



REST-запрос

Компонент позволяет взаимодействовать с внешним REST-сервисом, отправляя ему GET, PUT, DELETE, PATCH, POST запросы и получая ответы. В качестве входного набора данных используется таблица с параметрами запросов. REST-сервису будет отправлено столько запросов, сколько строк во входном наборе данных.

Примечание: Поскольку входной набор данных с параметрами запросов является необязательным, то при его отсутствии узел вызова REST-сервиса отправит единичный запрос без параметров.

В выходных портах узла формируются:

- таблица ответов сервиса;
- дополнительные данные о результатах выполнения каждого запроса.

В случае отправки GET и DELETE запросов, каждая строка входного набора должна содержать параметры запроса, а при PUT, PATCH, POST запросах — тело HTTP-запроса. При настройке узла задается произвольное количество заголовков HTTP-запроса.

Для работы с REST-сервисом необходимо настроенное подключение к REST-сервису. Выбор метода HTTP-запроса, а так же определение списка параметров запроса осуществляется при настройке этого подключения. Узел вызова REST-сервиса осуществляет проверку типа ответа, и если полученный ответ не соответствует заявленному в параметре *Tun Content-type ответа* используемого подключения, то выходные наборы данных узла будут содержать сообщения об ошибке.

Порты

Входные порты

-  — Подключение к REST-сервису (связь с компонентом *Подключение REST-сервиса*).
-  — Управляющие переменные (переменные), необязательный.
-  — Источник данных запроса (таблица с параметрами запроса), необязательный.

Выходные порты

-  — Выходной набор данных (таблица с ответами сервиса).

-  — Дополнительные данные (таблица с результатами выполнения запросов, описанием ошибок, кодами завершения и состояния HTTP).

Настройка

Мастер настройки вызова сервиса состоит из нескольких шагов.

Настройка запроса

На данном шаге задаются параметры запроса. Слева расположен список полей входного набора данных. Справа расположен список URL-параметров запроса, принимаемых REST-сервисом. Для дальнейшей настройки необходимо проставить связи между полями и URL-параметрами запроса, сделать это можно несколькими способами:

- Методом *Drag-and-drop* — перетащить метку поля из левого списка полей на элемент списка URL-параметров запроса.
- В правой таблице выбрать нужный URL-параметр запроса и в столбце *Связанные поля* выбрать из выпадающего списка метку поля из входного набора.
- Использовать кнопку  *Связать все автоматически*, при этом происходит автоматическое связывание входных полей и URL-параметров запроса, исходя из значений меток и типов данных полей входного набора и параметров.

Примечание:

- Если связь между полем и объектом была установлена неправильно, ее можно удалить. Для этого нужно выбрать метку и нажать на кнопку  на линии связи. При необходимости удаления всех связей используется кнопка  *Удалить все связи*.
- Поля, которые были связаны с параметрами, недоступны для выбора в качестве тела запроса, а поле, которое было выбрано в качестве тела запроса, не отображается в таблице URL-параметров.

При PUT, PATCH, POST запросах поля входной таблицы связываются с предопределенным параметром `request`, принимающим тело HTTP-запроса.

Параметр **Наличие временной зоны** определяет указывается ли информация о часовом поясе в рамках стандарта ISO 8601 при передачи параметров запроса типа `Дата/Время`. Возможные значения:

- Не указывать
- Указывать всегда

Настройка пользовательских заголовков

На данном шаге задаются имена заголовков HTTP-запроса и их значения.

Дополнительные параметры

- **Выдавать описание кодов завершения** — отображать описание ошибок, произошедших при отправке запроса, получении и разборе ответа сервиса.

Примечание: При кодах состояния HTTP ≥ 400 устанавливается *Код завершения* = 6 (*Ошибка HTTP-клиента*) и выводится соответствующее *Описание ошибки*.

- **Задержка между запросами (мс)** — задает минимальный интервал времени (в миллисекундах) между последовательными REST-запросами.
- **Количество повторных запросов при ошибке** — определяет, сколько раз система будет автоматически повторять запрос при возникновении ошибки.
- **Задержка между неудачными запросами (мс)** — задает паузу (в миллисекундах), которая происходит после каждой неудачной попытки выполнить запрос. Становится доступен, если параметр *Количество повторных запросов при ошибке* - не равен нулю (отключено).

Примечание: Параметры *Задержка между запросами*, *Количество повторных запросов при ошибке* и *Задержка между неудачными запросами* позволяют обрабатывать ситуации с ошибками из-за превышения допустимого количества запросов в единицу времени и из-за перегрузки сервиса. Повторные запросы отправляются, если:

- возникла сетевая ошибка;
- сервер вернул статус 408 (Request Timeout);
- сервер вернул статус 429 (Too Many Requests);
- сервер вернул любую из 5xx ошибок.

- **Общий тайм-аут (с)** — устанавливает максимально допустимое время выполнения всего запроса (в секундах).
- **Сохранять текст запроса на диск** — все запросы, отправленные сервису, записываются в указанную папку в файловом хранилище.
- **Имитировать запрос с ответом в файле** — при установке флага фактического запроса к сервису не происходит, а узел в качестве ответа сервиса подставляет содержимое указанного файла.

Описание узла

На данном этапе мастера настройки узла доступен выбор режима формирования метки узла:

Автоматическая метка — метка генерируется системой на основе имени файла.

Пользовательская метка задаётся вручную и не меняется автоматически. В этом режиме можно ввести любое текстовое описание. Рядом с полем ввода отображается значок обновления, который становится активным, если текущая пользовательская метка отличается от метки компонента по умолчанию. Нажатие на значок обновления восстанавливает метку компонента по умолчанию.

Режим формирования метки можно изменить как в мастере настройки узла (через выпадающий список в мастере настройки), так и через контекстное меню метки узла в сценарии (правой кнопкой мыши по метке узла).

У скопированного или клонированного узла сохраняется режим формирования метки.

Если у исходного узла метка формируется автоматически, то у скопированных и клонированных узлов не будет добавляться индекс.