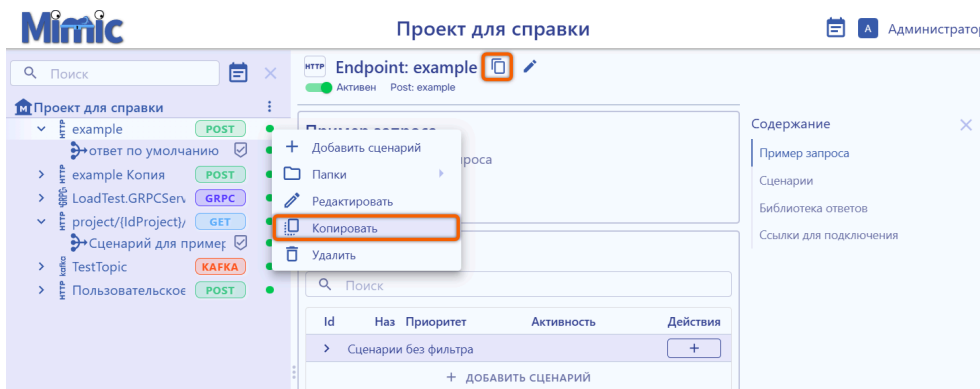
⚠ Важно Если у Mimic нет прав на создание топиков, вам потребуется создать его вручную с точно таким же именем

5. Проксирование - при включённой опции проксирования, Мимик перенаправляет сообщения из топика *Mimic{ProjectId}_{TopicIn}* в исходный топик, указанный в поле "Топик", если ни один из сценариев не сработал. Это позволяет комбинировать имитацию и реальное взаимодействие

6. После заполнения всех полей нажмите кнопку "Сохранить"


Копирование Endpoint

Вы можете быстро создать новый метод, скопировав и адаптировав уже настроенный. Это особенно полезно при работе с похожими сценариями или при переносе конфигураций между проектами. Процедура копирования одинакова для всех типов endpoint'ов (HTTP, gRPC, Kafka)

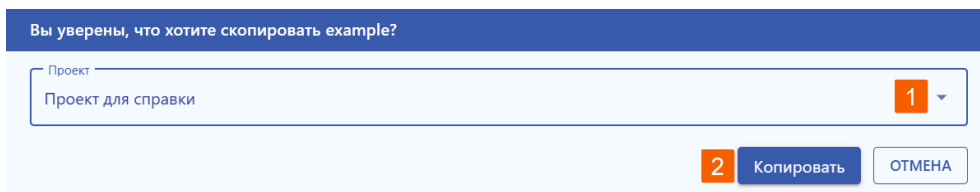


Выбор Endpoint

Описание действий

1. Перейдите в нужный проект и в левом навигационном дереве выберите endpoint, который хотите скопировать
2. Копирование доступно
 - через контекстное меню (значок : рядом с endpoint в дереве)
 - на странице самого endpoint по кнопке 

Откроется диалоговое окно копирования Endpoint



Копирование Endpoint

Описание действий

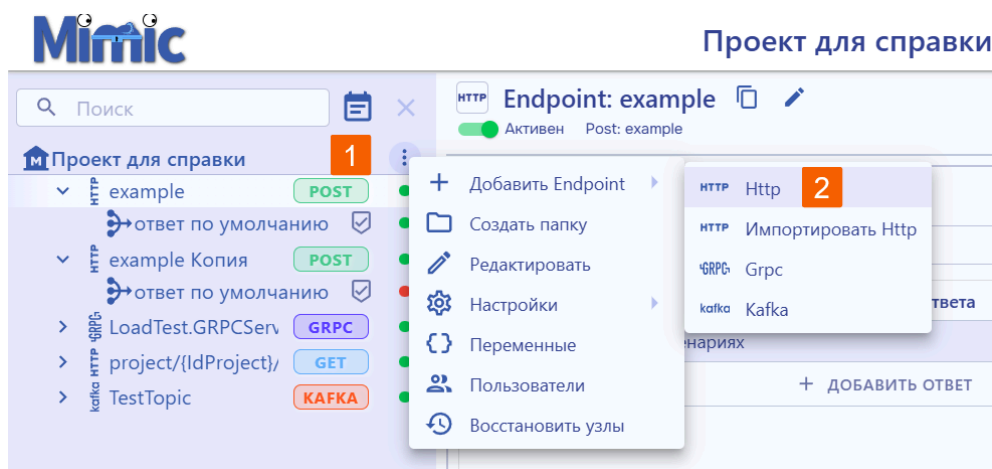
1. Выберите проект, в который нужно скопировать endpoint. По умолчанию выбран текущий проект
2. Нажмите кнопку **“Копировать”**, чтобы подтвердить операцию

После завершения копирования откроется страница нового маршрута с предзаполненными настройками.

Примечание Скопированный Endpoint сохраняет все настройки оригинала — сценарии, примеры запросов, фильтры, переменные и ответы — что позволяет быстро адаптировать его под новую имитацию

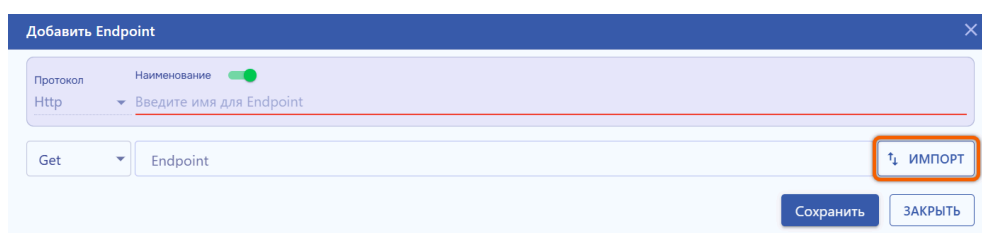
Импорт из спецификации OpenAPI

Перейдите в нужный проект и откройте форму добавления Endpoint:



Создание_маршрута

Форма Добавления/Редактирование Endpoint



Создание_маршрута_импорт

Описание действий Для импорта данных из OpenAPI необходимо нажать кнопку “Импорт” Вам будет доступна загрузка спецификации по URL или из файла. Подробнее процесс импорта описан в разделе Импорт

Пример запроса

Пример запроса позволяет:

- наглядно увидеть структуру входящего вызова (заголовки, параметры, тело),
- использовать его как шаблон при создании сценариев,
- проверять корректность настроенных фильтров и переменных.

Подробнее о создании и редактировании примера запроса см. в разделе Пример запроса.

Сценарии

На разделе сценария отображаются все созданные сценарии для выбранного Endpoint'a Сценарии сгруппированы по наличию условий срабатывания:

- **Сценарии без фильтра** Срабатывают только если ни один из сценариев с фильтрами не подошёл. Такие сценарии выполняют роль сценариев по умолчанию.
- **Сценарии с фильтрами** Активируются при полном совпадении всех заданных условий (например, по значениям в заголовках, параметрах URL или теле запроса).

Ознакомится с процессом создания сценариев можно в статье Сценарии



Сценарии

Библиотека ответов

В этом разделе отображаются все ответы, созданные для выбранного endpoint'a. Каждый ответ может использоваться в нескольких сценариях Ответы на странице сгруппированы по статусу использования:

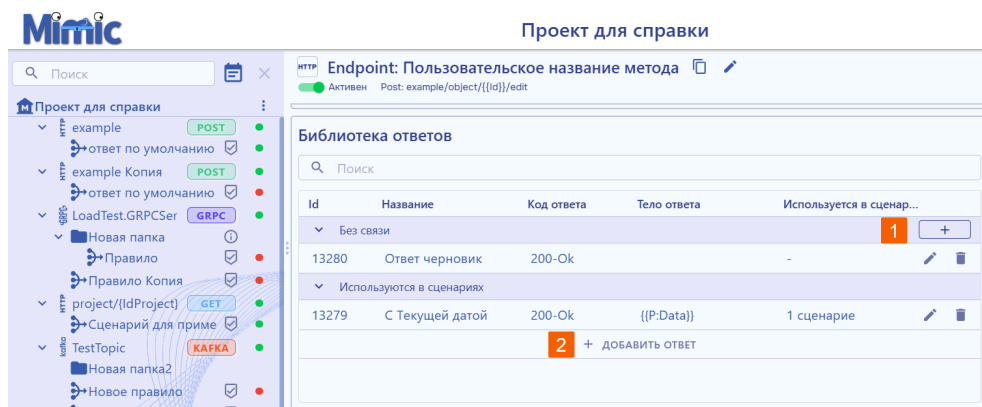
- **Без связи** — ответы, которые пока не назначены ни одному сценарию
- **Используются в сценариях** — ответы, задействованные в одном или нескольких сценариях



Библиотека_ответов

Создание ответа

Для создания нового ответа перейти в раздел “Библиотека ответов”



Библиотека_добавление_ответа

Нажмите одну из кнопок добавить:

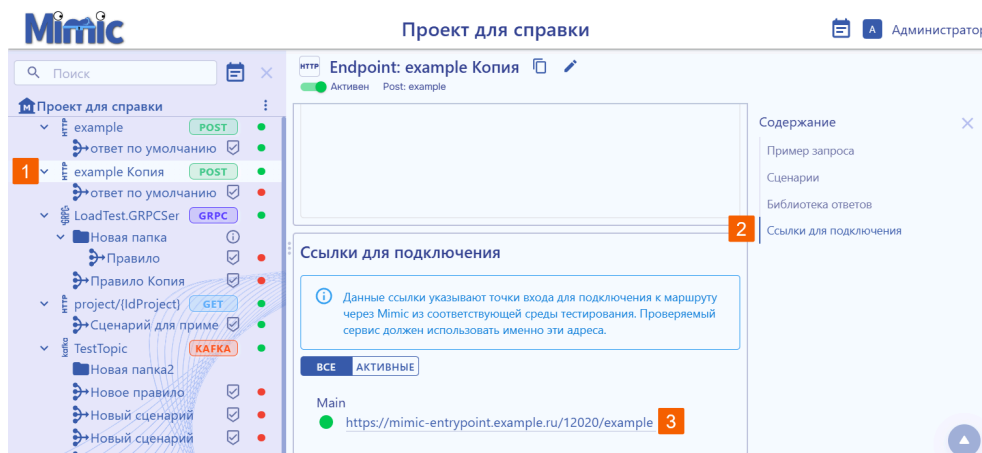
1. в списке ответов **Без связи** кнопка **+**
2. в конце таблицы, кнопка “**+ Добавить ответ**” Откроется стандартная форма создания ответа Подробнее о создании ответа см. в статье ответные сообщения

Ссылка для подключения сервиса

Готовые адреса для обращения к имитации. Генерируются автоматически для каждого контура и могут быть легко скопированы для использования в настройках ваших сервисов или внешних приложений.

⚠ Важно Раздел **Ссылки для подключения** доступен только для Endpoint типа **HTTP-маршруты** и **gRPC-методы**

Чтобы получить URL для подключения тестируемого сервиса к моку, перейдите на страницу нужного Endpoint.



Использование Urls для мокирования

Описание действий

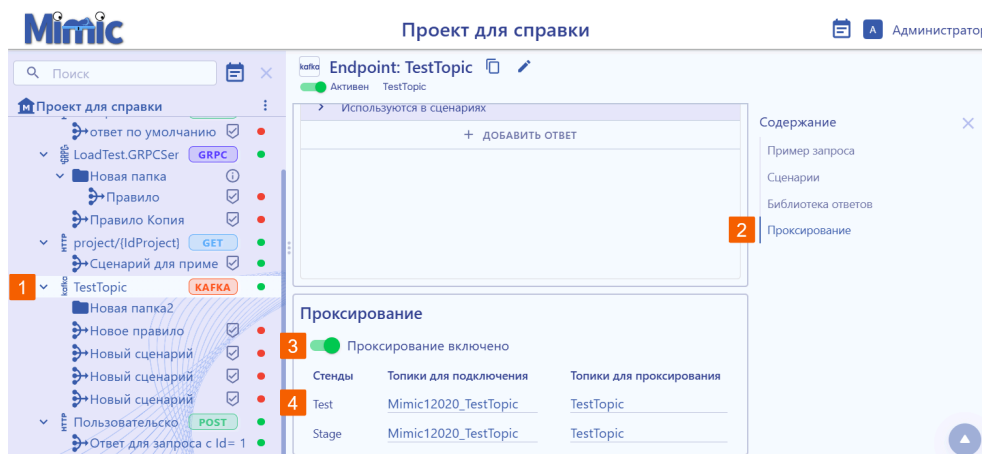
1. В навигационном дереве слева выберите требуемый Endpoint
2. Перейдите в раздел **Ссылки для подключения**
3. Скопируйте соответствующий вашей среде адрес подключения, который можно использовать для обращения к имитации

⚠ Важно Для маршрутов в URL могут присутствовать переменные, например, `{{id}}`. Замените их на конкретные значения перед использованием

i Примечание На этой вкладке отображаются не только адреса, но и статус endpoint'а: ● — активен, ● — отключён. Адреса могут быть сгруппированы по средам (например, Dev, Test, Stage). Количество доступных сред зависит от конфигурации развёртывания системы.

Проксирование Kafka и топик для подключения

В блоке **Проксирование** отображаются текущие настройки проксирования для Kafka-топиков.



Проксирование

Описание действий

1. В навигационном дереве слева выберите требуемый Endpoint
2. Перейдите в раздел **“Проксирование”**
3. При необходимости включите или отключите проксирование
4. Напротив каждого настроенного стенда отображаются:
 - **Входящий топик для подключения** — топик, из которого Mimic принимает сообщения
 - **Топик для проксирования** — топик, в который сообщение будет отправлено, если включено проксирование и ни один из сценариев не сработал

i Примечание Для стендов, у которых не настроено подключение к Kafka, поля остаются пустыми. Чтобы указать адрес Kafka, перейдите в настройки проекта. Поля “Топик для подключения” и “Топик для проксирования” заполняются автоматически на основе данных, указанных в endpoint’е.

🔒 Права пользователей Создавать, копировать Endpoint’ы и управлять ими могут пользователи с ролью **“Администратор”**, **“Владелец”** проекта и **“Пользователь”**. **“Читатель”**- может только просматривать уже созданные Endpoint’ы.

См. также разделы

- [Про gRPC](#)
- [Введение в gRPC](#)

- Создание сценария
- Создание проекта
- Журнал событий

Сценарии

Сценарий - набор условий (фильтров), переменных и логики ответа, определяющий, какой ответ вернуть при поступлении запроса на конкретный endpoint. Сценарий срабатывает только если все его фильтры совпадают с данными входящего запроса.

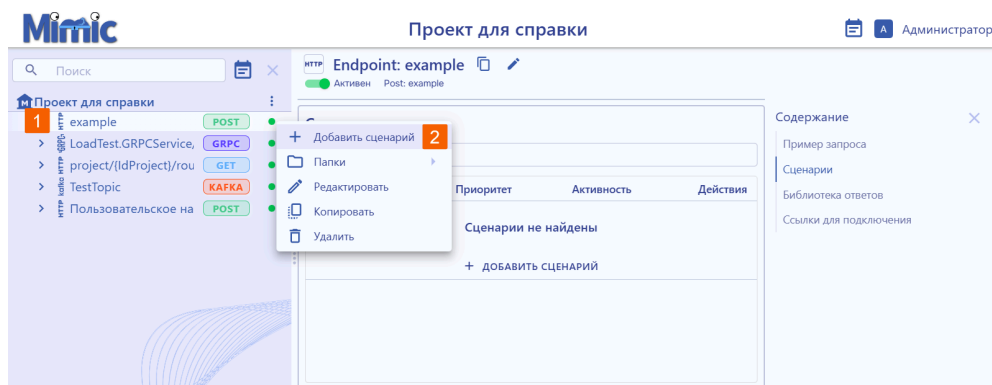
Создание сценария

Существует три способа создания сценария:

1. Создание с нуля

Доступно из дерева проекта или со страницы endpoint'a.

Из дерева:

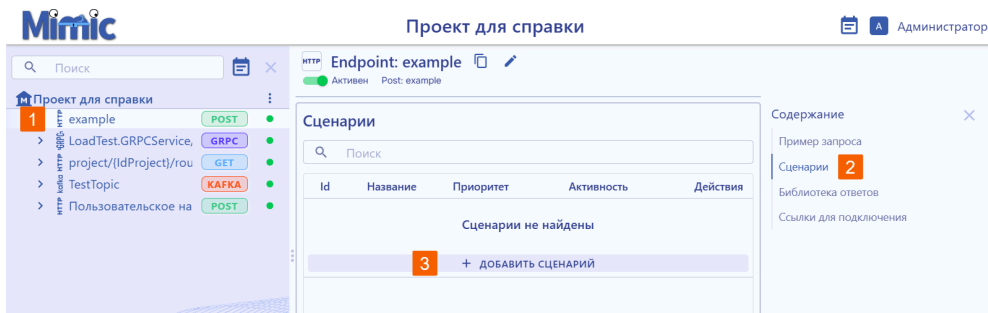


Создание через дерево

Описание действий

1. Вызовите контекстное меню endpoint'a (ПКМ или значок :)
2. Выберите **“+ Добавить сценарий”**
3. В дереве появится элемент **“Новый сценарий”**
4. Откройте его на редактирование через контекстное меню (ПКМ или значок :)

Со страницы endpoint'a:



Создание через Endpoint

Описание действий

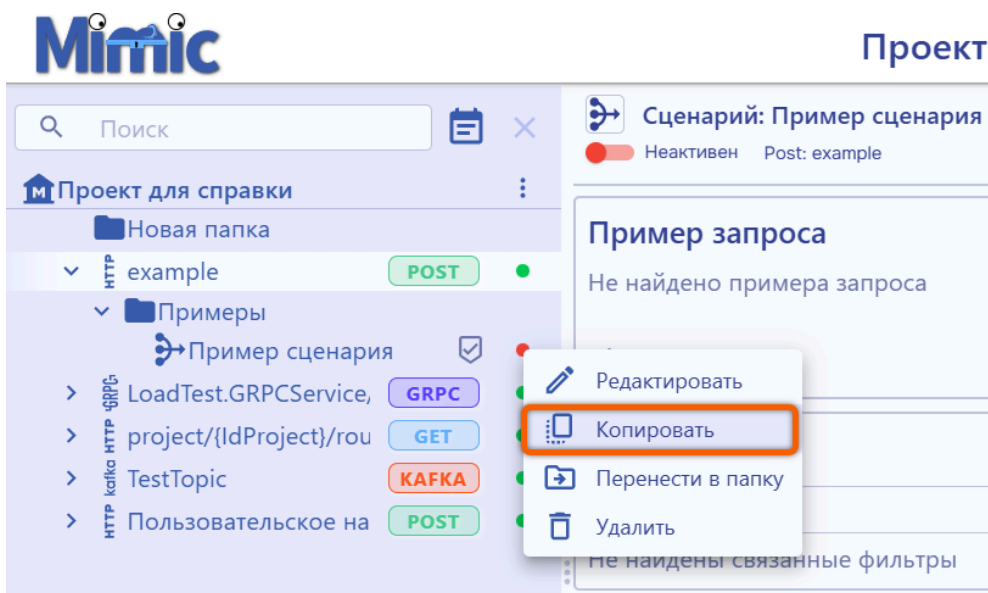
1. Перейдите на страницу нужного endpoint'a
2. Выберите раздел **“Сценарии”**
3. В разделе **“Сценарии”** нажмите **“+ Добавить сценарий”**

Во всех случаях открывается Конструктор сценария. Подробнее см. Конструктор сценария.

2. Копирование существующего сценария

Сценарий можно скопировать из дерева проекта или со страницы endpoint'a. Скопированный сценарий создаётся в неактивном состоянии и помещается в корень связанного endpoint'a. Копировать сценарии между Endpointами нельзя.

Из дерева:

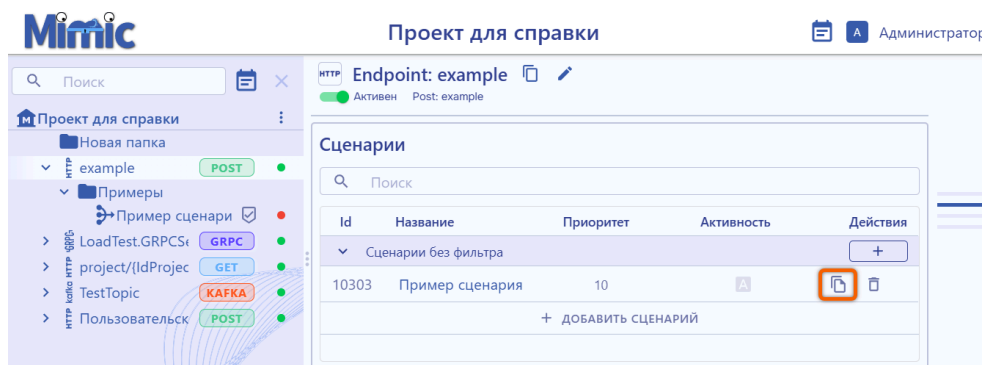


Копирование из дерева

Описание действий

1. Вызовите контекстное меню нужного сценария (ПКМ или значок :)
2. Выберите пункт **“Копировать”**

Со страницы endpoint'a:



Копирование через Endpoint

Описание действий

1. Перейдите на страницу endpoint'a и откройте раздел **“Сценарии”**
2. В таблице найдите сценарий для копирования
3. Нажмите кнопку **копировать**

После копирования в дереве появится новый элемент с названием: **<Исходное имя сценария> Копия** Например: “Проверка авторизации Копия”

3. На основе запроса из Журнала событий

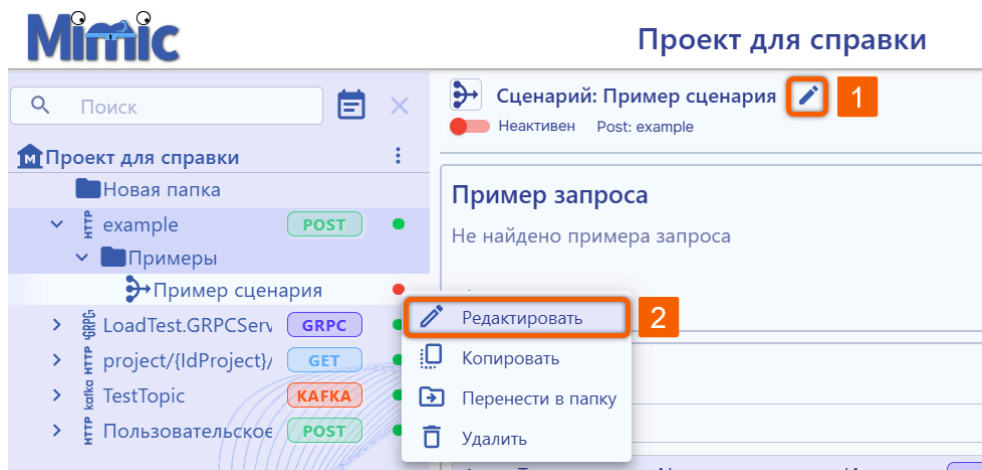
См. подробное описание в разделе [Журнал событий](#)

Редактирование сценария

Редактировать сценарий можно двумя способами:

1. **Через Конструктор сценариев** Конструктор предоставляет централизованный интерфейс для управления всеми элементами сценария: фильтрами, переменными, примером запроса и ответом.

Вызвать конструктор можно двумя способами:



Вызов конструктора из сценария

Описание

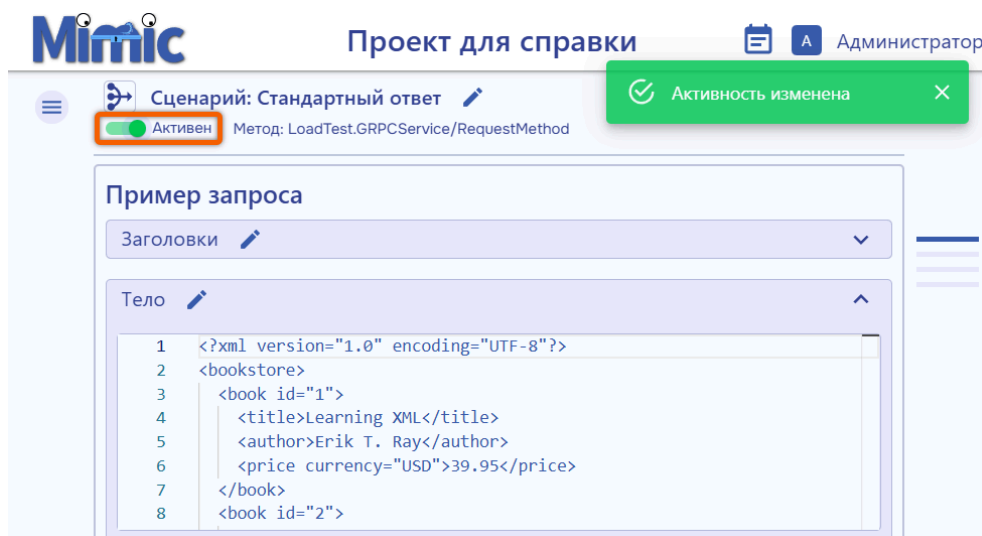
1. со страницы самого сценария — значок карандаша рядом с названием в верхней части страницы
2. из контекстного меню сценария в дереве проекта (ПКМ → “Редактировать”)

Подробнее о работе с конструктором см. Конструктор сценария.

2. Через отдельные разделы на странице сценария

На странице сценария можно быстро включить или отключить его с помощью переключателя активности. Неактивные сценарии не участвуют в обработке запросов, и ответы по ним не возвращаются.

Переключатель активности



Активность сценария

Компоненты сценария На той же странице доступны специализированные разделы для настройки отдельных элементов:

- **Пример запроса** — настройка входящих данных (подробнее)
- **Фильтры** — управление условиями срабатывания (подробнее)
- **Переменные** — определение извлекаемых значений (подробнее)
- **Ответы по сценарию** — конфигурация возвращаемого сообщения (подробнее)

Этот способ удобен, если нужно быстро внести изменения в один конкретный элемент без перехода в полный конструктор.

Такой подход даёт гибкость: вы можете либо настроить всё сразу в едином интерфейсе, либо работать с каждым блоком отдельно — в зависимости от задачи.

Пример запроса

Этот раздел содержит готовые примеры запросов к указанному Endpoint'у или сценарию, включая все необходимые элементы для корректного взаимодействия:

- **HTTP/gRPC** — заголовки, параметры и тело запроса если оно есть
- **Kafka** — шаблон сообщения для публикации в указанный топик, а также сопутствующие заголовки

Примеры запроса демонстрируют правильный формат данных, обеспечивающий успешное взаимодействие с API, и могут использоваться для:

- тестирования работы Endpoint
- валидации входящих запросов
- быстрой настройки новых сценариев

Пример запроса в endpoint'е и в сценарии используют одинаковый формат. Endpoint-пример выступает как шаблон: его можно импортировать в сценарий и адаптировать под конкретные фильтры или данные.

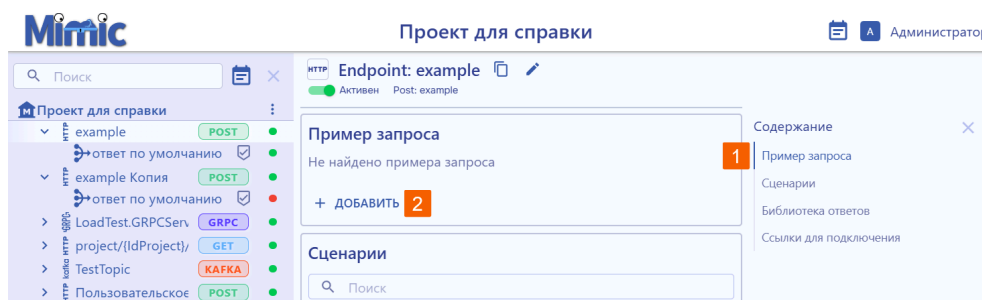
Создание примера запроса

Пример запроса можно создать двумя способами:

- **Вручную** — доступно для всех типов endpoint'ов и сценариев
- **Импортом** — доступно только для HTTP-маршрутов и сценариев

Вручную

Перейдите в нужный проект и выберите endpoint или сценарий, для которого хотите добавить пример:

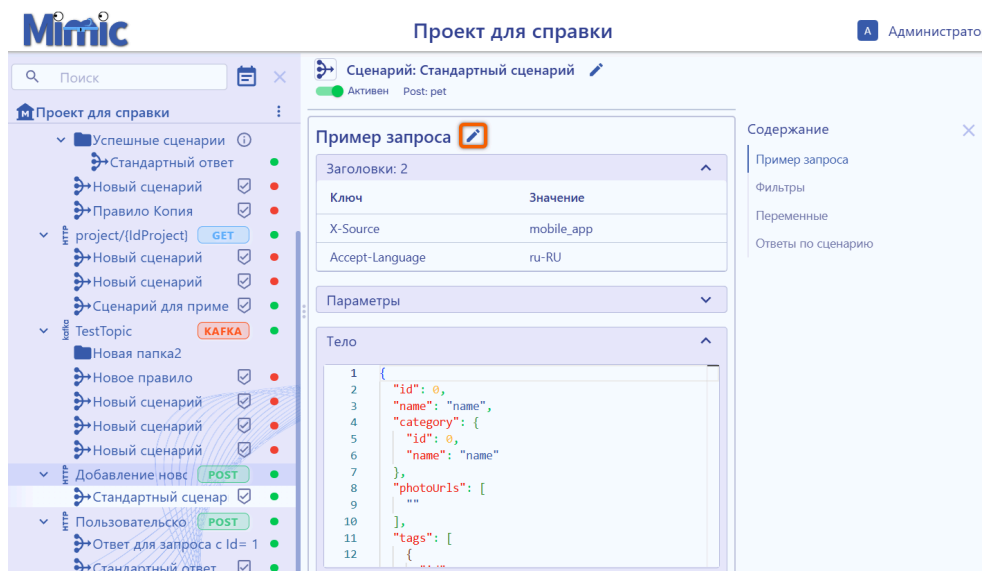


Пример_запроса_добавление

Описание действий

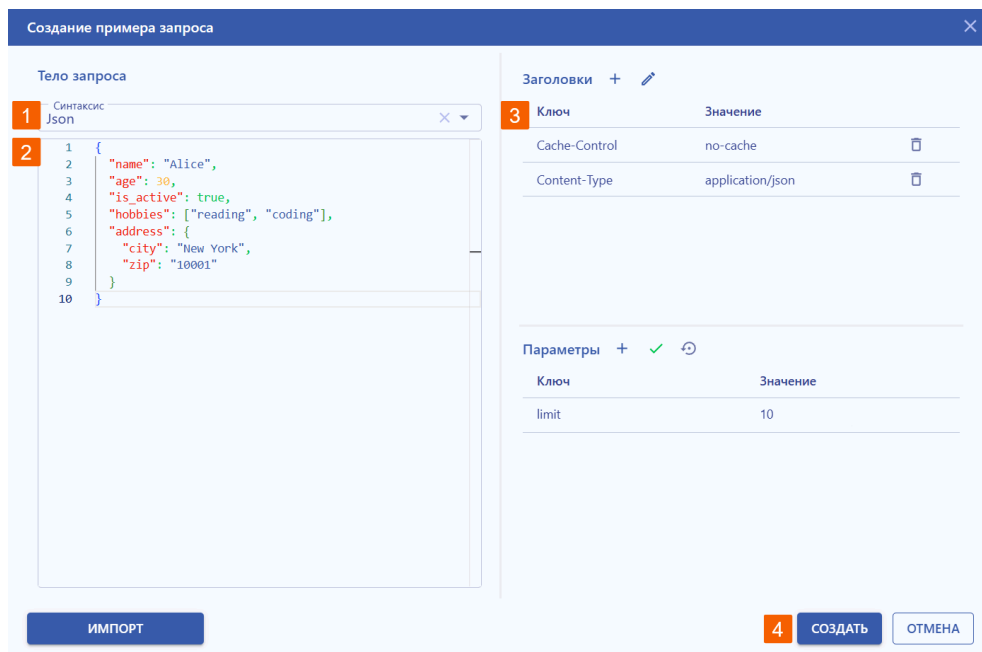
1. Перейдите в раздел **Пример запроса**
2. Если пример ещё не создан, отобразится кнопка **“+ Добавить”**

Если пример уже существует, будут доступны кнопки редактирования (см. изображение ниже).



Редактирование_примера_запроса

После нажатия **“+ Добавить”** откроется форма добавления/редактирования примера:



Добавление_примера_запроса

Описание действий

1. В поле **“Синтаксис”** выберите формат подсветки для тела запроса (Текст, JSON или XML) Это влияет только на отображение в интерфейсе
2. Ниже заполните тело запроса, если оно требуется для данного типа запроса.
3. Также заполните метаданные такие как **“Заголовки”**(Headers) и **“Параметры”** Для добавления нового заголовка или параметра нажмите иконку плюса **+** Для редактирования существующего — нажмите иконку **карандаша** Список перейдёт в режим редактирования. После ввода всех данных нажмите:
 - Для подтверждения изменения
 - для восстановления всех значений блока к предыдущим значениям
4. Подтвердите внесённые изменения нажав кнопку **“Сохранить”**

Редактирование примера запроса

Форма редактирования примера запроса поддерживает режимы различные режимы отображения в зависимости от того, какой сценарий выполняет пользователь:

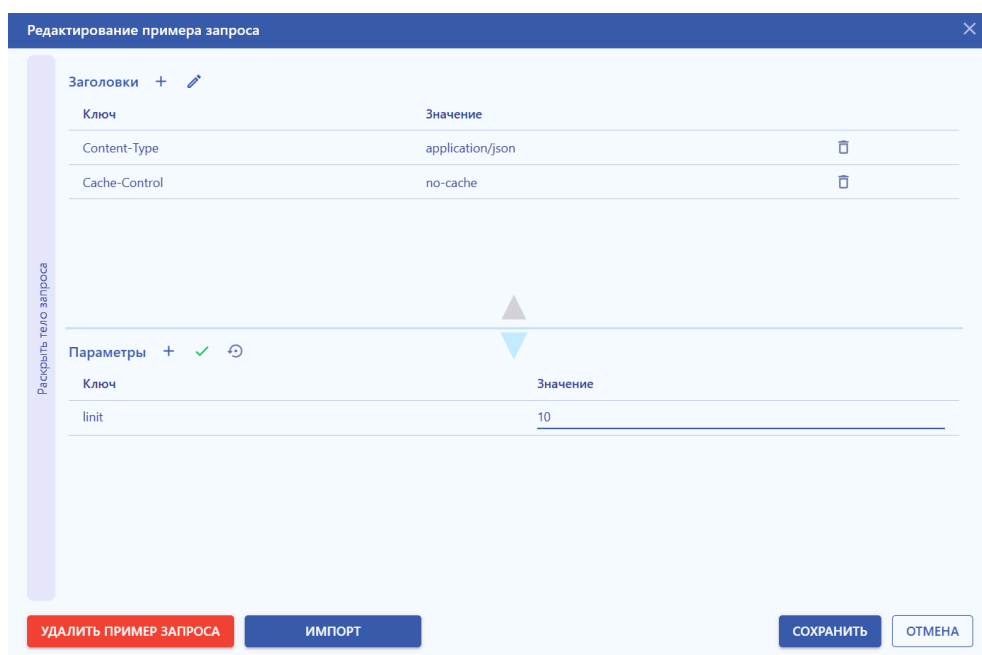
- При редактировании **заголовков** или **параметров** можно отобразить только блок метаданных
- При редактировании **тела запроса** только блок с телом

Вы можете в любой момент переключиться между режимами или настроить отображение по своему усмотрению:

- Перетащите разделительную полосу по центру экрана, чтобы изменить соотношение видимых блоков (тело / метаданные)
- Чтобы временно скрыть один из блоков, перетащите разделитель до упора или нажмите на стрелку которая появляется при наведении на разделительную линию
- Скрытый блок можно вернуть в любое время с помощью вертикальной кнопки **Раскрыть тело запроса** или **Раскрыть метаданные**, которая появляется на границе области редактирования

Это позволяет адаптировать рабочее пространство под текущую задачу — будь то настройка большого JSON-тела или массовое редактирование заголовков.

Отображение формы редактирования запроса только с Метаданными

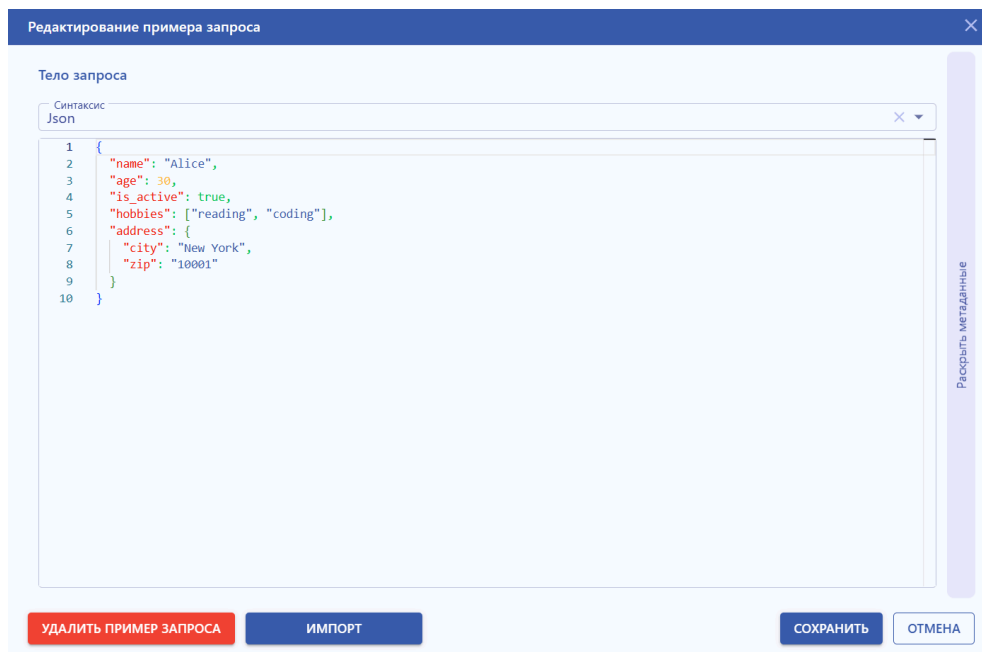


Ключ	Значение
Content-Type	application/json
Cache-Control	no-cache

Ключ	Значение
limit	10

Метаданными

Отображение формы редактирования запроса только с Телом запроса





Телом запроса

Описание действий


1. В поле **“Синтаксис”** можно выбрать формат подсветки для тела запроса: Текст, JSON или XML. Это влияет только на отображение в интерфейсе поля тела зароса
2. При необходимости можете раскрыть блок с **метаданными** слева

В разделе **Пример запроса** для endpoint Kafka содержит аналогичные поля как и для других endpoint. в разделе **Параметров** содержит только поле **“Ключ”**


Пример запроса 

Заголовки: 1 

Ключ	Значение
ExampleHeader	ExampleValue

Параметры: 1 


Ключ
Key

Тело 


1 Тело запроса из топика Kafka

Пример_запроса_Kafka_Endpoint

Пример редактирования:


Редактирование примера запроса 

Тело запроса


Синтаксис
Текст 


1 Тело запроса из топика Kafka

УДАЛИТЬ ПРИМЕР ЗАПРОСА

Топик для подключения Ключ 

Mimic17035_TestTopic Key

Заголовки + 

Ключ	Значение
ExampleHeader	ExampleValue 

СОХРАНИТЬ ОТМЕНА

Пример_запроса_Kafka

Описание действий

1. В поле **“Синтаксис”** можете выбрать формат подсветки для тела запроса (Текст, JSON или XML) Это влияет только на отображение в интерфейсе поля тела зароса
2. В блоке с **метаданными** можно отредактировать только **“Ключ”** или **“Заголовки”**

Примечание Поле “Топик для подключения” формируется автоматически на основе данных Endpoint’a

Импорт примера запроса

Для сценариев: Импорт примера запроса для сценария доступен только из endpoint’a. В этом случае форма примера запроса автоматически заполняется данными из связанного endpoint’a. После импорта значения можно отредактировать и сохранить.

Примечание Примеры запроса в сценариях и endpoint’ах не связаны между собой — каждый можно редактировать независимо.

Для endpoint: Импорт из спецификации OpenAPI поддерживается только для HTTP-маршрутов.

Для выполнения импорта, откройте форму добавления/редактирования примера запроса

Пример_запроса_импорт

Описание действий

1. Нажмите кнопку “**Импорт**” и выберите пункт “**Пример запроса**”
2. Остальные действия аналогичны на загрузку Endpoint. Подробнее можно ознакомиться в разделе импорт

Фильтры

Фильтры - это сущность, которая используется для отбора и проверки значений во входящих запросах. Они позволяют определить условия, которые должны быть выполнены, чтобы сработало определенный сценарий.

В одном сценарии может быть неограниченное количество фильтров. Все фильтры объединены логическим И: сценарий срабатывает только если все условия выполнены одновременно, и в этом случае возвращается связанный с ним ответ.

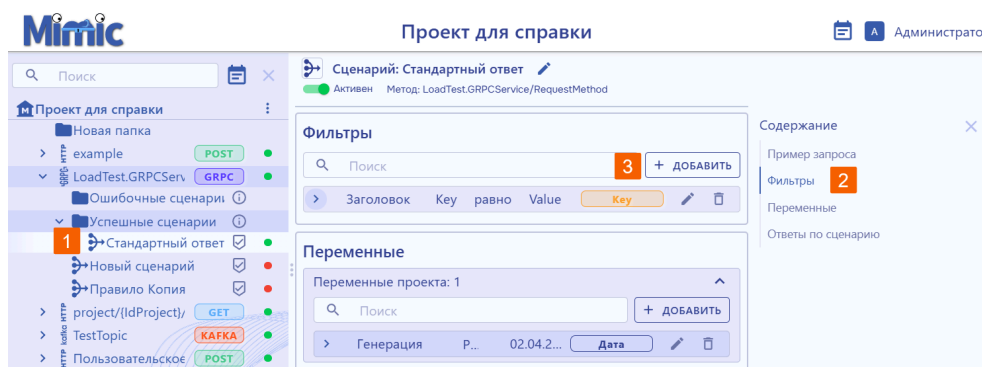
Если у сценария нет фильтров, он считается сценарием по умолчанию. Такие сценарии используются, когда ни один другой (с фильтрами) не подошёл.

⚠ Важно Если активных сценариев без фильтров несколько, система выбирает один из них: сначала — по наименьшему значению приоритета, при равных приоритетах — первый, отобранный внутренней логикой системы (порядок не гарантируется). Рекомендуется задавать явные приоритеты, чтобы избежать неопределённого поведения.

Создание фильтров

Вызов формы создания фильтров доступен двумя способами

Со страницы сценария



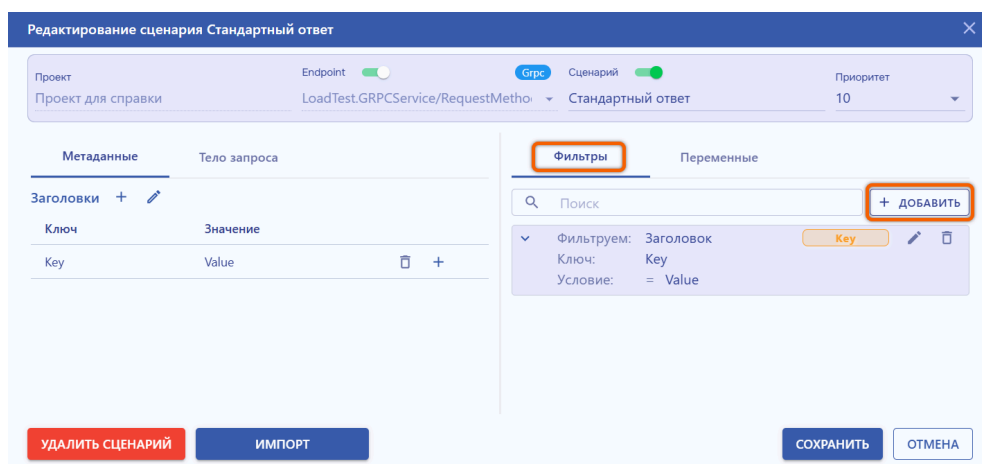
Добавить фильтр из сценария

Описание действий

1. Выберите нужный сценарий в дереве проекта
2. Перейдите в раздел “**Фильтры**”

3. Нажмите кнопку “+ Добавить”

Из конструктора сценария При редактировании сценария через конструктор сценария по кнопке “+ Добавить” на вкладке “Фильтры”



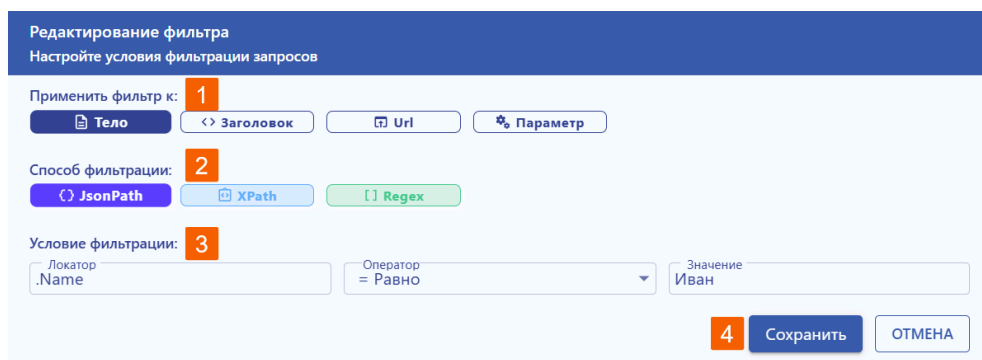
Добавить фильтр из конструктора

Оба варианта вызывают одну и ту же форму создания фильтра. Конкретный способ использования определяется вашим рабочим контекстом. Подробное описание полей формы приведено далее по каждому типу фильтра.

Типы фильтров

1. Фильтр по телу запроса

Применяется к содержимому тела запроса (JSON, XML, plain text)



Фильтр по телу

Описание

1. Выберете источник для фильтрации данных: “Тело”;

2. Укажите **Способ фильтрации**: **JsonPath**, **XPath** или **Regex**

3. После выбора отобразятся соответствующие поля для настройки условия:

- **Для JsonPath — отбор из JSON.**
 - **Локатор**- путь в формате JsonPath \$(например, `.user.name`)
 - **Оператор** - Равно / Не равно
 - **Значение**- ожидаемое значение (например, "Иван") [Проверить JsonPath](#)
- **Для XPath — отбор из XML.**
 - **Локатор**- путь в формате XPath (например, `//user/name`)
 - **Оператор**- Равно / Не равно
 - **Значение**- ожидаемое значение (например, "Иван") [Проверить XPath](#)
- **Для Regex - отбор из любого текста**
 - **Выражение** - регулярное выражение (например, "--")
 - **Оператор**- Равно / Не равно
 - **Значение**- строка, которая должна быть найдена по указанному выражению (например, "123-45-6789") [Проверить Regex](#)

4. Нажмите **“Сохранить”**, чтобы создать фильтр

i Примечание Фильтр считается совпавшим, если результат применения регулярного выражения или отбора локатора к телу запроса точно соответствует указанному значению (при операторе «Равно»)

2. Фильтр по заголовкам

Применяется к HTTP-заголовкам запроса.

Редактирование фильтра
Настройте условия фильтрации запросов

Применить фильтр к: 1
Тело <> **Заголовок** Uri Параметр

Способ фильтрации: 2
Key Regex

Условие фильтрации: 3
Ключ: Server Оператор: = Равно Значение: nginx

4 Сохранить ОТМЕНА

Фильтр по заголовку

Описание

1. Выберите источник для фильтрации данных: **“Заголовок”**;
2. Укажите **Способ фильтрации: Key** или **Regex**
3. После выбора отобразятся соответствующие поля для настройки условия:

- **Для Key — точное совпадение по имени и значению заголовка .**
 - **Ключ**- имя HTTP-заголовка (например, “Server”)
 - **Оператор** - Равно / Не равно
 - **Значение**- ожидаемое значение (например, “nginx”)
- **Для Regex - поиск по имени заголовка** с использованием регулярного выражения
 - **Выражение** - регулярное выражение (например, “--”)
 - **Оператор**- Равно / Не равно
 - **Значение**- строка, которая должна быть найдена по указанному выражению (например, “123-45-6789”) Проверить Regex

4. Нажмите **“Сохранить”**, чтобы создать фильтр

3. Фильтр по переменной из адреса обращения

Используется, когда значения передаются в самом пути запроса, например: /project/158/route/56 Такие переменные в EntryPoint должны быть определены как: project/{{{IdProject}}}/route/{{{IdRoute}}}

Редактирование фильтра
Настройте условия фильтрации запросов

Применить фильтр к: 1
Тело <> Заголовок **Url** Параметр

Способ фильтрации: 2
Key Regex

Условие фильтрации: 3
Ключ: ProjectID Оператор: = Равно Значение: 12

4 Сохранить ОТМЕНА

Фильтр по адресу

Описание

1. Выберите источник для фильтрации данных: **“Url”**;
2. Укажите **Способ фильтрации: Key** или **Regex**
3. После выбора отобразятся соответствующие поля для настройки условия:

- **Для Key — точное совпадение по имени и значению заголовка.**

- **Ключ** - имя переменной из URL (выбор из списка доступных значений)
- **Оператор** - Равно / Не равно
- **Значение**- ожидаемое значение (например, “123”)
- **Для Regex - поиск по имени заголовка** с использованием регулярного выражения
 - **Выражение** - регулярное выражение (например, “--”)
 - **Оператор**- Равно / Не равно
 - **Значение**- строка, которая должна быть найдена по указанному выражению (например, “123-45-6789”) Проверить Regex

4. Нажмите “**Сохранить**”, чтобы создать фильтр

4. Фильтр по параметрам входящего запроса

Редактирование фильтра
Настройте условия фильтрации запросов

Применить фильтр к: 1

Способ фильтрации: 2

Условие фильтрации: 3
 Выражение:
 Оператор:
 Значение:

4

Фильтр по параметру

Параметры в url-запросе - это то, что передаётся после “?”. Например:

`https://host.ru/MethodName?Parameter1=value1&Parameter2=value2`


Alt text

Описание

1. Выберете источник для фильтрации данных: “**Параметр**”;
2. Укажите **Способ фильтрации**: **Key** или **Regex**
3. После выбора отобразятся соответствующие поля для настройки условия:
 - **Для Key** — **точное совпадение по имени и значению параметра.**
 - **Ключ** - имя параметра для отбора значения (например, user_id)
 - **Оператор** - Равно / Не равно
 - **Значение**- ожидаемое значение (например, “123”)

- **Для Regex - поиск по имени параметра** с использованием регулярного выражения
 - **Выражение** - регулярное выражение (например, "--")
 - **Оператор**- Равно / Не равно
 - **Значение**- строка, которая должна быть найдена по указанному выражению (например, "123-45-6789") Проверить Regex

4. Нажмите **“Сохранить”**, чтобы создать фильтр

 **Права пользователей** Создавать фильтры и управлять ими могут “Владелец” проекта и “Пользователь”, просматривать - “Читатель”.

Переменные

Переменные — это именованные значения, которые позволяют делать ответы сценариев динамическими и гибкими. С их помощью можно возвращать сгенерированные данные (например, ФИО, GUID, дату, инкрементируемый ID) или извлекать информацию из входящего запроса и подставлять её в ответ.

Типы переменных

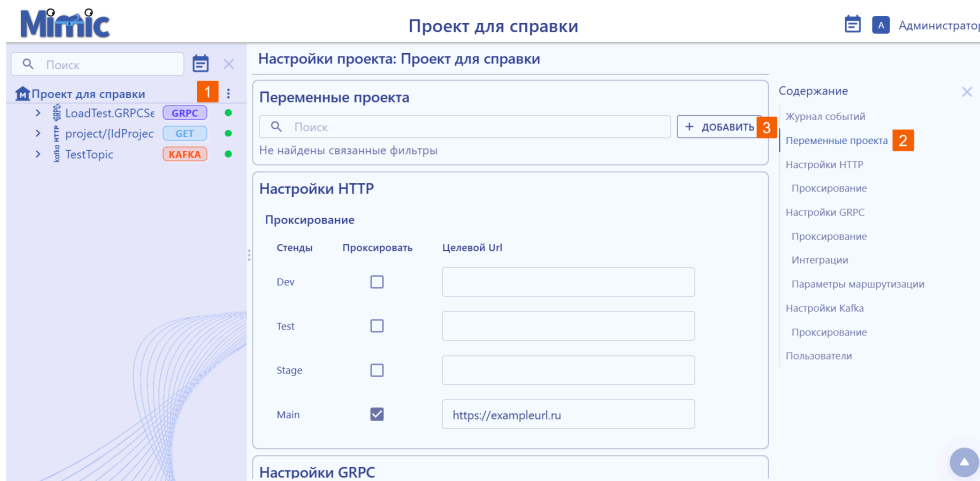
- **Переменные проекта** Доступны во всех endpoint и сценариях проекта. Используются для общих значений, например, токенов, префиксов или глобальных счётчиков
- **Переменные URL** Извлекаются из параметров пути или query-строки. Например, в адресе `http://mimic.local/project/{id}` переменная `id` автоматически становится доступной в сценарии
- **Переменные сценария** Привязаны к конкретному сценарию. Их имена и значения доступны только в рамках этого сценария и могут использоваться для формирования динамического или персонализированного ответа

Создание переменной

Переменные можно создать из трёх мест в интерфейсе Mimic:

Переменные проекта

Создаются в разделе Настройки проекта:



Переменные_проекта

Описание действий:

1. Нажмите на имя проекта в левом дереве навигации
2. Выберите пункт **“Переменные проекта”**
3. Нажмите на кнопку **“+ Добавить”**

Эти переменные доступны во всех маршрутах и сценариях проекта, к ним автоматически добавляется префикс **“P:”**

Переменные URL

Создаются автоматически при определении пути маршрута (endpoint). Пример: В запросе `https://host.ru/project/158/route/56` значения 158 и 56 передаются как параметры пути:

- `IdProject = 158`
- `IdRoute = 56`

Чтобы использовать их, при создании Endpoint укажите путь с переменными: `project/{{IdProject}}/route/{{IdRoute}}` После этого в любом сценарии этого Endpoint, в разделе **“Переменные”**, в блоке **“Переменные URL”** появятся автоматически созданные переменные с префиксом **“E:”**

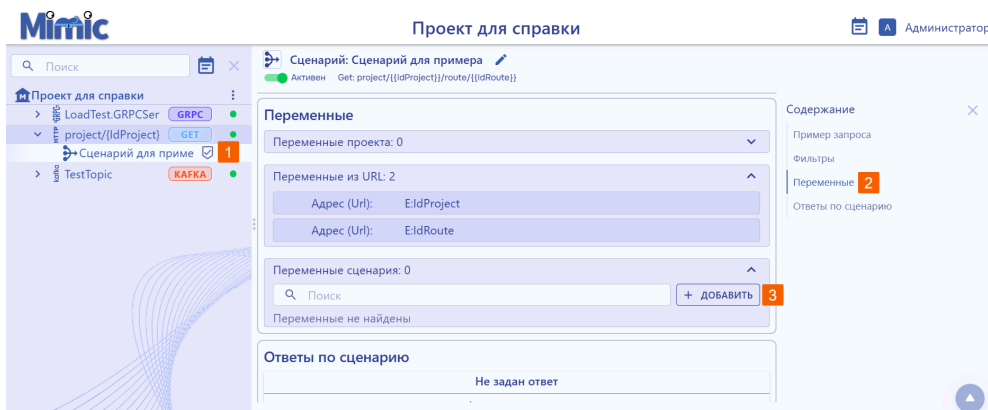
- `E:IdProject`

- *E:IdRoute*

В эти переменные будут отбираться значения, переданные в реальном запросе.

Переменные сценария

Переменные сценария создаются непосредственно внутри конкретного сценария и действуют только в его рамках.



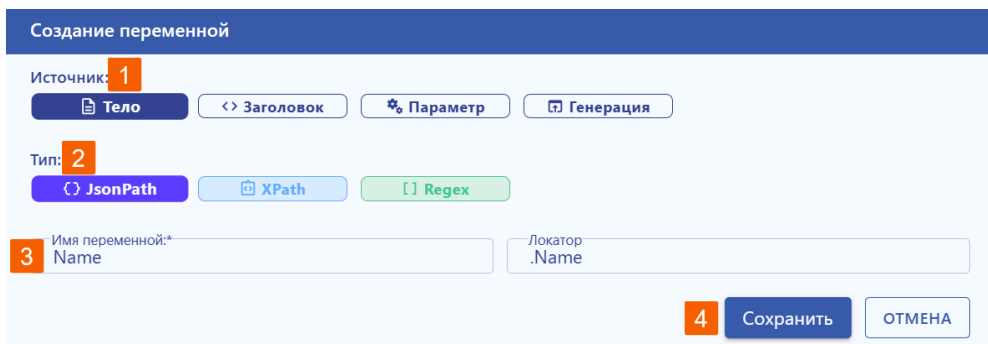
Переменные_сценария

Описание действий:

1. Откройте нужный сценарий
2. Перейдите в раздел “Переменные”
3. Нажмите на кнопку “+ Добавить”

Эти переменные действуют только в рамках этого сценария.

Форма “Создание переменной”



Переменные_сценария

1. Выберите “Источник” — откуда будет браться значение:

- **Тело** — из тела входящего запроса
- **Заголовок** — из HTTP-заголовка запроса
- **Параметр** — параметры из адреса например (*?user=123*)
- **Генерация** — значение создаётся автоматически по заданным правилам

2. После выбора источника отобразятся доступные типы извлечения или генерации Допустимые комбинации:

- *Для «Тело»:*
 - JsonPath
 - [Описание синтаксиса](#)
 - [Проверка отбора значения JsonPath](#)
 - XPath
 - [Описание синтаксиса](#)
 - [Проверка отбора значения XPath](#)
 - Regex (Регулярное выражение)
 - [Проверка отбора значения Regex](#)
- *Для «Заголовок» и «Параметр»:*
 - Key - извлечение по имени ключа
 - Regex (Регулярное выражение)
- *Для «Генерация»:*
 - Текст
 - Кириллица (случайная строка заданной длины)
 - Латиница (случайная строка из латинских символов)
 - Число +1 (инкрементируемое значение)
 - Число (случайное число с заданным количеством разрядов)
 - GUID (уникальный идентификатор)
 - Дата
 - ФИО

3. После выбора типа появятся соответствующие поля:

- **“Имя переменной”** Укажите уникальное имя:
 - для переменных сценария — в рамках сценария,
 - для переменных проекта — в рамках проекта
- Для динамических источников (Тело, Заголовок, Параметр)

- **“Локатор”** - используется при выборе JSONPath или XPath. Пример: \$.user.id (для JSON), /user/name (для XML)
 - **“Выражение”** - указывается при выборе Regex. Пример: .[0-1][0-9]
 - **“Ключ”** - отображается при выборе Key. Укажите имя заголовка или параметра, значение которого нужно извлечь. Пример: X-Request-ID, user
- Для источника **“Генерация”** отображаются дополнительные настройки в зависимости от типа:
 - Текст **“Пользовательский текст”**- многострочное значение, которое будет подставляться в ответ
 - Кириллица/Латиница **“Длина строки”** - количество символов в генерируемой строке (для каждого запроса — новое значение)
 - Число +1 **“Текущее значение для инкрементации”** - число, которое будет возвращено при следующем запросе. После использования оно увеличится на 1
 - Число **“Длина числа”** - указывается количество разрядов случайного числа
 - GUID - генерируется уникальный идентификатор при каждом срабатывании сценария в формате: `6F9619FF-8B86-D011-B42D-00CF4FC964FF`
 - Дата - позволяет создать дату Пример формы для генерации даты:

Создание переменной

Источник:

Тело
<> Заголовок
⚙️ Параметр
Генерация

Тип:

Тр Текст
A+ Кириллица
A+ Латиница
📄 Число + 1
📅 Число

<> GUID
Дата
👤 ФИО

Имя переменной:*

Сгенерированная дата 4

Системная Вручную

Дата 1

Время

Формат даты 2

Смещение даты

Год	Месяц	День
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Час	Минуты	Секунды
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

3

Сохранить
ОТМЕНА

Генератор_даты

1. **Режим генерации** Выберите один из двух вариантов:

- **Системная** - текущая дата и время по UTC будут вычисляться динамически для каждого запроса
- **Вручную** - используется фиксированная дата и время, заданная один раз (одинаковая для всех запросов)

2. **Формат даты** В поле “Формат даты” укажите желаемый шаблон вывода.
Примеры:

- *dd.MM.yyyy* → *05.04.2026*
- *yyyy-MM-dd'T'HH:mm:ss* → *2026-04-05T14:30:00*

3. **Смещение даты** В блоке “Смещение даты” задайте сдвиг относительно базовой даты:

- годы, месяцы, дни, часы, минуты, секунды
- поддерживаются отрицательные значения (например, -1 день для получения «вчера»)

4. **Сгенерированная дата** В поле “Сгенерированная дата” отображается пример значения, которое будет подставляться в переменную при обработке запроса

• **ФИО** Пример формы для генерации ФИО:

Создание переменной

Источник:

Тип:

Имя переменной:*
FIO

Сгенерированное ФИО
Петрова А. И. 2

1 Генерировать Фамилию Женщина
 Генерировать Имя Фамилия + Инициалы
 Генерировать Отчество

Генератор_ФИО

1. Настройте параметры генерации:


- Включить/исключить фамилию, имя, отчество

- Выбрать пол
- Указать формат: полное ФИО или инициалы (например, “Иванов И.И.”)

2. В поле **“Сгенерированное ФИО”** отображается результат, который будет использоваться в ответе

4. Нажмите **“Сохранить”**

5. При необходимости переменную можно удалить, нажав на значок корзины

 **Права пользователей “Владелец”** проекта и **“Пользователь”** могут создавать, изменять и удалять переменные проекта. **“Читатель”** имеет доступ только на просмотр.

См. также разделы

- Настройки проекта

Конструктор сценариев

Конструктор сценариев — единый рабочий интерфейс, в котором можно создать сценарий, привязать к нему пример входящего запроса и настроить фильтры (условия срабатывания) и переменные (извлекаемые значения) на основе этого примера.

Структура формы: Интерфейс разделён на две части:

Левая панель	Правая панель
Пример запроса: заголовки, параметры, тело	Настройки сценария: фильтры, переменные

Пример запроса заполняется либо вручную при создании сценария с нуля, либо автоматически — если форма вызвана из Журнала событий на основе реального входящего запроса.

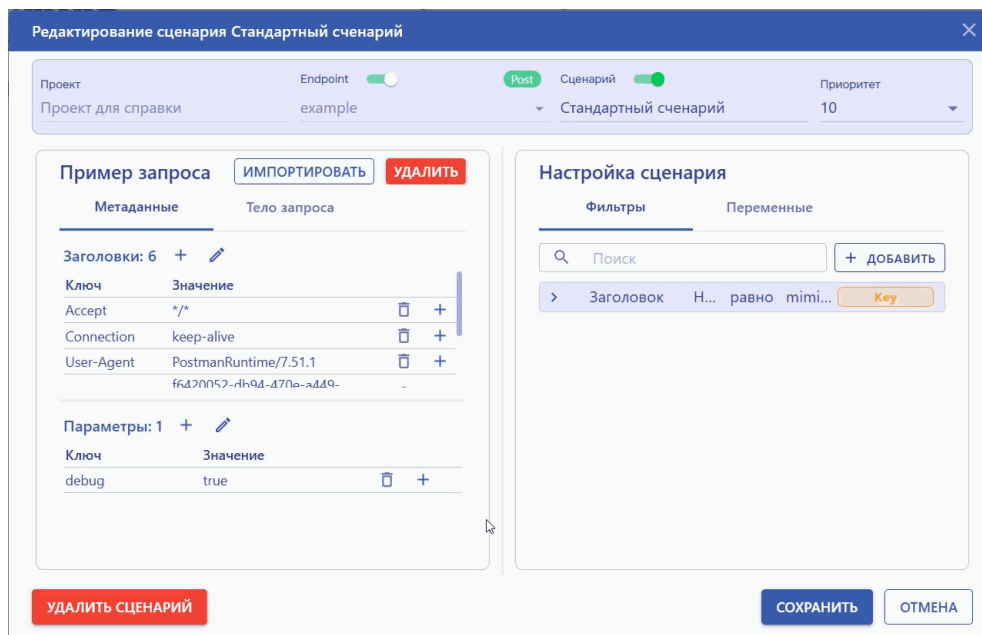
Данный функционал позволяет создавать сценарий как с нуля так и используя реальные запросы в качестве основы (Вызов из журнала событий).

Режимы отображения

Форма конструктора открывается в одном из трёх режимов в зависимости от контекста вызова:

Режим	Описание
Редактирование примера запроса	Отображается только левая панель: вкладки “Метаданные” и “Тело запроса” . Используется при редактировании примера со страницы сценария.
Редактор фильтров/переменных	Отображаются только соответствующие блоки в правой панели. Активируется при редактировании этих элементов из настроек сценария.
Создание/редактирование сценария	Полный интерфейс, отображается и пример запроса, и настройки. Используется при: – создании сценария из “Журнала событий” – редактировании сценария или создания сценария с нуля

Варианты отображения формы в разных режимах



Режимы отображения

Примечание В любой момент вы можете переключиться между режимами, отобразив или скрыв нужные блоки интерфейса для удобства работы.

Созадине сценария через конструктор

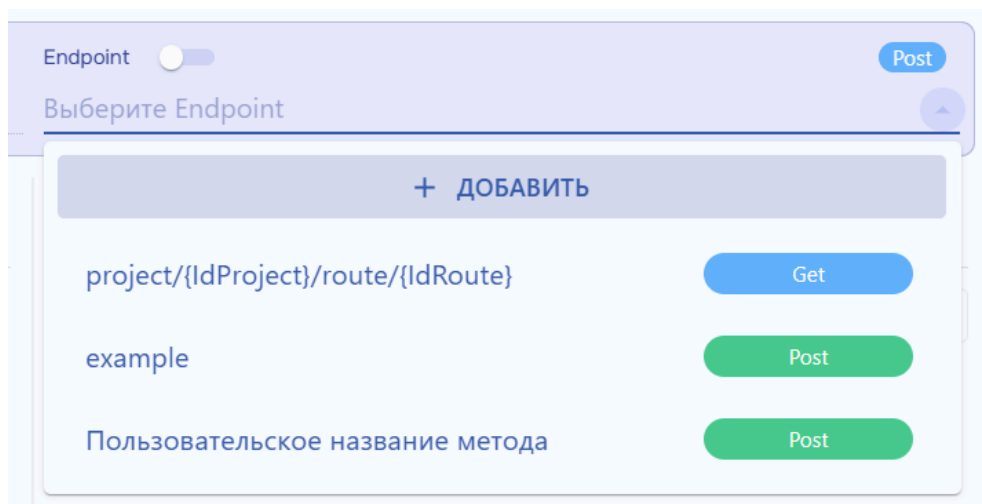
В верхней части окна конструктора заполняются ключевые поля:



Создать сценарий

Описание

- Endpoint** При редактировании сценария заполняется автоматически. При вызове конструктора из **“Журнала событий”** может быть не указан. В этом случае:
 - выберите существующий endpoint из списка
 - нажмите кнопку **“+ Добавить”**, чтобы создать новый



Создать сценарий

2. **Сценарий** Задайте понятное и уникальное имя для быстрой идентификации в интерфейсе (например, “Ошибка 404 при отсутствии пользователя”).
3. **Приоритет** (опционально) Числовое значение, определяющее порядок обработки сценариев. При совпадении условий нескольких сценариев будет применён тот, у которого меньший приоритет (например, 1 имеет преимущество перед 2)

Переключатели активности Для endpoint’a: отображается в режиме только для чтения и носит информационный характер (показывает, активен ли сам endpoint).
Для сценария: позволяет включить или отключить его. По умолчанию все новые сценарии создаются активными.

Если сценарий не требует фильтров или переменных, его можно сразу сохранить, и перейти к настройке ответа. (см. раздел Ответные сообщения) В противном случае — переходите к настройке условий.

Быстрое создание фильтров и переменных

Помимо ручного заполнения полей, в Конструкторе сценариев доступен упрощённый способ создания фильтров и переменных — на основе данных из примера запроса.

i Примечание Описание ручного заполнения можно найти в соответствующих статьях: [Фильтры](#) и [Переменные](#).

Как это работает:

1. В правой панели переключитесь на нужную вкладку:
 - “**Фильтры**” — чтобы создать условие срабатывания,
 - “**Переменные**” — чтобы извлечь значение для использования в ответе.
2. В левой панели выберите источник данных:
 - “**Метаданные**” — содержит заголовки и параметры URL,
 - “**Тело запроса**” — отображает содержимое тела (JSON, XML и др.).
3. Нажмите кнопку **+**, расположенную рядом с нужным элементом (заголовок, параметр или поле в теле).

Фильтр или переменная будет создана автоматически на основе выбранного элемента.

Особенности автоматического заполнения: Для заголовков и параметров: создаётся правило типа “Key”, где:

- **Наименование** = имя заголовка/параметра
- **Значение** = текущее значение из примера запроса Для тела запроса:
- в **JSON** генерируется локатор в формате JsonPath (например, \$.user.name),
- в **XML** — выражение XPath (например, //user/name),
- **Значение** берётся из соответствующего тега или поля.

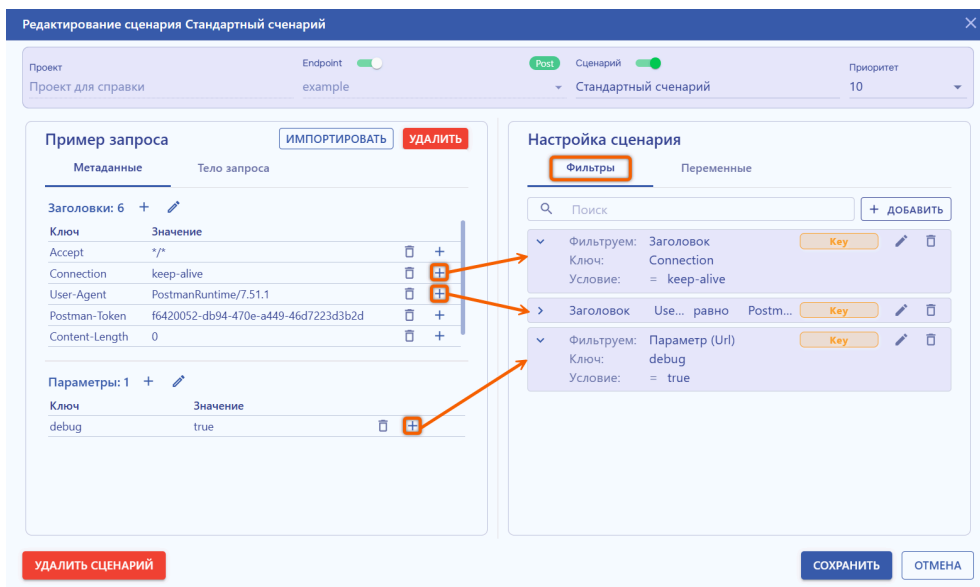
Все автоматически созданные объекты можно в любой момент отредактировать вручную.

Такой подход позволяет быстро настраивать сценарии, используя реальные или шаблонные данные в качестве основы, и при этом сохраняет полный контроль над конечной конфигурацией.

Примеры заполнения из метаданных

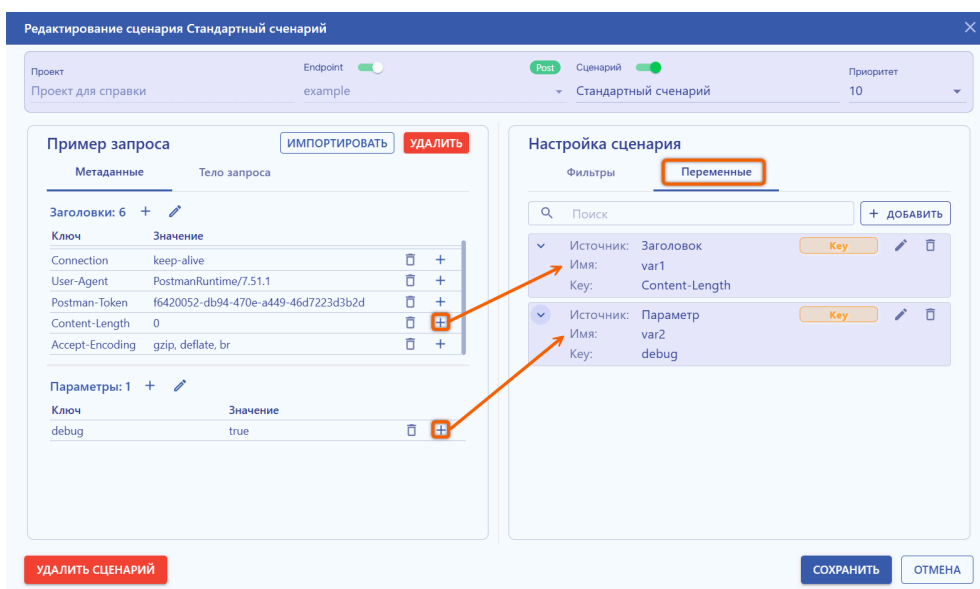
На вкладке “**Метаданные**” можно выбрать нужный заголовок или параметр и, нажав значок “**+**”, добавить его либо как фильтр, либо как переменную для будущего сценария — в зависимости от выбранной вкладки в правой части окна (“**Фильтр**” или “**Переменные**”), как показано на рисунке ниже:

Создание фильтров



Конструктор_создание_фильтра_из_метаданных

Создание переменных

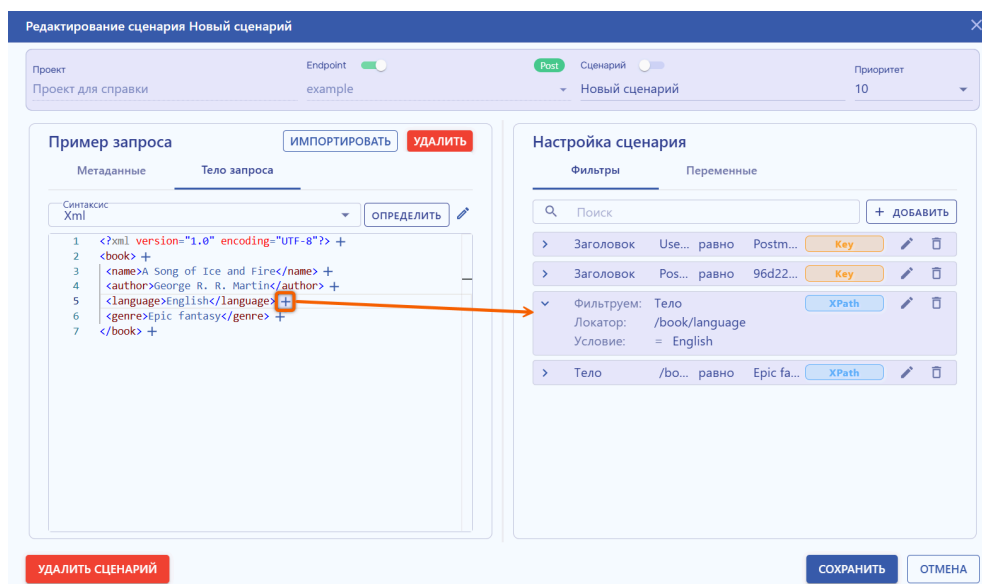


Конструктор_создание_переменных_из_метаданных

И **Примечание** При необходимости значение заголовков или параметров можно изменить — для этого нажмите кнопку карандаша у нужного объекта.

Пример заполнения из тела запроса

Аналогичным образом можно создать фильтр или переменную на основе данных из тела запроса: **Создание фильтров из тела**



Конструктор_создание_переменных_из_метаданных

Примечание Тело запроса можно отредактировать с помощью кнопки карандаша в верхней части экрана.

После нажатия одной из кнопок создания, в сценарий автоматически добавится в выбранный маршрут указанного проекта.

Примечания Все изменения, внесенные пользователем вручную в левой части окна относятся к примеру запроса и автоматически сохраняются и отражаются в разделе “Пример запроса” после сохранения сценария.

Ответные сообщения

Ответ- настраиваемая структура данных, возвращаемая при срабатывании сценария.сценария. Он привязан к конкретному endpoint'у и может использоваться в нескольких сценариях одного маршрута (HTTP/gRPC/Kafka).

Содержимое и формат зависят от протокола: **HTTP/HTTPS**

- **Заголовки:** произвольные пары “ключ–значение” (например, Content-Type, X-Request-ID)
- **Код ответа:** HTTP-статус (например, 200, 404, 500)
- **Задержка:** время в секундах перед отправкой ответа
- **Тело ответа:** текст в любом формате (JSON, XML, plain text)

gRPC

- **Метаданные:** дополнительные заголовки (аналог HTTP-заголовков)
- **Код ответа:** стандартный gRPC-статус (например, OK, NOT_FOUND, INTERNAL)
- **Задержка:** время в секундах перед отправкой ответа
- **Тело ответа:** данные в формате JSON, соответствующие структуре из .proto-контракта

Kafka

- **Заголовки:** дополнительные метаданные (ключи–значения)
- **Топик для ответа:** Kafka-топик, в который будет отправлено сообщение
- **Ключ:** значение ключа сообщения
- **Задержка:** время в секундах перед отправкой ответа
- **Значение:** основное содержимое сообщения

i Примечание Ответ применяется только если выполнены все условия из фильтров сценария. Подробнее о фильтрах см. Сценарий.

Ответ можно создать или отредактировать как из страницы сценария, так и из Библиотеки ответов. Все изменения синхронизируются автоматически.

Библиотека ответов

Доступна на странице Endpoint Здесь отображаются все ответы, созданные для выбранного endpoint'a. Любой ответ может использоваться в нескольких сценариях.

Project for reference

Endpoint: Пользовательское название метода
Активен Post: example/object/{[id]}/edit

Библиотека ответов

Id	Название	Код ответа	Тело ответа	Используется в сценар...
▼ Без связи				
13280	Ответ черновик	200-Ok		
▼ Используются в сценариях				
13279	С Текущей датой	200-Ok	{{P.Data}}	1 сценарие

+ ДОБАВИТЬ ОТВЕТ

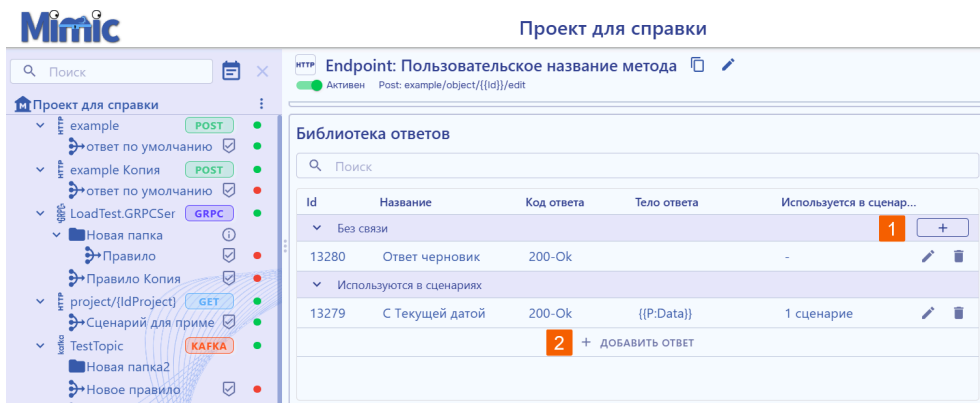
Библиотека_ответов

Ответы сгруппированы по статусу использования:

- **Без связи** — не назначены ни одному сценарию
- **Используются в сценариях** — задействованы в одном или нескольких сценариях


Создание/редактирование из Библиотеки ответов

Для создания нового ответа перейти на вкладку “Библиотека ответов”



Библиотека_добавление_ответа

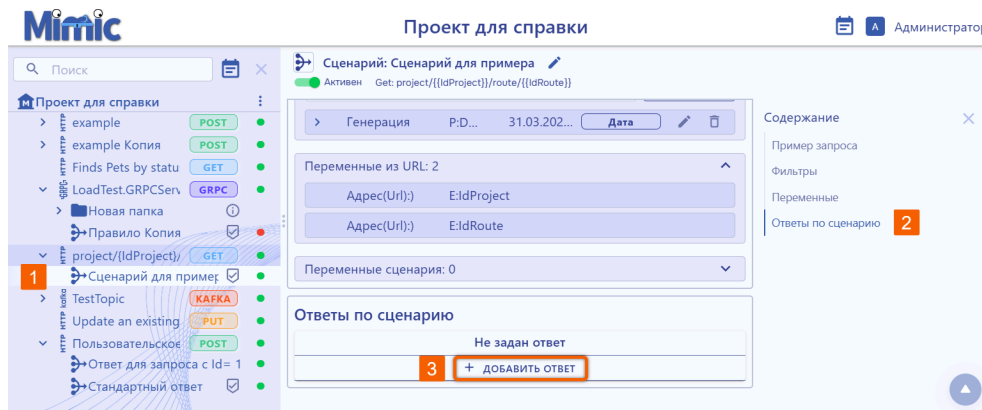
Нажмите одну из кнопок:

1. в списке ответов **Без связи** кнопка **+**
2. в конце таблицы, кнопка **“+ Добавить ответ”**
3. карандаш  рядом с существующим ответом - чтобы отредактировать его

Откроется форма редактирования ответа.


Создание/редактирование из сценария

Вызов создание ответа из сценария



Вызов_создание_ответов

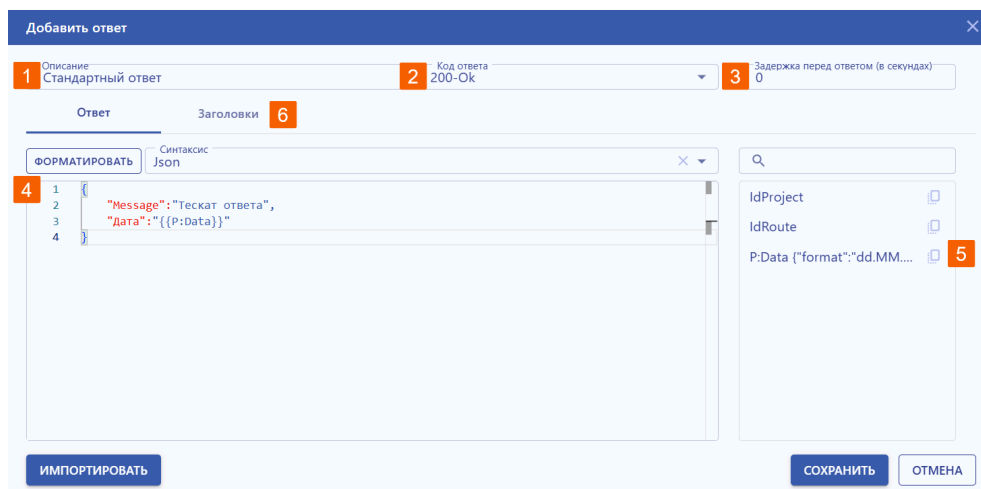
Описание действий:

1. Откройте нужный сценарий
2. Перейдите в раздел “**Ответы по сценарию**”
3. Нажмите кнопку “**+ Добавить ответ**”- для нового ответа, или кнопку карандаш  для редактирования существующего

Откроется форма редактирования ответа.

Заполнение формы создания-редактирования ответа

Форма добавления/редактирования ответа



Добавить_ответ_тело

Описание действий:

1. **“Описание”** - задайте понятное имя для поиска в “Библиотеке ответов”
2. **“Код ответа”** - выберите из списка
3. **Задержка** - укажите время в секундах (0 — без задержки)
4. **Тело ответа** - заполните в требуемом формате, если оно есть
5. **Переменные** - используется синтаксис `{{<префикс>:<имя>}}`:
 - `{{P:Var}}` — переменная проекта
 - `{{E:Var}}` — переменная Endpoint
 - `{{Var}}` — переменная сценария Доступные переменные отображаются на панели справа. Кнопка копировать позволяет вставить шаблон без ручного набора.
6. Для настройки заголовков перейдите на вкладку **“Заголовки”**

Форма добавления/редактирования ответа, вкладка Заголовки

Добавить ответ

Описание: Стандартный ответ | Код ответа: 200-Ok | Задержка перед ответом (в секундах): 0

Ответ | **Заголовки**

Наименование	Значение	Действия
HeaderTest	123	🗑️
Key	Test	🗑️

+ ДОБАВИТЬ | Элементов: 10 | 1-2 из 2 | < >

1

ИМПОРТИРОВАТЬ | 2 СОХРАНИТЬ | ОТМЕНА

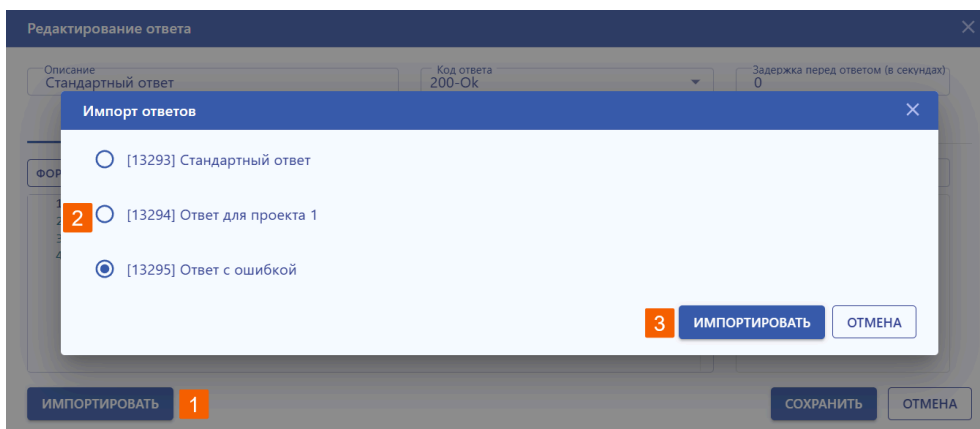
Добавить_ответ_заголовки

Описание действий:

1. Для добавления заголовка нажмите кнопку **“+ Добавить”**
2. Укажите **Наименование** (ключ) и **Значение**
3. Подтвердите нажатием **✓**
4. Сохраните ответ кнопкой **“Сохранить”**

Использование общих ответов

Импорт ответа из Библиотеки ответов



Импорт_ответа

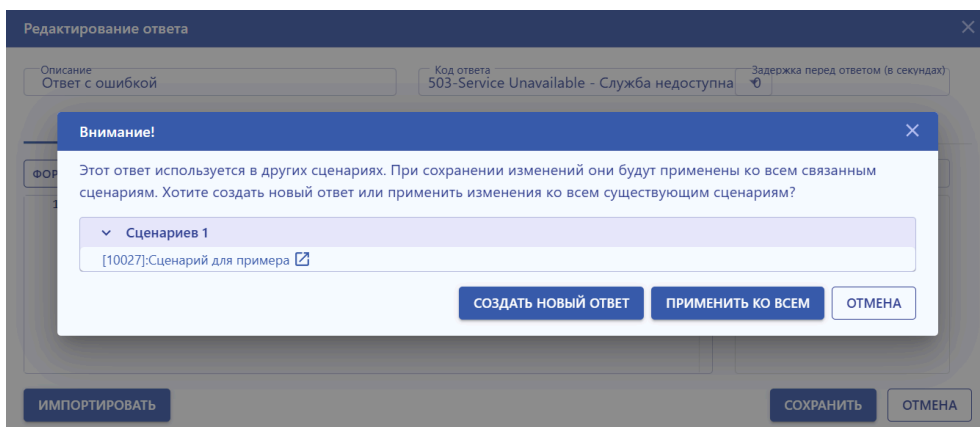
Описание действий:

1. В форме редактирования ответа нажмите **“Импортировать”**
2. В открывшемся окне выберите нужный ответ из списка
3. Нажмите кнопку **“Импортировать”** — данные загрузятся в форму **“Редактирования ответа”**

Редактирование общего ответа

При редактировании ответа, при попытке сохранения изменений система проверяет, является ли он уникальным для одного сценария или общим (используется в нескольких).

- **в одном сценарии** - изменения применяются сразу
- **в двух и более** - открывается диалоговое окно с предупреждением:



Редактирование общего ответа

Варианты действий

- “Создать новый ответ” — дублирует ответ и привязывает к текущему сценарию (остальные сценарии останутся с оригиналом)
- “Применить ко всем” — сохраняет изменения для всех связанных сценариев
- “Отмена” — отменяет сохранение и возвращает в редактор

Особенности ответа для gRPC

Ответное сообщение для gRPC должно строго соответствовать структуре, определённой в .proto-файле, и задаётся в формате JSON. При этом необходимо учитывать:

- Типы данных (например, string, int32, bool, вложенные сообщения)
- Иерархию полей (вложенность объектов)
- Регистр имён полей
- Возможно преобразования имён полей из .proto в JSON

⚠ Важно Если .proto-файл использует имена полей в **snake_case** (например, `user_id`, `order_status`), то при сериализации в JSON по умолчанию такие поля будут автоматически преобразованы в `camelCase` (`userId`, `orderStatus`). Это поведение вызвано тем, что сгенерированные библиотеки gRPC для C# следуют соглашениям .NET и применяют стандартное JSON-преобразование, где **snake_case** → **camelCase**. В результате, если вы укажете в теле ответа `“user_id”: “123”`, сервер может не распознать это поле - ожидая `“userId”`.

Способы решения преобразования `snake_case` → `camelCase`

1. Заменить в .proto-файле `snake_case` на `camelCase` Если вы управляете контрактом, рекомендуется использовать `camelCase` для имён полей прямо в .proto-файле. Этот стиль является универсальным и корректно поддерживается большинством языковых стеков (включая Java, Go, JavaScript и C#), что позволяет избежать расхождений при генерации кода.

Примечание: Это самый простой и надёжный способ, однако он применим только в том случае, если вы можете изменять .proto-контракт

2. Указать явное JSON-имя в .proto-файле В .proto можно задать желаемое имя поля в JSON с помощью опции `json_name`:

```
message GetUserResponse {  
    string user_id = 1 [json_name = "user_id"];  
}
```

В этом случае сериализация будет использовать указанное имя, независимо от языка реализации

3. Использовать camelCase в теле ответа Просто задавайте поля ответа в camelCase Пример:

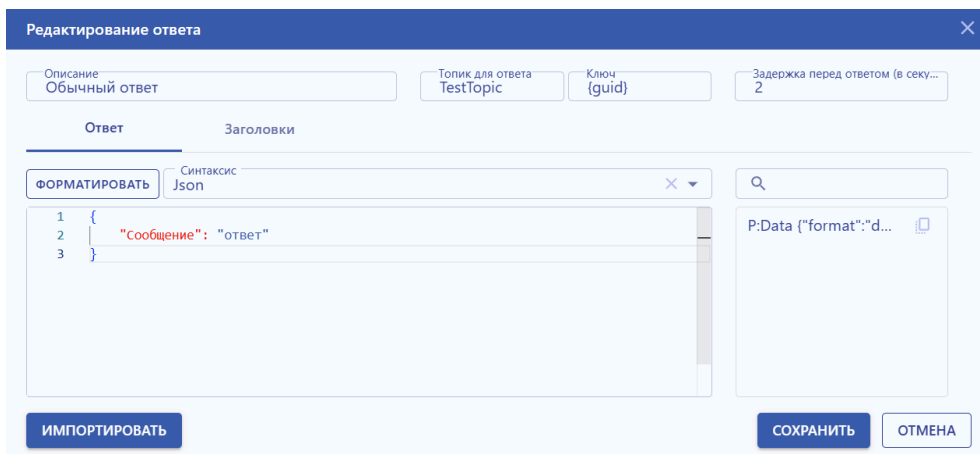
```
{  
    "userId": "123", "orderStatus": "PENDING"  
}
```

В этом случае они будут корректно переданы конечному сервису в ответе и он их прочитает согласно формату protobuf.

Ответное сообщение для Kafka

Все действия по открытию формы создания редактирования ответа аналогичны как и для HTTP/GRPC


Форма редактирования ответа Kafka



Ответ кафка

Форма аналогична HTTP/gRPC, но содержит специфичные поля:

- **Топик для ответа** - топик, в который будет отправлен ответ при срабатывании сценария
- **Ключ** - поле, в котором можно задать значение для передачи в ключе kafka-события. Также возможно задать `{{guid}}`, тогда для ключа сообщения ответа будет сгенерировано guid-значение.

 **Права пользователей** Создавать ответные сообщения и управлять ими могут “Владелец” проекта и “Пользователь”, просматривать - “Читатель”.

См. также

- Создание переменных
- Добавление сценария
- Создание проекта
- Журнал событий

Импорт из спецификации OpenAPI

Mimic на основе спецификации OpenAPI позволяет реализовать три основных сценария:

- Создать один маршрут — при настройке конкретного endpoint'a
- Создать сразу несколько маршрутов — на уровне проекта
- Загрузить примеры запросов и ответов в имеющийся маршрут

Загрузить спецификацию OpenAPI можно из двух источников:

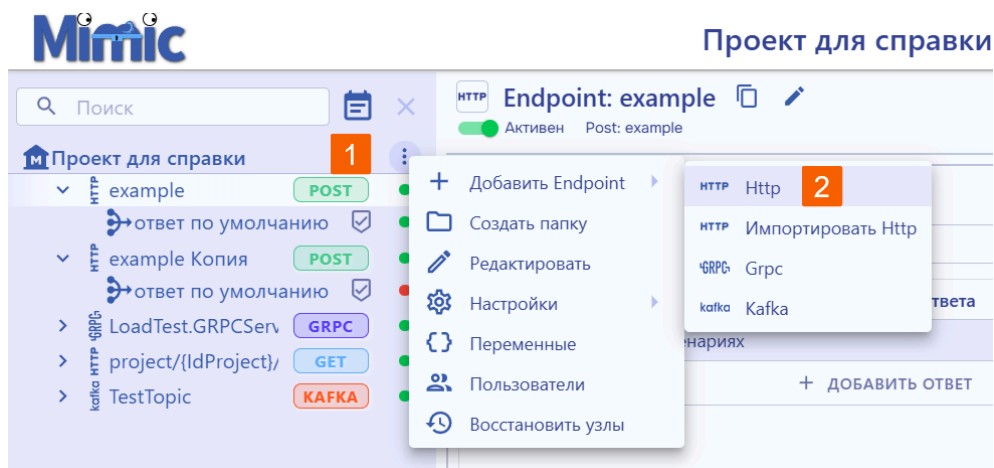
- Из файла (JSON)
- По URL (указывается ссылка до спецификации сервиса)

Ниже описаны все сценарии. Шаги по загрузке спецификации (файл/URL) одинаковы во всех случаях.

Импорт одного маршрута

Используется, когда нужно быстро создать мок на основе одного метода из спецификации.

Открытие формы создания метода



Создание_маршрута

Описание действий

1. Вызовите контекстное меню проекта или папки уровня проекта (например, правой кнопкой мыши по имени в дереве слева или через значок : рядом с его названием)
2. В открывшемся меню выберите **“+ Добавить Endpoint”**, затем подпункт **“HTTP”**

Форма Добавления/Редактирование Endpoint



Создание_маршрута_импорт

Описание действий Для импорта данных из OpenAPI необходимо нажать кнопку **“Импорт”**

Далее следуйте инструкциям из раздела **“Загрузка спецификации”**

Подтверждение создания метода

Добавить Endpoint

Протокол: Http

Наименование: Создание пользователя

Endpoint: user

Post

ИМПОРТ

Создать примеры: Запрос: 1, Ответы: 1

Сохранить, ЗАКРЫТЬ

Создание_маршрута_импорт_загруженный

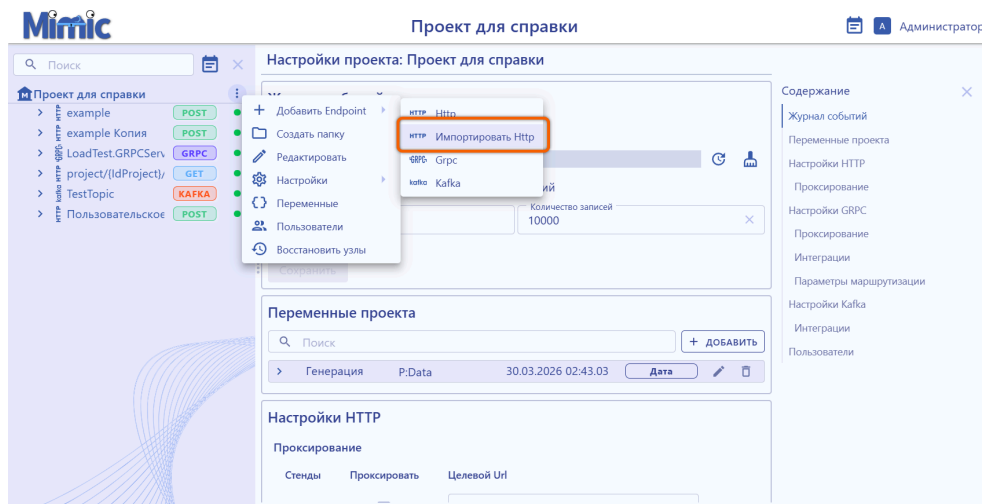
Описание действий

1. Поля **“Наименование”**, **“Тип”** и **“Endpoint”** заполняются автоматически на основе данных из спецификации выбранного маршрута. При необходимости их можно изменить.
2. Ниже отображаются связанные данные:
 - Пример входящего запроса
 - Ответы, указанные в спецификации Все ответы автоматически добавляются в **Библиотеку ответов** и могут использоваться при настройке сценариев маршрута. Если запрос или ответы не требуется создавать, их можно удалить, нажав на соответствующий значок крестика.
3. Нажмите **“Сохранить”**, чтобы завершить создание маршрута.

Импорт нескольких маршрутов

Используется для массового создания моков (например, при первоначальной настройке сервиса).

Открытие формы импорта



Открытие импорта

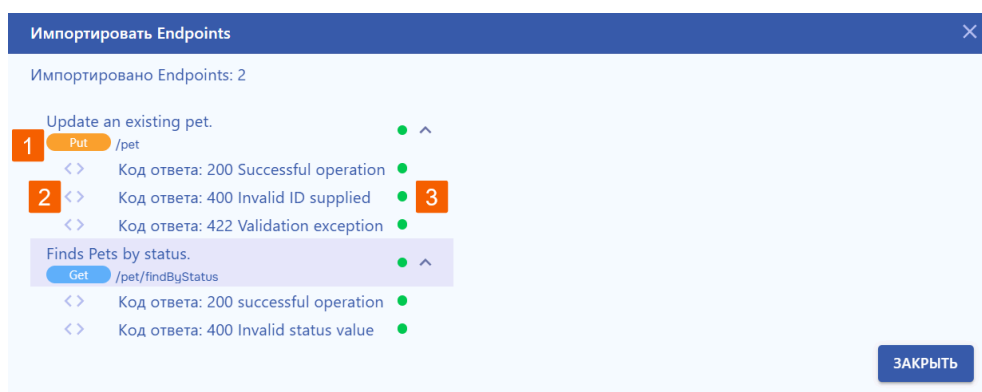
Описание действий

1. Вызовите контекстное меню проекта (например, правой кнопкой мыши по имени в дереве слева или через значок : рядом с его названием)
2. В открывшемся меню выберите **“Импортировать”**, затем подпункт **“HTTP”**

Откроется форма “Импортировать Endpoint”

*Далее следуйте инструкциям из раздела **“Загрузка спецификации OpenAPI”***

Форма процесса создания Endpoints Откроется после подтверждения загрузки.



Форма_результата_импорта

Описание:

1. Отображается список всех загружаемых маршрутов

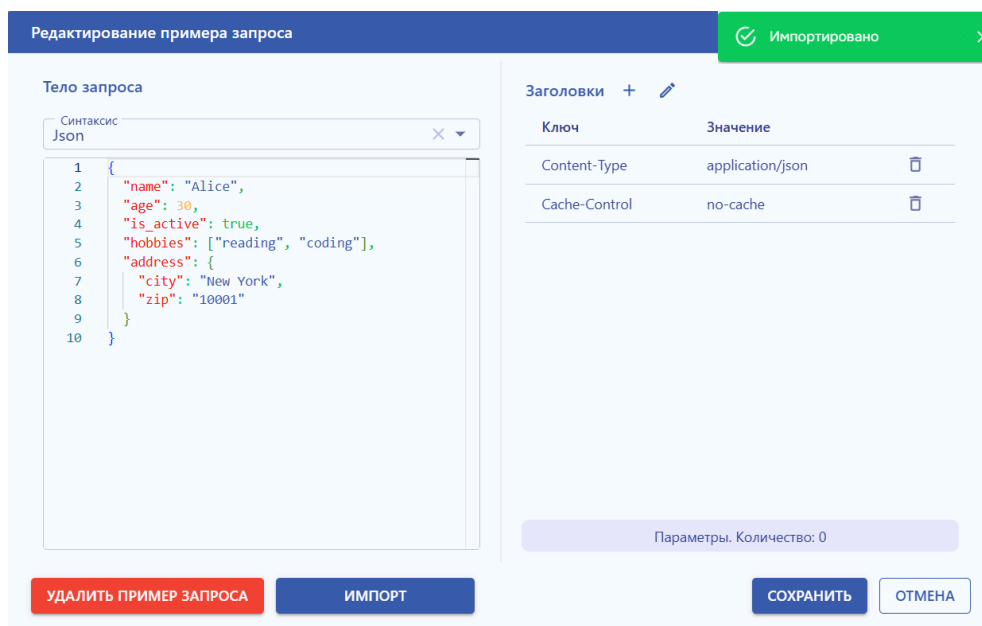
2. Для каждого маршрута можно развернуть детали и просмотреть создаваемые ответы
3. Указан статус загрузки:
 - **Зелёный** — маршрут успешно создан.
 - **Красный** — при создании возникли ошибки.

Импорт примера запроса

Операция импорта примера запроса доступна из формы “**Редактирования примера запроса**” по кнопке “**Импорт**” (подробней смотри раздел Endpoint)

Далее следуйте инструкциям из раздела “**Загрузка спецификации OpenAPI**”

После выбора метода его данные будут подгружены в форму:



The screenshot shows a web interface for editing a request example. At the top, there's a blue header with the text 'Редактирование примера запроса' and a green status bar on the right that says 'Импортировано' with a checkmark and a close button. The main area is split into two panels. The left panel, titled 'Тело запроса', has a dropdown menu for 'Синтаксис' set to 'Json'. Below it is a code editor with a JSON object:

```
1 {
2   "name": "Alice",
3   "age": 30,
4   "is_active": true,
5   "hobbies": ["reading", "coding"],
6   "address": {
7     "city": "New York",
8     "zip": "10001"
9   }
10 }
```

 The right panel, titled 'Заголовки', has a table with two columns: 'Ключ' and 'Значение'. It contains two entries: 'Content-Type' with value 'application/json' and 'Cache-Control' with value 'no-cache'. At the bottom of the form, there are four buttons: 'УДАЛИТЬ ПРИМЕР ЗАПРОСА' (red), 'ИМПОРТ' (blue), 'СОХРАНИТЬ' (blue), and 'ОТМЕНА' (grey). A status bar at the bottom right says 'Параметры. Количество: 0'.

Пример_запроса_загружен

- Проверьте и при необходимости отредактируйте данные
- Нажмите “**Сохранить**”, чтобы применить изменения

Загрузка спецификации OpenAPI

Этот блок используется во всех сценариях импорта.

- По URL

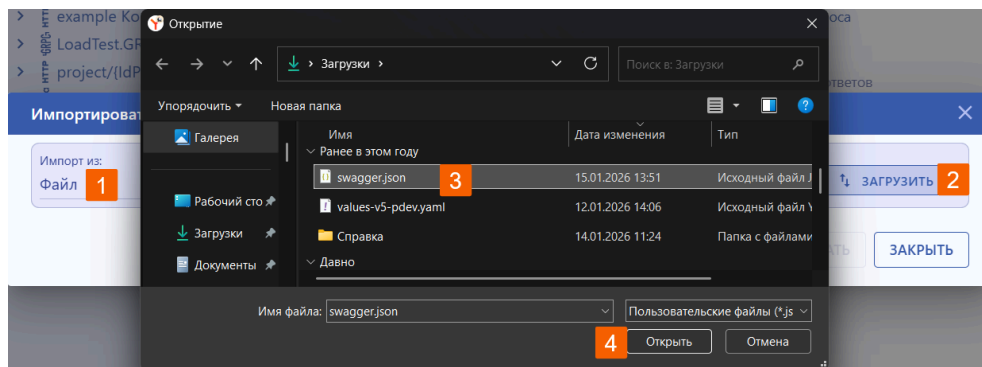


Импорт_OpenAPI_url

Описание действий

1. В поле **“Импорт из”** выберите **“Url”**
2. Укажите прямую ссылку на JSON-файл спецификации
3. Нажмите кнопку **“Загрузить”**

• Из файла

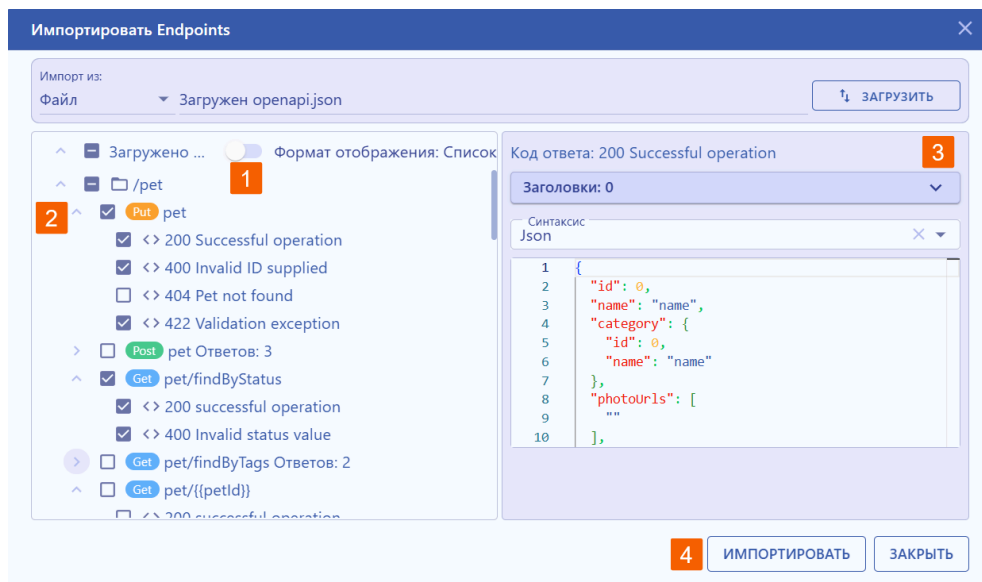


Импорт_OpenAPI_file

Описание действий

1. В поле **“Импорт из”** выберите **“Файл”**
2. Нажмите **“Загрузить”**
3. Выберите файл со спецификацией OpenAPI (.json или .yaml)
4. Нажмите **“Открыть”**


Работа с загруженными данными После загрузки спецификации вы можете выбрать, маршрут для импорта



Импорт_OpenAPI_загруженный

Описание действий

1. Для удобства просмотра можете переключиться между отображением маршрутов в виде списка или в виде папок
2. Отметьте галочкой методы и ответы для импорта (при загрузке в Endpoint или в пример запроса можно выбрать только один объект)
3. Перед импортом вы можете ознакомиться с деталями запроса: параметрами, заголовками, телом и т.д.
4. Нажмите кнопку **“Импортировать”**, после чего выбранный метода и связанные с ним данные будут переданы в форму создания маршрута.

 **Права пользователей** Импорт маршрутов доступны пользователям со следующими ролями: “Администратор”, “Владелец” проекта и “Пользователь”. Пользователи с ролью “Читатель” могут только просматривать данные и не имеют права вносить изменения.

См. также разделы

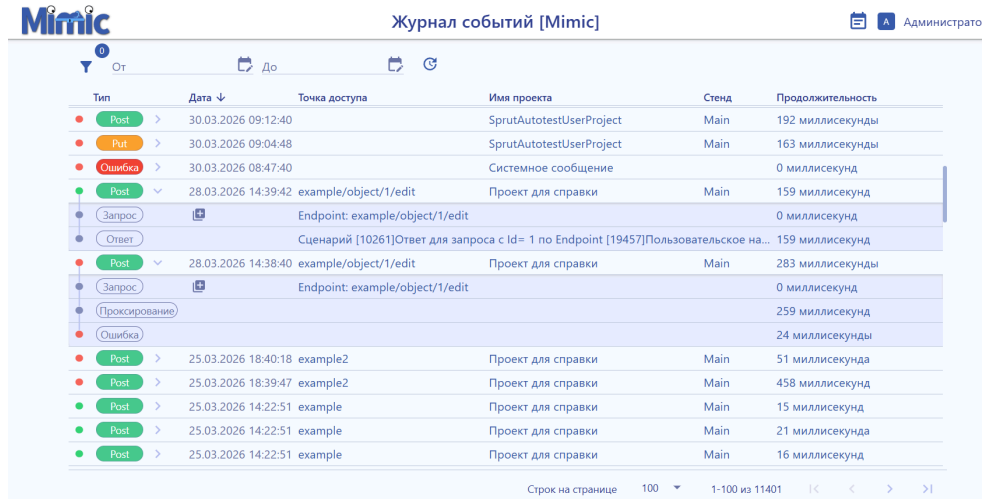
- Создание проекта
- Создание Endpoint
- Создание сценария
- Журнал событий

Журнал событий

Журнал событий отображает все процессы, происходящие в системе. Каждый процесс представляет собой цепочку событий, начинающуюся с входящего запроса и завершающуюся ответом, проксированием или ошибкой.

i **Процесс** - это уникально идентифицируемая последовательность связанных событий, охватывающая обработку одного запроса от начала до конца. Процесс всегда начинается с запроса и завершается одним из следующих исходов: проксированием запроса, формированием ответа или возникновением ошибки.

i **Событие** — это отдельная операция в рамках процесса, фиксирующая конкретное действие или этап обработки. Каждое событие имеет тип (например, Запрос, Ответ, Проксирование или Ошибка) и содержит данные, соответствующие его роли в процессе.



Журнал событий [Mimic] | Администратор

Тип	Дата	Точка доступа	Имя проекта	Стенд	Продолжительность
Post	30.03.2026 09:12:40		SprutAutotestUserProject	Main	192 миллисекунды
Post	30.03.2026 09:04:48		SprutAutotestUserProject	Main	163 миллисекунды
Ошибка	30.03.2026 08:47:40		Системное сообщение		0 миллисекунд
Post	28.03.2026 14:39:42	example/object/1/edit	Проект для справки	Main	159 миллисекунд
Запрос		Endpoint: example/object/1/edit			0 миллисекунд
Ответ		Сценарий [10261]Ответ для запроса с Id= 1 по Endpoint [19457]Пользовательское на...			159 миллисекунд
Post	28.03.2026 14:38:40	example/object/1/edit	Проект для справки	Main	283 миллисекунды
Запрос		Endpoint: example/object/1/edit			0 миллисекунд
Проксирование					259 миллисекунд
Ошибка					24 миллисекунды
Post	25.03.2026 18:40:18	example2	Проект для справки	Main	51 миллисекунда
Post	25.03.2026 18:39:47	example2	Проект для справки	Main	458 миллисекунд
Post	25.03.2026 14:22:51	example	Проект для справки	Main	15 миллисекунд
Post	25.03.2026 14:22:51	example	Проект для справки	Main	21 миллисекунда
Post	25.03.2026 14:22:51	example	Проект для справки	Main	16 миллисекунд

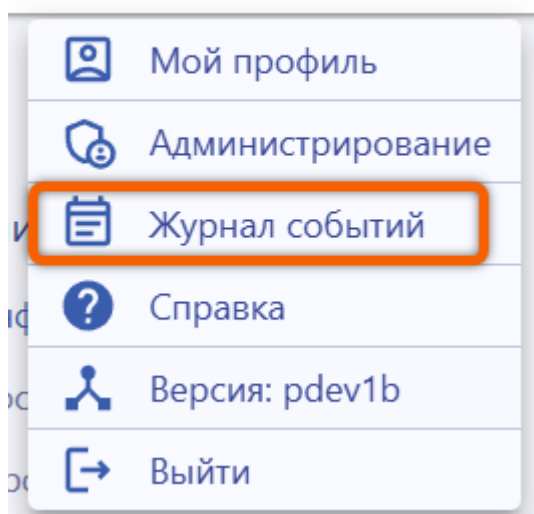
Строк на странице 100 | 1-100 из 11401

Журнал событий


Работа с Журналом событий

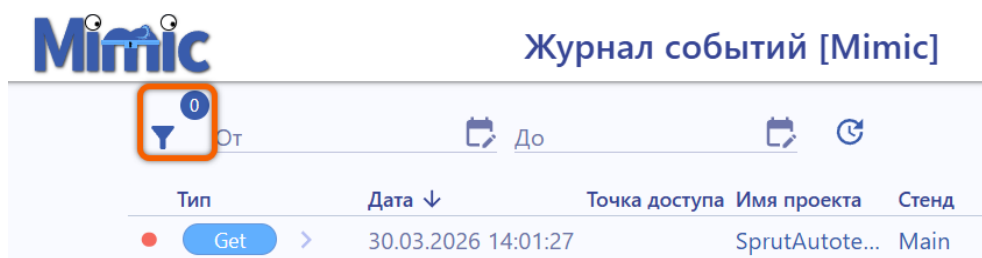
Переход к Журналу событий доступен со всех страниц Mimic, кроме самой страницы Журнала событий и разделов справки.

A Администратор



Главная страница

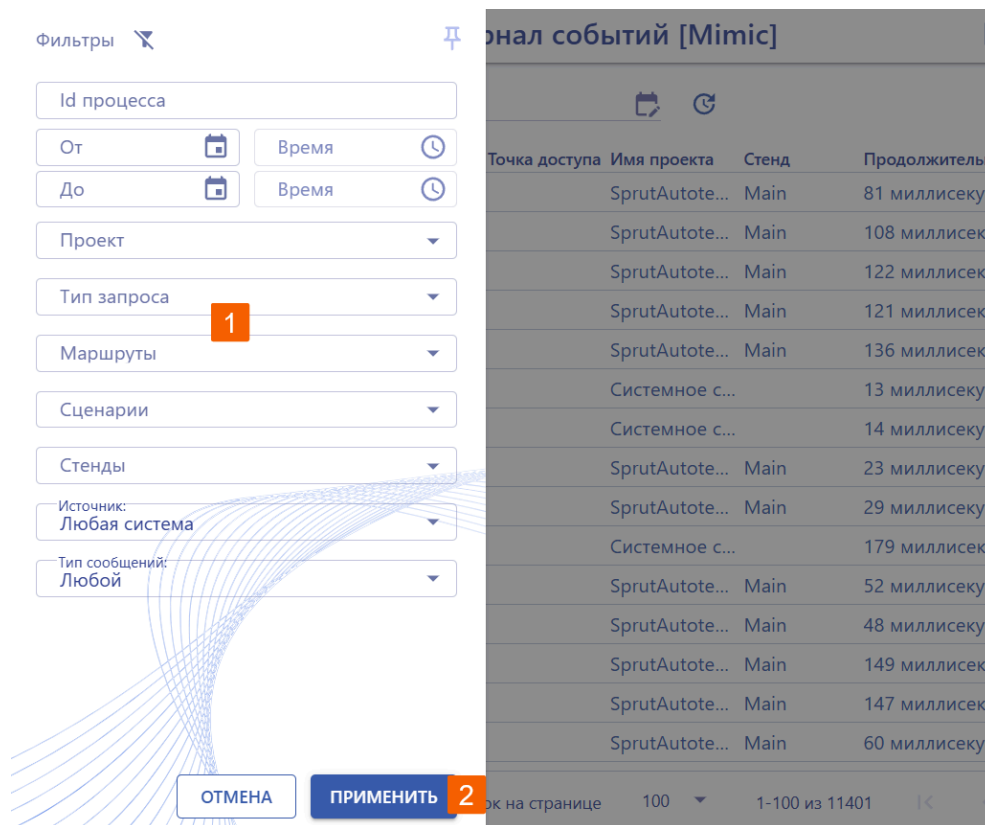
Чтобы перейти в журнал событий, нажмите кнопку  из меню пользователя в правом верхнем углу страницы.



Кнопка фильтра

В верхнем меню нажмите значок фильтра, чтобы просмотреть активные фильтры или настроить дополнительные.

Настраиваем фильтры журнала

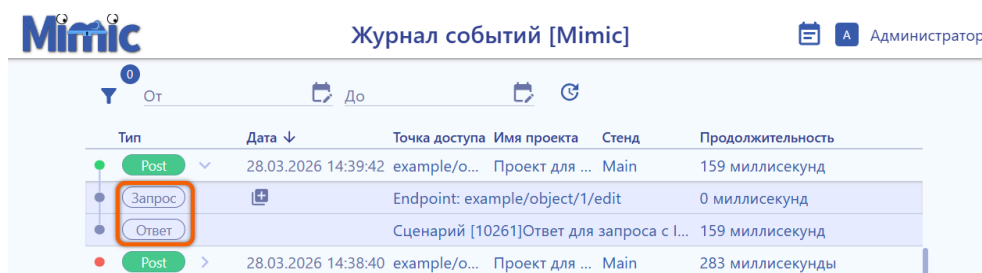


Фильтры журнала событий

1. Укажите необходимые значения в **фильтры**
2. Нажимаем кнопку **“Применить”** для отбора процессов

Процесс и события

В каждом процессе есть возможность просмотра событий.



Журнал просмотр событий

При **просмотре запроса** указывается информация о: проекте, маршруте, заголовках, параметрах и теле запроса.

Запрос
Продолжительность: 0 миллисекунд 28.03.2026 14.39.42 962

Проект [12020] Проект для справки Post example/object/1/edit

Заголовки: 8

Параметры: 0

Тело запроса

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <book>
3 <name>A Song of Ice and Fire</name>
4 <author>George R. R. Martin</author>
5 <language>English</language>
6 <genre>Epic fantasy</genre>
7 </book>
```

Id процесса: 738ff2d7-1b05-4447-bdd5-0d07f3ffefe4

[СОЗДАТЬ СЦЕНАРИЙ](#) [OK](#)

Запрос

При **успешном ответе** есть возможность **просмотра ответа**, где указывается информация о: проекте, маршруте, сценарии, заголовках, теле ответа и задержке;

Ответ
Продолжительность: 159 миллисекунд 28.03.2026 14.39.43 121

Проект [12020] Проект для справки Post [19457] Пользовательское названи Сценарий [10261] Ответ для запроса с Id= 1

Статус код [200] Ok Задержка ответа 00:00:00

Заголовки: 0

Тело ответа

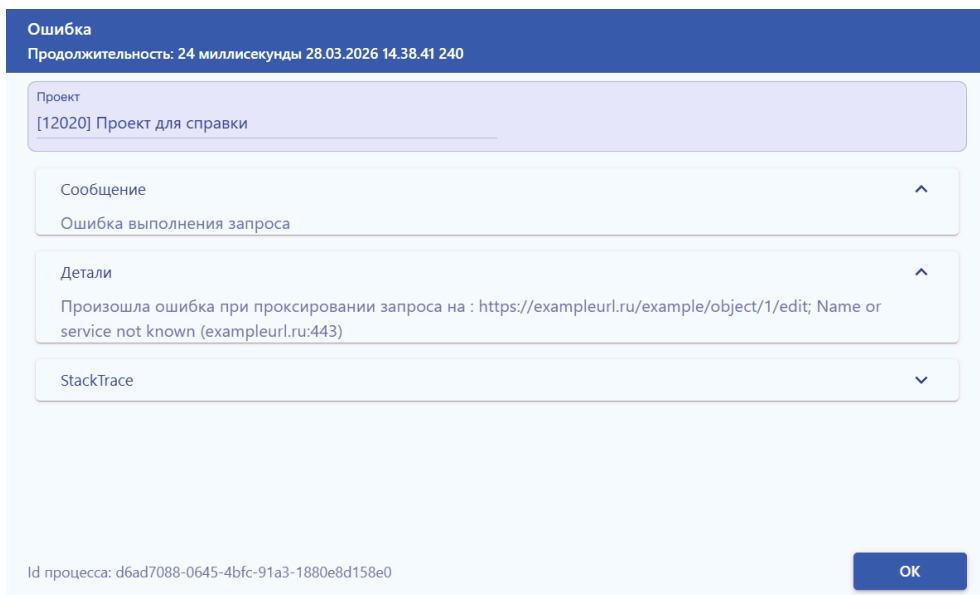
```
1 28.03.2026 11:40.26
```

Id процесса: 738ff2d7-1b05-4447-bdd5-0d07f3ffefe4

[OK](#)

Ответ

Если при обработке **произошла ошибка**, то у соответствующего процесса статус подсвечивается красным цветом, и в описании появляется **событие Ошибка**, при раскрытии которого можно ознакомиться с полным текстом ошибки.



Ошибка

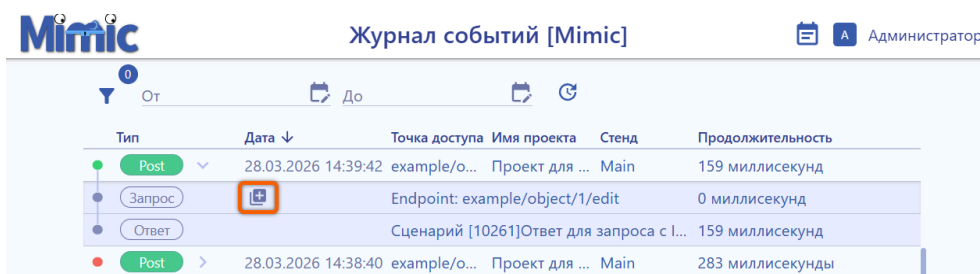
Вызов конструктора сценария

В Журнале событий можно быстро создать сценарий на основе реального входящего запроса. Этот подход особенно удобен, когда нужно:

- настроить обработку типовых запросов
- создать точную имитацию на основе конкретного примера

При выборе события система автоматически извлекает его метаданные — заголовки, параметры URL и тело запроса — и предзаполняет ими форму в конструкторе. Вы можете доработать условия, ответ или переменные перед сохранением.

Вызвать конструктор можно двумя способами: 1. **Через таблицу событий** - нажмите кнопку “+” в строке нужного события



Создать сценарий

2. **Из детального просмотра запроса** - Откройте событие “Запрос” и в форме нажмите кнопку “Создать сценарий”

Запрос
Продолжительность: 0 миллисекунд 28.03.2026 14.39.42 962

Проект
[12020] Проект для справки
Post
example/object/1/edit

Заголовки: 8 ▼

Параметры: 0 ▼

Тело запроса ▲

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <book>
3 <name>A Song of Ice and Fire</name>
4 <author>George R. R. Martin</author>
5 <language>English</language>
6 <genre>Epic fantasy</genre>
7 </book>

```

Id процесса: 738ff2d7-1b05-4447-bdd5-0d07f3ffef4

СОЗДАТЬ СЦЕНАРИЙ
OK

Запрос

Подробнее о работе с конструктором см. в разделе Конструктор сценариев

🔒 Права пользователей Просматривать журнал событий могут пользователю с любой ролью в проекте. Вызывать конструктор сценариев могут только пользователи, чья роль отличается от “Читатель” и “Гость”

Управление сценариями через API

Активность сценария можно изменять программно — через REST API Mimic.

⚠️ Важно Для вызова API необходимо предварительно создать API-токен в профиле пользователя.

Описание метода

Параметр	Значение
HTTP-метод	<i>PUT</i>
Эндпоинт	<i>{MimicUIhost}/rule/activity</i>
Заголовок авторизации	<i>Authorization: Bearer</i>

Тело запроса (JSON):

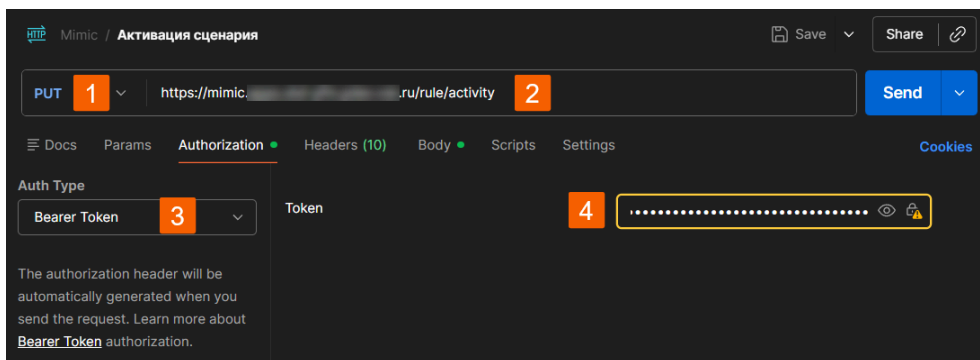
```
{
  "Id": 10257,
  "IsActive": true
}
```

Параметр	Тип	Описание
Id	integer	Идентификатор сценария (берётся из URL: /rule/{id})
IsActive	boolean	true — активировать, false — деактивировать

Примечание Идентификаторы сценариев уникальны в рамках всей системы, поэтому для управления не требуются дополнительные параметры (проект или endpoint).

Пошаговая инструкция

1. Подготовка запроса В качестве примера мы выполним запрос через Postman



Заголовок авторизации

Описание действий

1. Выберите HTTP-метод **PUT**
2. Укажите адрес: **{MimicUIhost}/rule/activity**
3. Укажите тип авторизации **Bearer Token**
4. Укажите созданный API-токен

2. Активация сценария посредством Api Пример URL сценария в интерфейсе: ***https://mimic.example.ru/Project/12020/route/17036/rule/10257#example_request***

- `mimic.example.ru` приведён как пример, нужно подставить свой хост для интерфейса
- Идентификатор сценария: 10257

Описание действий

1. Скопируйте `Id` из URL (после `/rule/`)
2. Сформируйте тело запроса:

```
{  
  "Id": 10257,  
  "IsActive": true  
}
```

3. Отправьте запрос
4. После успешного ответа (200 OK) проверьте интерфейс Mimic — статус сценария обновится

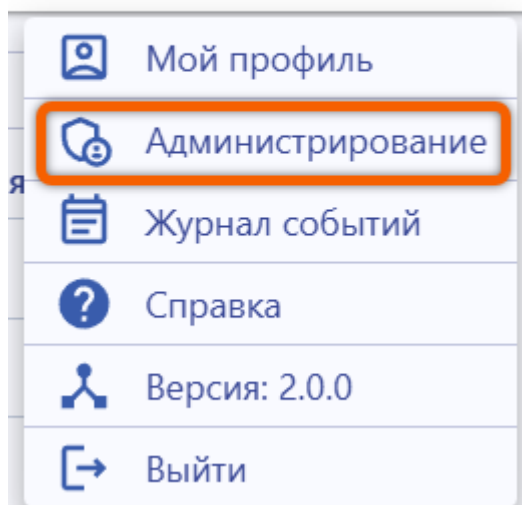
Пример полного запроса (cURL)

```
curl -X PUT \  
  https://mimic.example.com/rule/activity \  
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \  
  -H "Content-Type: application/json" \  
  -d '{ "Id": 10257, "IsActive": true}'
```

Администрирование

Чтобы перейти к функциям администратора, откройте **Меню профиля** → **Администрирование**. Для открытия меню профиля нажмите на имя пользователя в правом верхнем углу интерфейса.

A Администратор



Администрирование

Администрирование позволяет:

- управлять пользователями и их ролями
- добавлять новых пользователей
- просматривать состояние лицензии
- вводить лицензионные ключи

Управление пользователями и глобальными ролями

Интерфейс раздела **“Пользователи”** представлен единым списком всех учётных записей, в котором для каждого пользователя отображается его глобальная роль (если назначена).

ФИО	Логин	Email	Роль	Авторизация	Последний вход
...	Администратор	Keycloak	04.02.2025 17:52
...	Администратор	Keycloak	06.04.2026 16:54
...	Администратор	Keycloak	06.04.2026 15:39
...	Администратор	Mimic	05.02.2025 11:01
Админ Мимика	AdminMimic	admin@admin.mi	Администратор	Mimic	06.04.2026 21:15
Test_Admin	Test_Admin	Test_Admin@test.com	Администратор	Mimic	18.12.2025 10:09
...	Администратор	Keycloak	08.04.2026 12:38
Администратор (Вы)	admin	admin@admin.ru	Администратор	Mimic	08.04.2026 12:59
...	Пользователь	Mimic	19.09.2024 14:22
...	Пользователь	Keycloak	23.09.2025 17:15
Имя пользователя	user	HelpUser@exampl.ru	Пользователь	Mimic	13.03.2026 14:58
...	Зритель	Keycloak	19.08.2024 14:31
Reader	Readre	Reader@read.ru	Зритель	Mimic	18.02.2026 12:16
...		Mimic	15.07.2024 09:40
Тестер Один Автогестович	tester1@test.ru	tester1@test.ru		Mimic	15.01.2026 16:01

Пользователи Mimic

Общие функции списка пользователей

- **Поиск** Введите текст в поле **“Поиск”** — список отфильтруется по совпадениям в любом из полей: ФИО, Логин или Email.
- **Сортировка** Нажмите на заголовок любого столбца, чтобы отсортировать список по этому полю. Например, чтобы увидеть давно не входивших в систему пользователей, нажмите на заголовок **“Последний вход”**.
- **Фильтрация по ролям** В столбце **“Роль”** доступна кнопка фильтра — с её помощью можно отобразить только пользователей с выбранными глобальными ролями или, наоборот, без роли.

Роль	Количество
<input checked="" type="checkbox"/> Пользователь	3
<input checked="" type="checkbox"/> Администратор	8
<input type="checkbox"/> Владелец	0
<input type="checkbox"/> Зритель	2
<input type="checkbox"/> Нет роли	43

Фильтр ролей Mimic

- **Постраничная навигация** Список разбит на страницы. Для перехода между ними используйте стрелки навигации внизу таблицы.

Добавление пользователя в систему Mimic

Система поддерживает два способа аутентификации:

- OAuth2 (Keycloak)
- Локальная аутентификация по логину и паролю

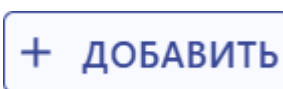
Пользователи, впервые входящие через Keycloak, автоматически регистрируются в системе — дополнительных действий не требуется.

i Примечание Наличие OAuth2 авторизации зависит от конфигурации системы при развёртывании

Для использования локальной аутентификации, нового пользователя необходимо создать вручную:

1. Перейдите в раздел **“Администрирования”** → **“Пользователи”**

2. В верхнем углу нажмите кнопку добавить:



3. Заполните поля формы:

Скриншот формы «Добавить пользователя» с полями: ФИО, Email, Логин, Пароль (с иконкой глаза), выпадающий список «Mimic» и кнопки «ДОБАВИТЬ» и «ОТМЕНА».

Создать пользователя

Поля формы:

- **Email** - электронная почта пользователя

- **ФИО** - фамилия, имя и отчество
- **Логин** - уникальное имя пользователя (только латинские символы)
- **Пароль** - может содержать латинские и кириллические буквы, цифры и спецсимволы, не короче 8 символов
- **Способ авторизации** - при ручном создании всегда указывается “Mimic”

4. Нажмите кнопку “**Добавить**” для сохранения учётных данных

i **Примечание** После создания передайте логин и пароль пользователю

Глобальные роли

Глобальные роли определяют уровень доступа пользователя ко всей системе Mimic.

i **Примечание** Интерфейс администрирования доступен только пользователям с глобальной ролью “**Администратор**”

Типы глобальных ролей Каждой учётной записи может быть назначена одна из следующих ролей:

- **Администратор** Полный доступ: настройка системы, управление пользователями, назначение ролей, редактирование всех проектов.
- **Пользователь** Может просматривать и изменять все проекты, но не может управлять пользователями в них.
- **Зритель** Только просмотр всех проектов.

i **Примечание** Если у пользователя не назначена глобальная роль, он может работать с проектами только через проектные роли. Подробнее см. [Настройки проекта/Пользователи](#)

Назначение, изменение и удаление глобальной роли

Все операции с глобальными ролями выполняются прямо в списке пользователей.

Чтобы назначить или изменить роль:

1. Перейдите в **“Администрирования”** → **“Пользователи”**
2. Найдите нужного пользователя в списке
3. В столбце **“Роль”** выберите желаемую роль из выпадающего списка.

ФИО	Логин	Email	Роль ↑ ▼	Авторизация	Последний вход
Администратор	admin	admin@admin.mi	Администратор × ▼	Keycloak	04.02.2025 17:52
Администратор	admin	admin@admin.mi	Администратор × ▲	Keycloak	06.04.2026 16:54
Администратор	admin	admin@admin.mi	Администратор	Keycloak	06.04.2026 15:39
Владелец	owner	owner@owner.mi	Владелец	Mimic	05.02.2025 11:01
Пользователь	user	user@user.mi	Пользователь	Mimic	06.04.2026 21:15
Зритель	viewer	viewer@viewer.mi	Зритель	Mimic	18.12.2025 10:09
Администратор	admin	admin@admin.mi	Администратор × ▼	Keycloak	08.04.2026 12:38

Назначение роли

Чтобы удалить глобальную роль:

1. Найдите нужного пользователя в списке
2. В столбце **“Роль”** нажмите значок удаления (крестик) ✕ Поле очиститься и доступ пользователя будет определяться только проектными ролями

Примечание Учётная запись должна существовать в системе. Если пользователя нет в списке — сначала создайте его.

Права пользователей К данному разделу доступ есть только у **“Администратора”** системы Мимик.

Лицензия

Лицензия — это лицензионный ключ, предоставляющий пользователю или организации право на использование системы Mimic в соответствии с условиями выбранного тарифа. При развёртывании в облаке лицензия может быть активирована автоматически в рамках приобретённого тарифного плана.

Условия лицензии

Ограничение по сроку действия Лицензия действует в течение установленного периода. По его истечении: - Создание новых объектов (проектов, endpoint'ов,

сценариев и т.д.) становится недоступно - Работа с существующими данными (просмотр, отправка запросов, анализ журналов) остаётся полностью доступной

Ограничение на количество проектов

Лицензия устанавливает максимальное число проектов, которые можно создать в системе. - При попытке превысить лимит система блокирует создание новых проектов. - Если новая (например, продлённая или пониженная) лицензия разрешает меньше проектов, чем уже создано, редактирование существующих проектов будет ограничено. - Чтобы восстановить полную функциональность, необходимо удалить избыточные проекты до соответствия текущему лимиту.

Количество проектов: 17 из 5

лимит проектов

Ограничение на количество маршрутов в проекте Для каждого проекта устанавливается индивидуальный лимит на число маршрутов (endpoint'ов).

- Ограничение применяется независимо к каждому проекту. Пример: при лимите 10 проектов и 10 маршрутов на проект система поддерживает до 100 маршрутов суммарно.
- Попытка создать маршрут сверх лимита в любом проекте будет автоматически заблокирована.

Отсутствие ограничений на количество пользователей Лицензия не ограничивает число пользователей, которые могут одновременно работать в системе.

Активация лицензии

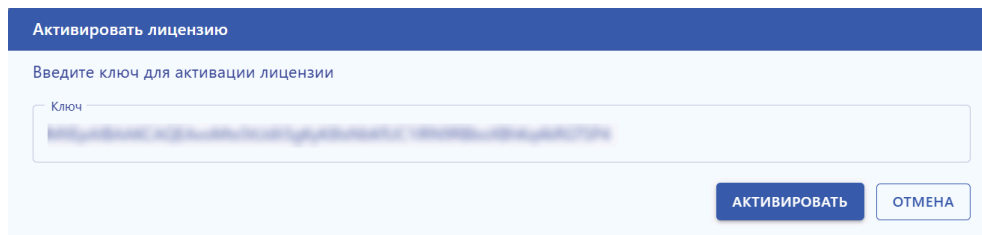
Активировать лицензионный ключ может только системный администратор.

← Администрирование	Лицензия Mimic
Пользователи	Владелец: ООО <Высокие цифровые технологии> Корпоративный пакет
Лицензия	Статус лицензии: Активна ↻
	Количество проектов: 367 из 1000
	Максимальное количество маршрутов по проекту: 1000
	Срок действия от: 19.03.2026
	Срок действия до: 01.01.2030
	Осталось дней: 1384
	ВВЕСТИ КЛЮЧ

лицензия


Порядок активации:

- В верхнем меню нажмите на имя пользователя и выберите раздел **“Администрирование”**
- В боковом меню перейдите в пункт **“Лицензия”**
- Нажмите кнопку **“Ввести ключ”**
- В открывшемся окне вставьте лицензионный ключ и нажмите кнопку **“Активировать”**



лицензия

После успешной активации система автоматически применит все параметры лицензии: срок действия, лимиты проектов и маршрутов.

 **Права пользователей** К данному разделу доступ есть только у **“Администратора”** системы Мимик.

Авторизация

В системе **Mimic** реализованы два варианта авторизации:

1. Основной способ — внутренняя авторизация в системе Mimic Пользователи могут войти в систему, используя учетные данные, созданные администратором Mimic. Данный способ подходит для всех категорий пользователей, включая внешних специалистов, подрядчиков и других лиц, не входящих в штат компании.

2. Дополнительный способ — авторизация через интеграцию с Keycloak Этот вариант доступен при наличии настроенной интеграции между Mimic и KeyCloak в вашей организации. Учетные записи штатных сотрудников хранятся и управляются в KeyCloak, что позволяет им автоматически получать доступ к системе Mimic с правами на просмотр. Интеграция с KeyCloak обеспечивает поддержку единого входа (Single Sign-On, SSO): пользователю достаточно один раз

аутентифицироваться в корпоративной учетной записи, чтобы получить доступ ко всем подключённым системам. Это упрощает процесс входа, снижает количество паролей, которые необходимо запоминать, и повышает общий уровень безопасности за счёт централизованного управления идентификацией и доступом.

Роли в системе

При первом входе в систему пользователь автоматически получает роль **“Гостя”**, если администратором заранее не были назначены иные права.

- **Гость** Может самостоятельно создать новый проект и стать его владельцем. (На созданный проект автоматически будут выданы права с ролью **“Владелец”**)

Роли на проекте

Права доступа к конкретному проекту определяются проектной ролью. Одна и та же учётная запись может иметь разные роли в разных проектах.

- **Владелец** Имеет полный контроль над проектом: может просматривать и редактировать все объекты, переименовывать или удалять проект, а также управлять доступом — добавлять пользователей и назначать им роли («Пользователь», «Зритель» и т.д.)
- **Пользователь** Может просматривать и вносить изменения в содержимое проекта, но не имеет прав на управление участниками проекта или его метаданными.
- **Зритель** Обладает только правом просмотра проекта. Редактирование, создание объектов или изменение настроек недоступны.

Глобальные роли

Глобальные роли определяют уровень доступа пользователя ко всей платформе Mimic.

- **Администратор** Имеет полный доступ ко всем проектам и системным функциям: может просматривать и редактировать любой контент, управлять пользователями (добавлять, удалять, назначать глобальные роли), а также настраивать параметры системы.

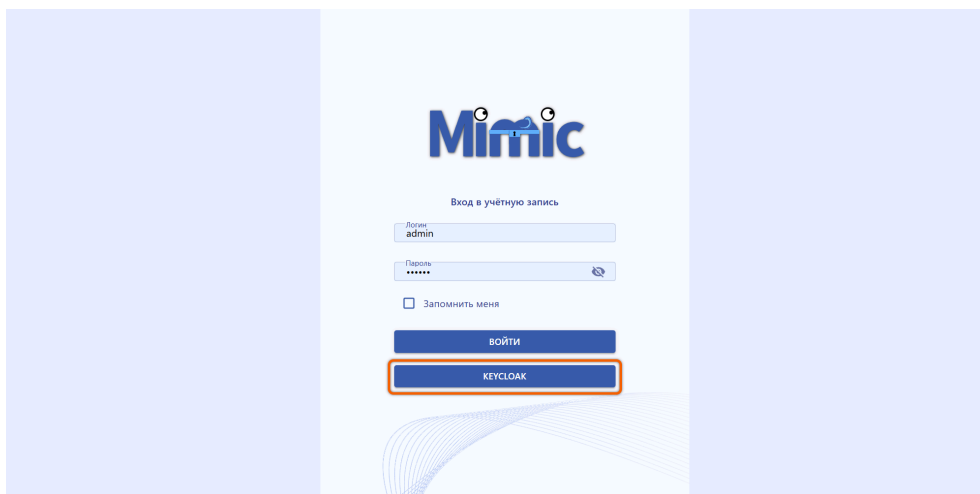
- **Пользователь** Может просматривать и редактировать все проекты на платформе, но не имеет прав на управление пользователями или системными настройками.
- **Зритель** Имеет право только на просмотр всех проектов. Создание, редактирование и управление контентом недоступны.

Такой двухуровневый подход — с разграничением глобальных и проектных ролей — обеспечивает гибкость в управлении доступом и позволяет точно настраивать права в зависимости от задач и контекста использования.

Авторизация пользователя через KeyCloak

Если при развёртывании Mimic настроена интеграция с Keycloak, на странице входа появляется дополнительная кнопка **“Keycloak”**.

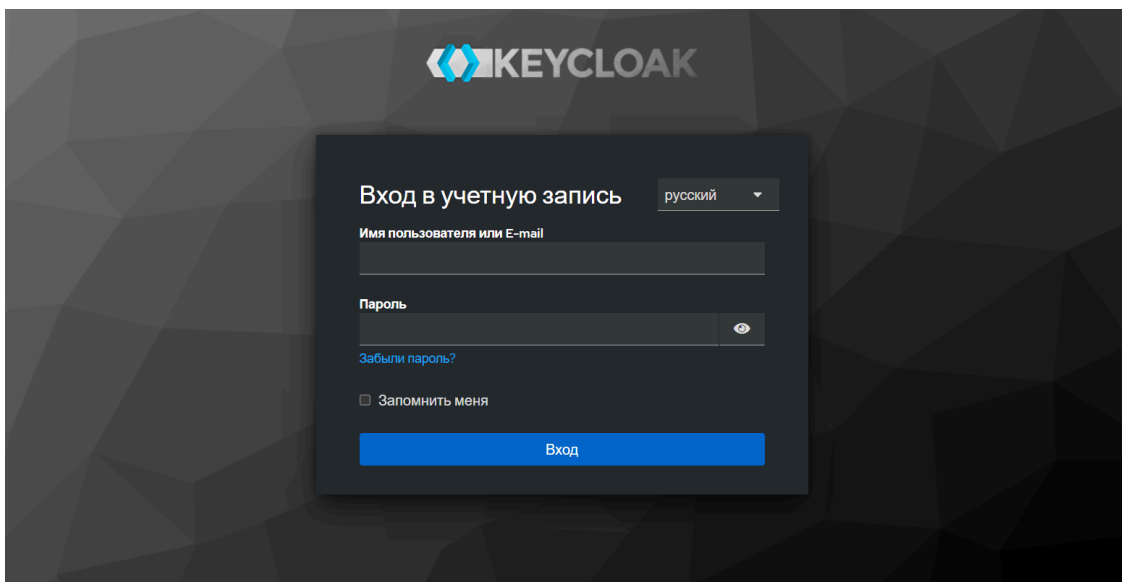
Этот способ доступен только для пользователей, учётные записи которых зарегистрированы в вашем экземпляре Keycloak. При успешной аутентификации пользователь автоматически получает доступ к системе с ролью «Гость».



Вход в систему

Описание действий: Нажимаем кнопку **“KeyCloack”** на странице входа:

- Если вы уже авторизованы в Keycloak — вход в Mimic произойдёт автоматически.
- Если не авторизованы — вы будете перенаправлены на страницу аутентификации Keycloak



Введите учётные данные (логин и пароль) в интерфейсе Keycloak и нажмите “Войти”

См. также

- Проект

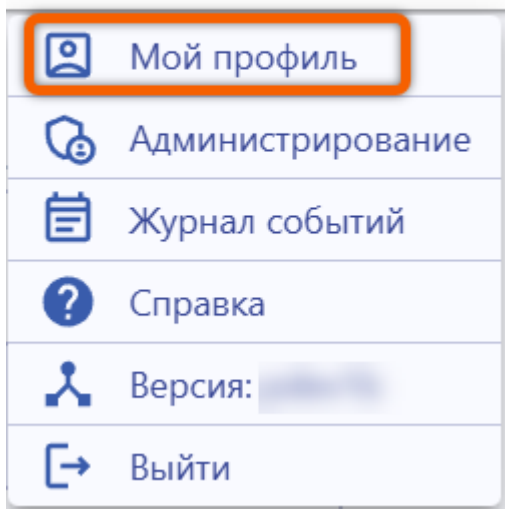
Профиль

Профиль пользователя - персонализированный раздел для просмотра и управления учётной записью.

Переход в профиль

Чтобы открыть профиль, нажмите на свой аватар в правом верхнем углу интерфейса и выберите пункт **Мой профиль**

M Максим Боков



Переход в профиль

Содержимое страницы Профиль

Страница "Профиль":

A screenshot of the Mimic user profile page. The page title is 'Профиль [Mimic]'. The user's name 'Максим Боков' is visible in the top right. The profile section is titled 'Профиль: Максим Боков'. It contains three main sections: 'Личная информация' (Personal information), 'Безопасность' (Security), and 'Роли на проектах' (Roles on projects). The 'Личная информация' section shows fields for 'ФИО' (Максим Боков), 'Логин' (bokov), 'Email', and 'Тип авторизации' (Keycloak). The 'Безопасность' section indicates that password management is unavailable for Keycloak. The 'Роли на проектах' section contains a table with columns 'Id', 'Проект', and 'Роль'.

Id	Проект	Роль
158	Спрут Сервис Агентов	Пользователь
6471	Demo GRPC	Владелец
6952	Demo Kafka	Владелец

Профиль

В профиле доступны следующие функции:

1. Просмотр личных данных

- **ФИО:** - полное имя, отображаемое в интерфейсе Mimic
- **Логин:** - уникальный идентификатор для входа в систему

- **Email:** - адрес электронной почты
- **Тип авторизации:** - способ входа: Keycloak или Mimic

2. Смена пароля



Смена пароля

Пользователь может изменить свой пароль. Процесс включает ввод текущего пароля, нового пароля и его подтверждение.

⚠️ Важно Для пользователей, авторизованных через Keycloak, смена пароля в интерфейсе Mimic недоступна. Пароль необходимо менять в самом Keycloak.

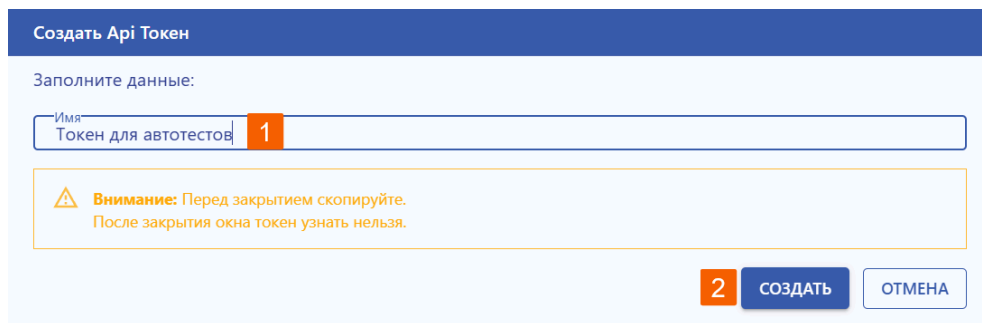
3. **Отслеживание ролей на проектах** Отображает список проектов, в которых вы участвуете, и ваши роли в них (Владелец, Пользователь, Зритель).
4. **API Токены** Для интеграции внешних систем с Mimic требуется API-токен.

💡 Что такое токен? Уникальный строковый идентификатор, используемый для аутентификации и авторизации при вызове API Mimic.

Создание API-токена

1. Перейдите в раздел **“API Токены”**
2. Нажмите кнопку **“Добавить”**

Откроется **Форма создания токена**

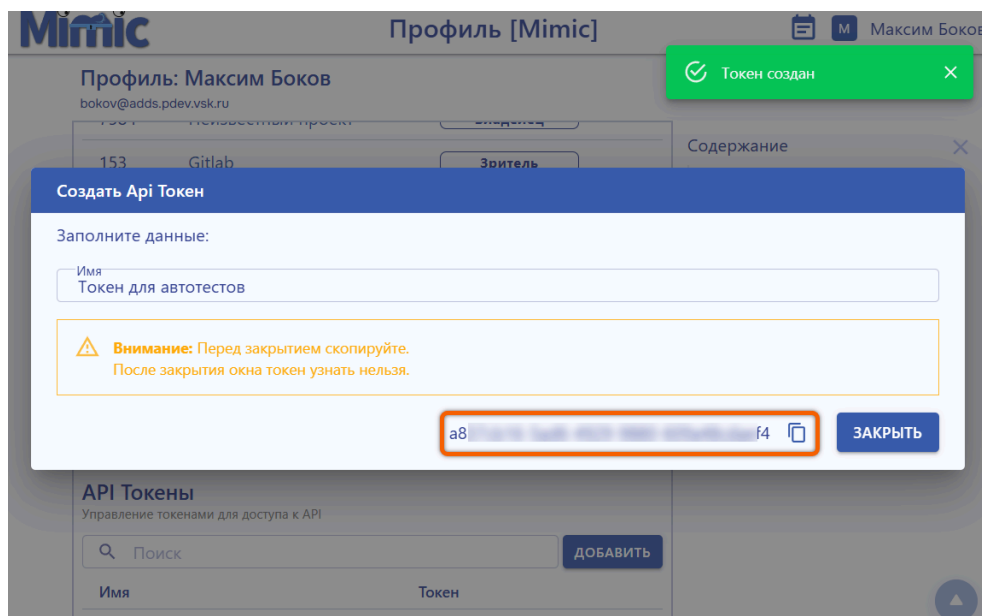


Создание токена

Описание действий

1. Укажите **Название** токена (например, «Автотест» или «Postman»)
2. Нажмите **“Создать”**

После генерации токен отобразится внизу окна:



Токен

⚠ Важно! Скопируйте токен сразу и сохраните его в безопасном месте. После закрытия окна повторный просмотр токена невозможен — его можно только удалить и создать заново.