

# Инструменты тестирования запросов

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

# Содержание

Postman	4
<b>Выполнить запрос к Creatio в Postman</b>	<b>4</b>
1. Добавить запрос	4
2. Настроить запрос	6
3. Выполнить запрос	8
4. Сохранить запрос	8
<b>Выполнить коллекцию запросов к Creatio в Postman</b>	<b>9</b>
1. Добавить коллекцию запросов	9
2. Добавить запрос в коллекцию	11
3. Настроить переменные коллекции запроса	11
4. Выполнить коллекцию запросов	12
Fiddler	15
<b>Выполнить запрос к Creatio в Fiddler</b>	<b>16</b>
1. Сформировать запрос	16
2. Выполнить запрос	18
Результат выполнения	19

# Postman



**Postman** — это набор инструментов для тестирования API. Он является средой разработки для создания, тестирования, контроля и публикации API-документации. **Назначение** Postman — тестирование отправки запросов с клиента на сервер и получения ответа от сервера.

Postman рекомендуется использовать для тестирования запросов при интеграции с Creatio по протоколу [OData 4](#) и [OData 3](#).

**Этапы** работы с запросами в Postman:

- Добавление запроса.
- Настройка запроса.
- Выполнение запроса.
- Сохранение запроса.

**Коллекция запросов** позволяет выполнять несколько запросов один за другим. Использование коллекций значительно сокращает время тестирования запросов.

Коллекция запросов к Creatio должна включать:

1. `post` -запрос к сервису аутентификации `AuthService.svc`. Работа с сервисом аутентификации подробно описана в статье [Аутентификация](#).
2. Пользовательский запрос для работы с данными.

**Этапы** работы с коллекциями запросов в Postman:

- Добавление коллекции запросов.
- Добавление запроса в коллекцию.
- Настройка переменных коллекции запроса.
- Выполнение коллекции запросов.

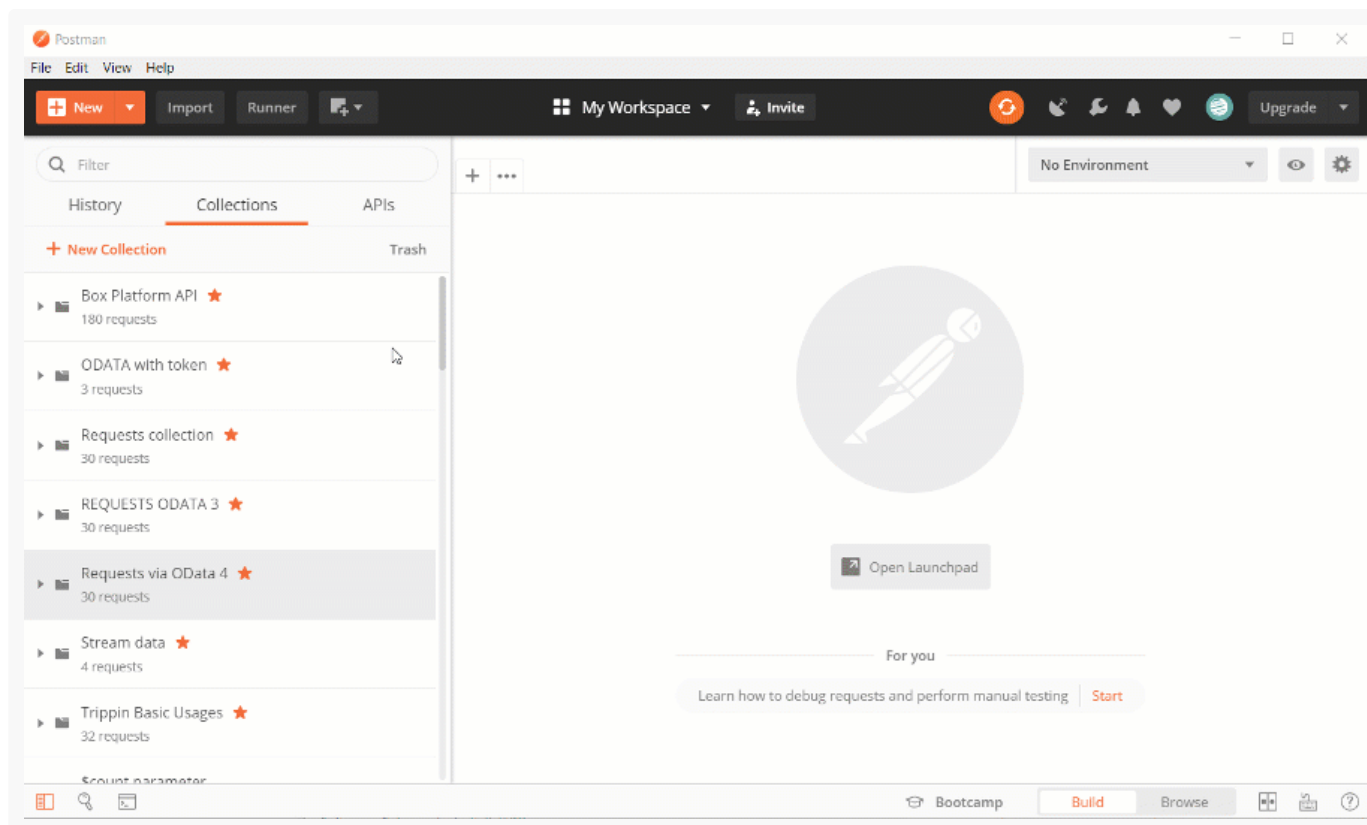
Работа с Postman подробно описана в официальной [документации Postman](#).

## Выполнить запрос к Creatio в Postman



### 1. Добавить запрос

1. На панели инструментов в Postman нажмите кнопку [ *New* ].
2. В блоке [ *Building blocks* ] вкладки [ *Create New* ] выберите [ *Request* ].

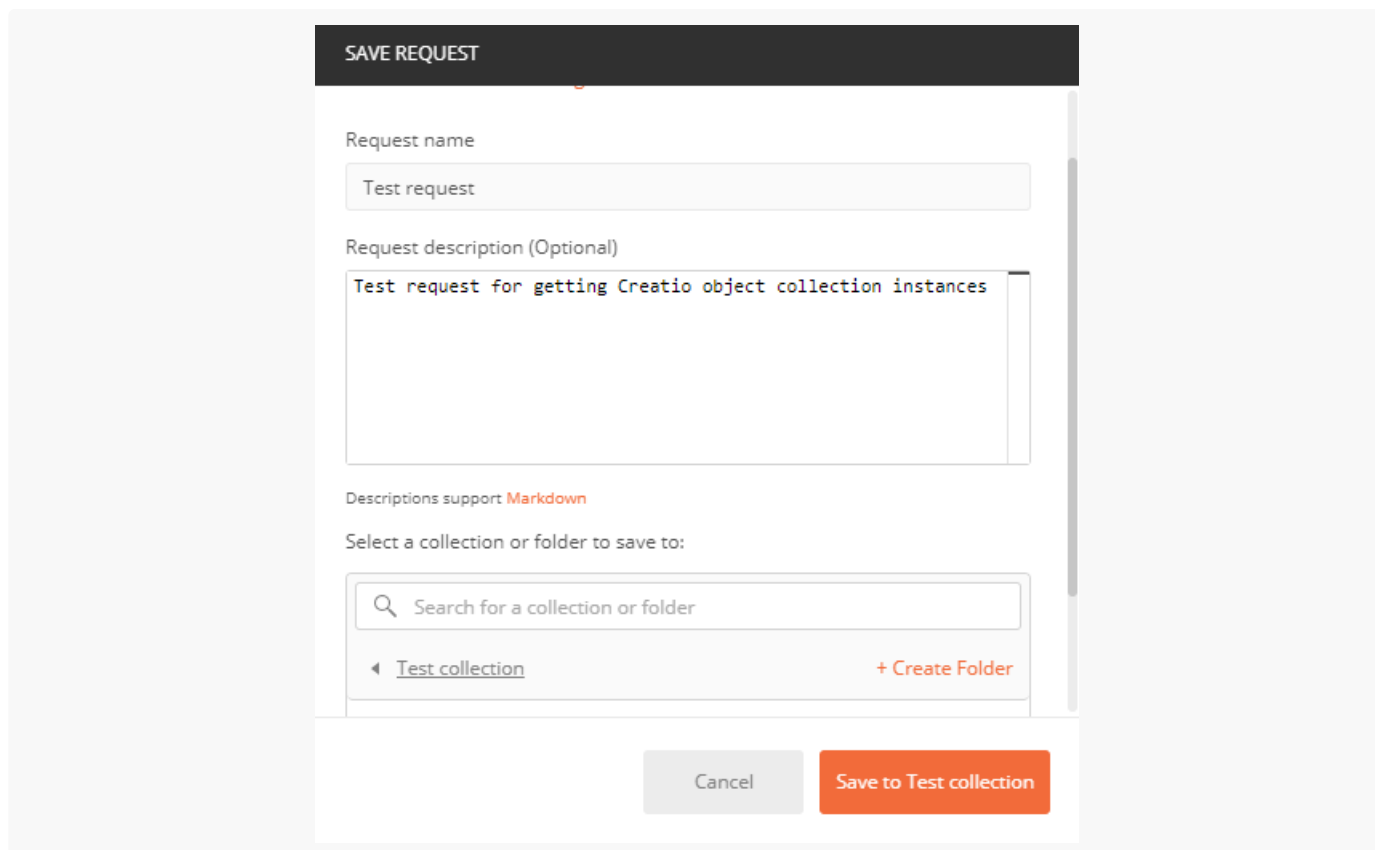


### 3. Заполните поля запроса.

Поля для создания запроса

Имя поля	Описание	Пример заполнения поля
<b>[ Request name ]</b>	Имя запроса.	Test request
<b>[ Request description (Optional) ]</b>	Описание запроса (при необходимости).	Test request for getting Creatio object collection instances
<b>[ Search for a collection or folder to save to: ]</b>	Поиск ранее созданной коллекции запросов или создание новой.	Test collection

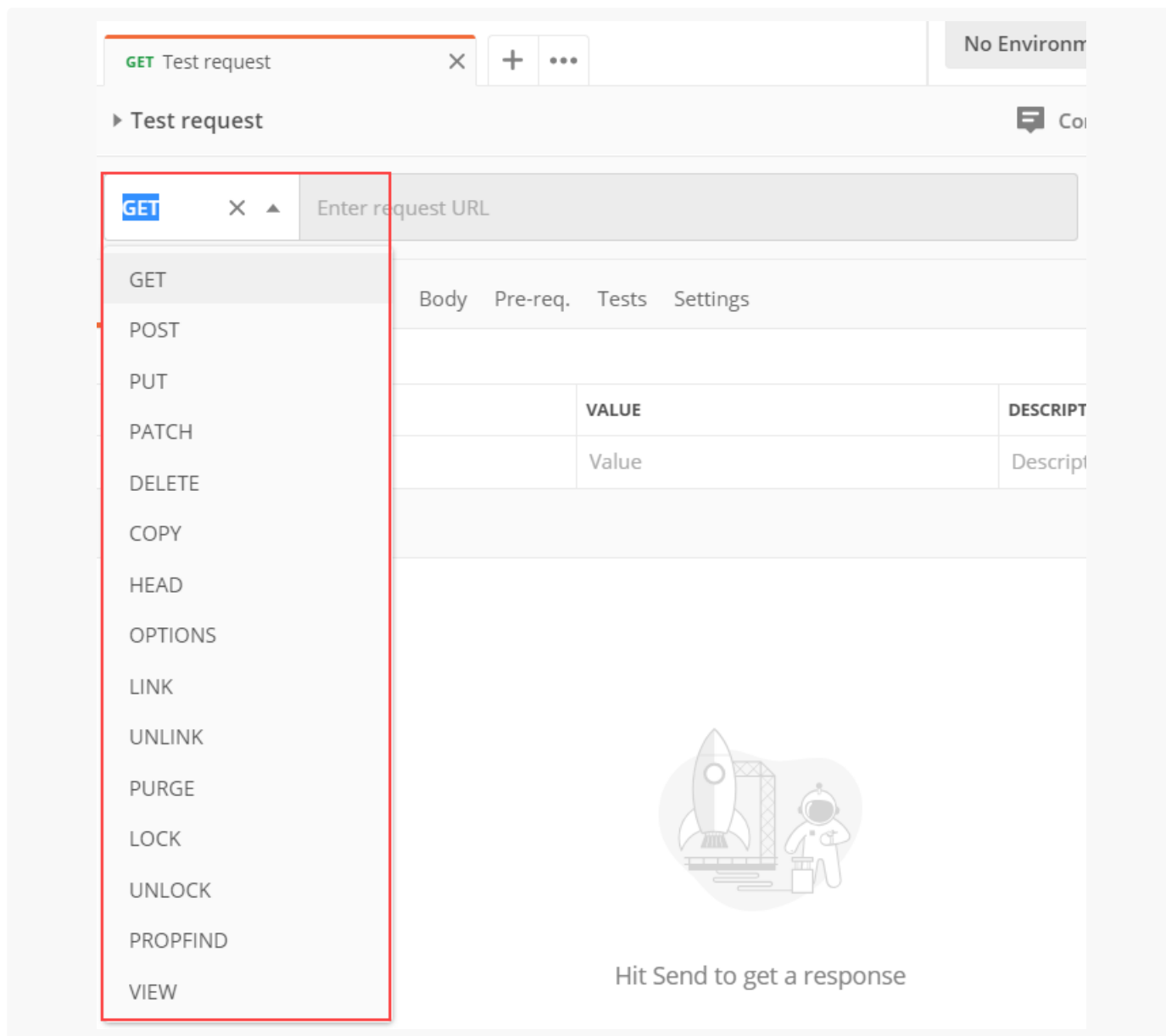
Окно создания запроса



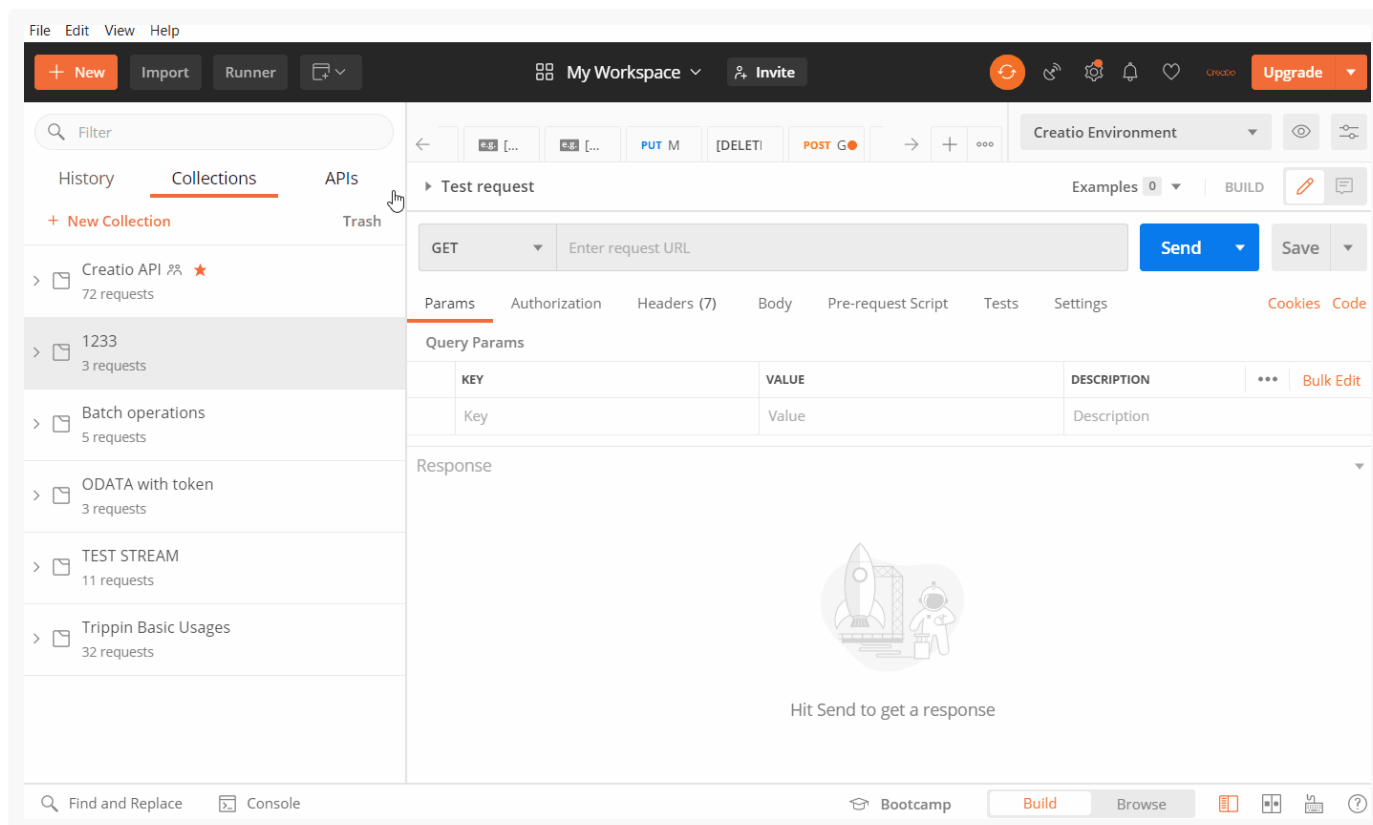
4. Нажмите [ Save ]. Кнопка становится активной после заполнения поля [ *Search for a collection or folder to save to:* ].

## 2. Настроить запрос

1. В выпадающем списке панели инструментов рабочей области Postman выберите метод запроса.



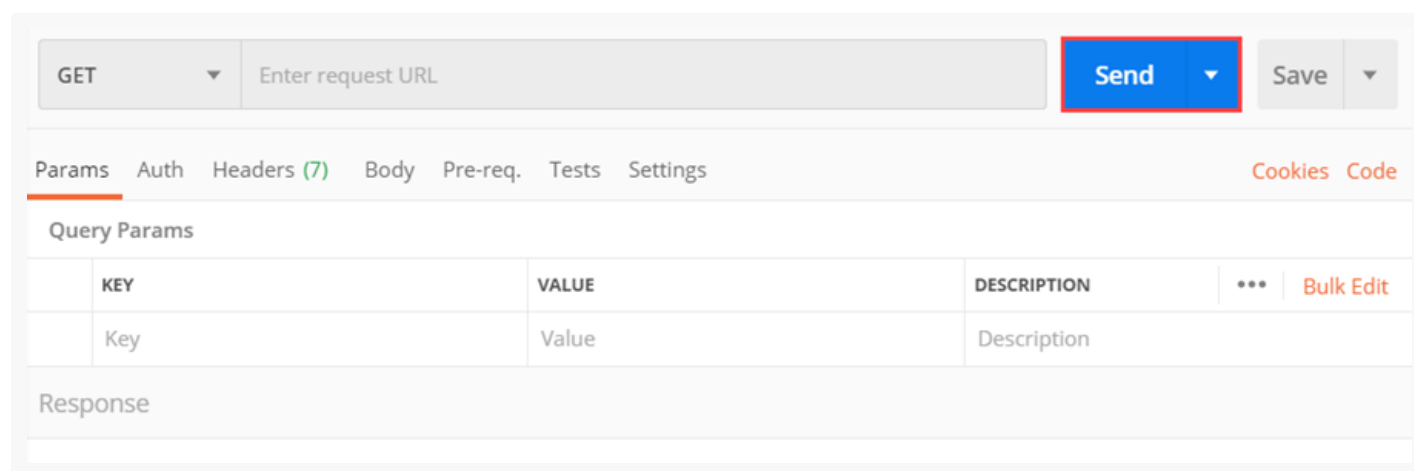
2. В поле запроса панели инструментов рабочей области Postman введите строку запроса.
3. Установите **формат данных** запроса:
  - a. Перейдите на вкладку [ *Body* ].
  - b. Установите опцию "raw".
  - c. Выберите тип "JSON".



4. В рабочей области Postman перейдите на вкладку [ *Body* ] и для методов `POST` и `PATCH` заполните тело запроса.
5. В рабочей области Postman перейдите на вкладку [ *Headers* ] и установите заголовки.

### 3. Выполнить запрос

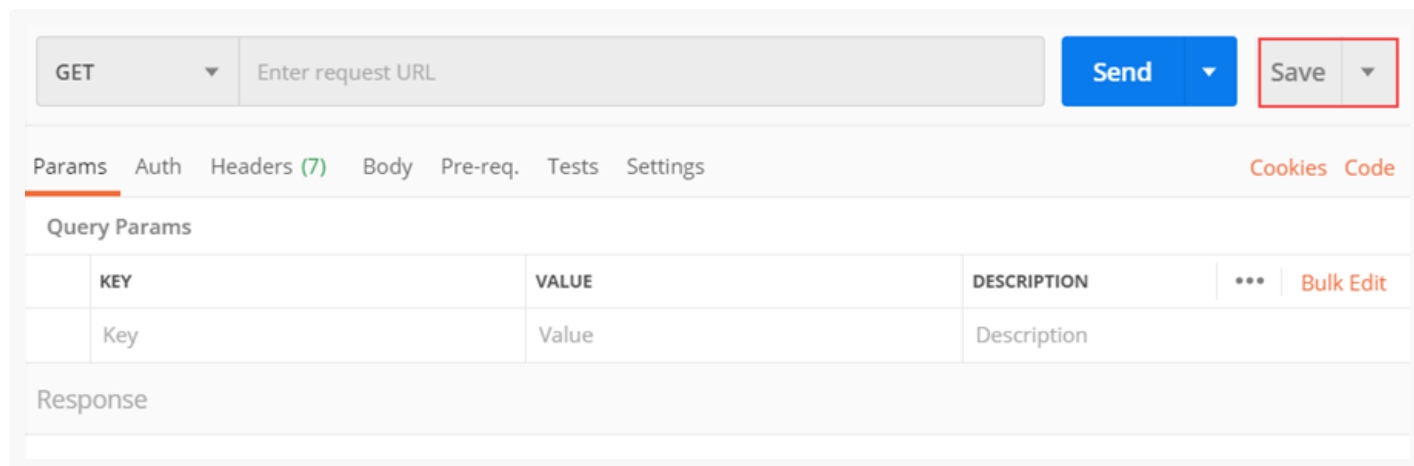
Чтобы **выполнить запрос** в Postman, на панели инструментов рабочей области нажмите [ *Send* ].



### 4. Сохранить запрос

Чтобы **сохранить запрос** в Postman, на панели инструментов рабочей области нажмите [ *Save* ].





Работа с Postman подробно описана в официальной [документации Postman](#).

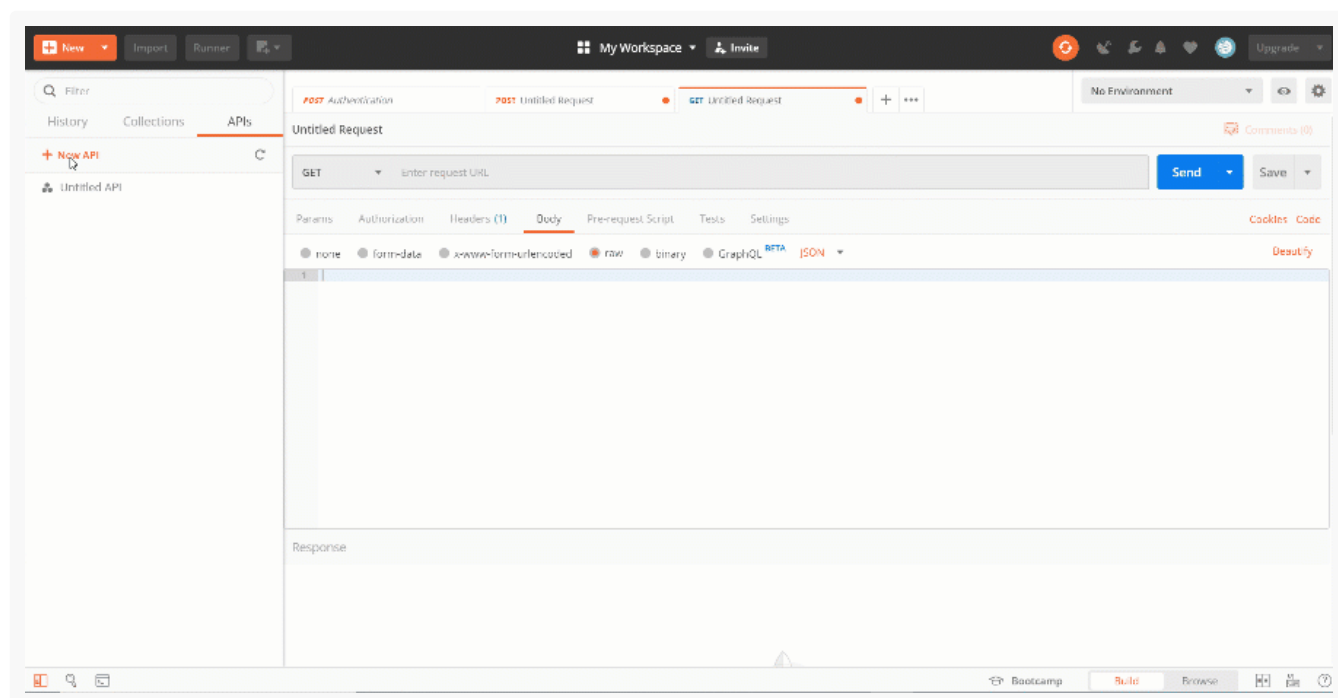
# Выполнить коллекцию запросов к Creatio в Postman

 Средний

## 1. Добавить коллекцию запросов

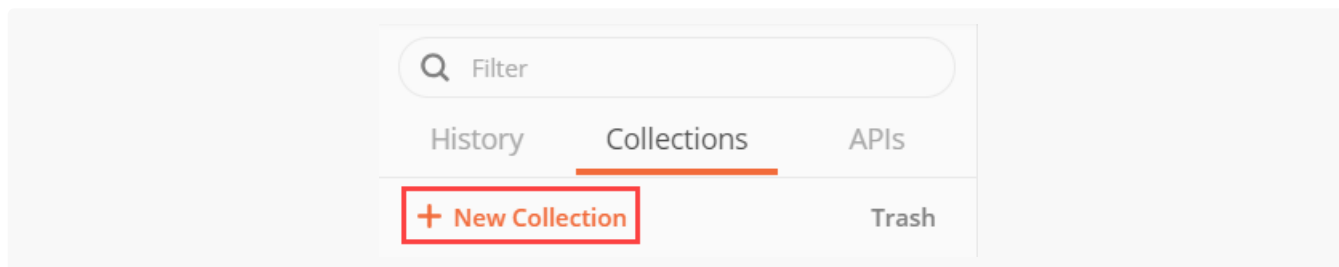
1. Создайте коллекцию запросов. **Способы** создания коллекции запросов в Postman:

- На панели инструментов в Postman нажмите кнопку [ *New* ]. В блоке [ *Building blocks* ] вкладки [ *Create New* ] нажмите [ *Collection* ].



- На панели работы с запросами в Postman перейдите на вкладку [ *Collections* ]. Нажмите [ + *New* ]

Collection ].



## 2. Заполните поля коллекции запросов.

Поля для создания коллекции запросов

Имя поля	Описание	Пример заполнения поля
[ <b>Name</b> ]	Имя коллекции.	Test collection
[ <b>Description</b> ]	Описание коллекции (при необходимости).	Test collection for test requests

Окно создания коллекции запросов

CREATE A NEW COLLECTION

Name

Test collection

Description

Authorization

Pre-request Scripts

Tests

Variables

This description will show in your collection's documentation, along with the descriptions of its folders and requests.

Test collection for test requests

Descriptions support **Markdown**

Cancel

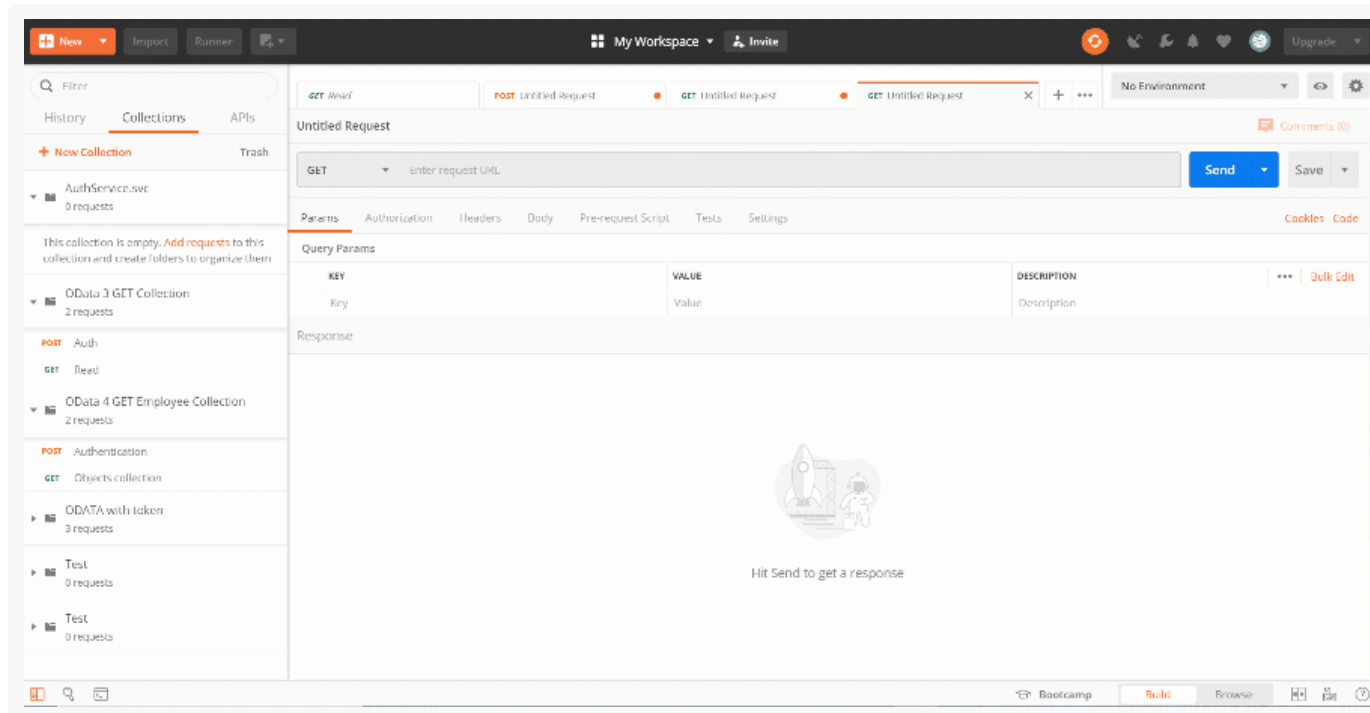
Create

## 3. Нажмите [ Create ].

## 2. Добавить запрос в коллекцию

**Способы** добавления запроса в коллекцию Postman:

- На панели работы с запросами в Postman перетащить курсором мыши ранее созданный запрос в коллекцию.
- На панели работы с запросами в Postman по нажатию правой кнопки мыши по имени созданной коллекции —> пункт [ *Add Request* ].

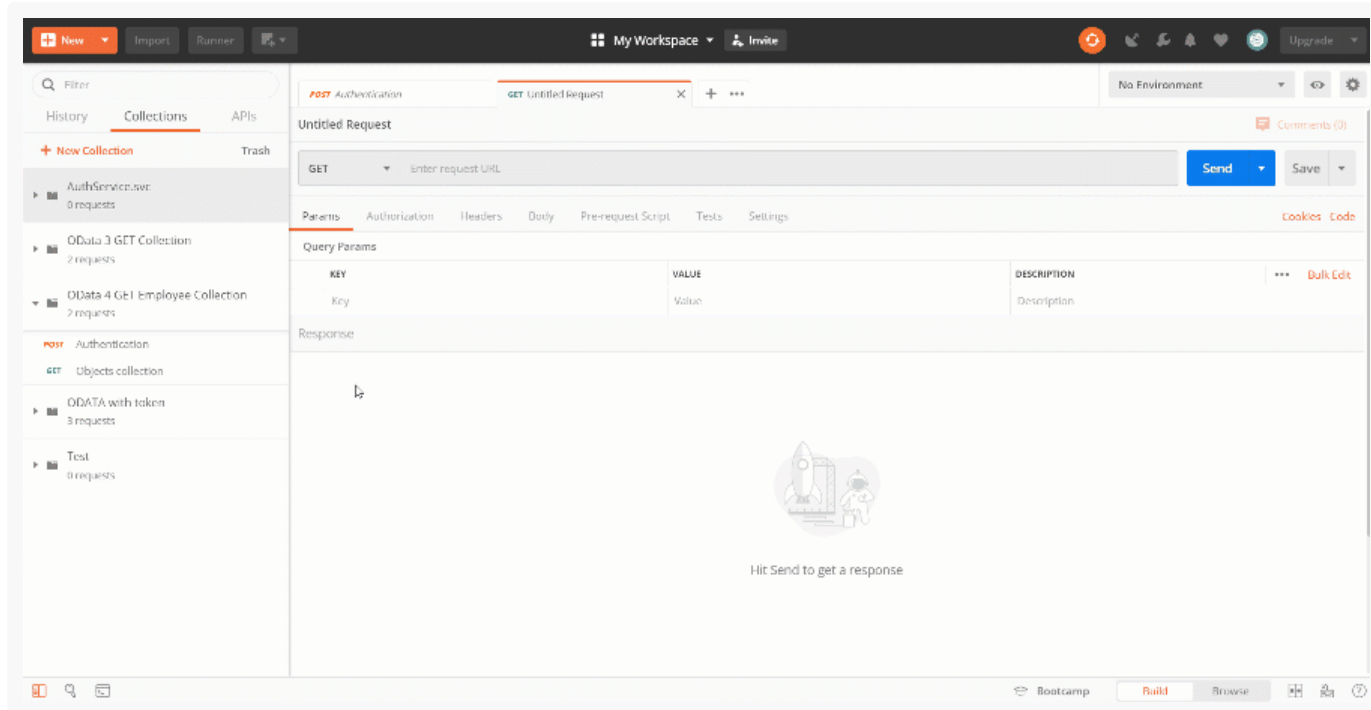


## 3. Настроить переменные коллекции запроса

Коллекция позволяет установить общие переменные и параметры для всех запросов коллекции.

Чтобы **настроить переменные** коллекции запроса в Postman:

1. На панели работы с запросами в Postman нажмите правой кнопкой мыши по имени созданной коллекции.
2. Выберите пункт [ *Edit* ] и перейдите на вкладку [ *Variables* ].




### 3. Создайте переменные коллекции.

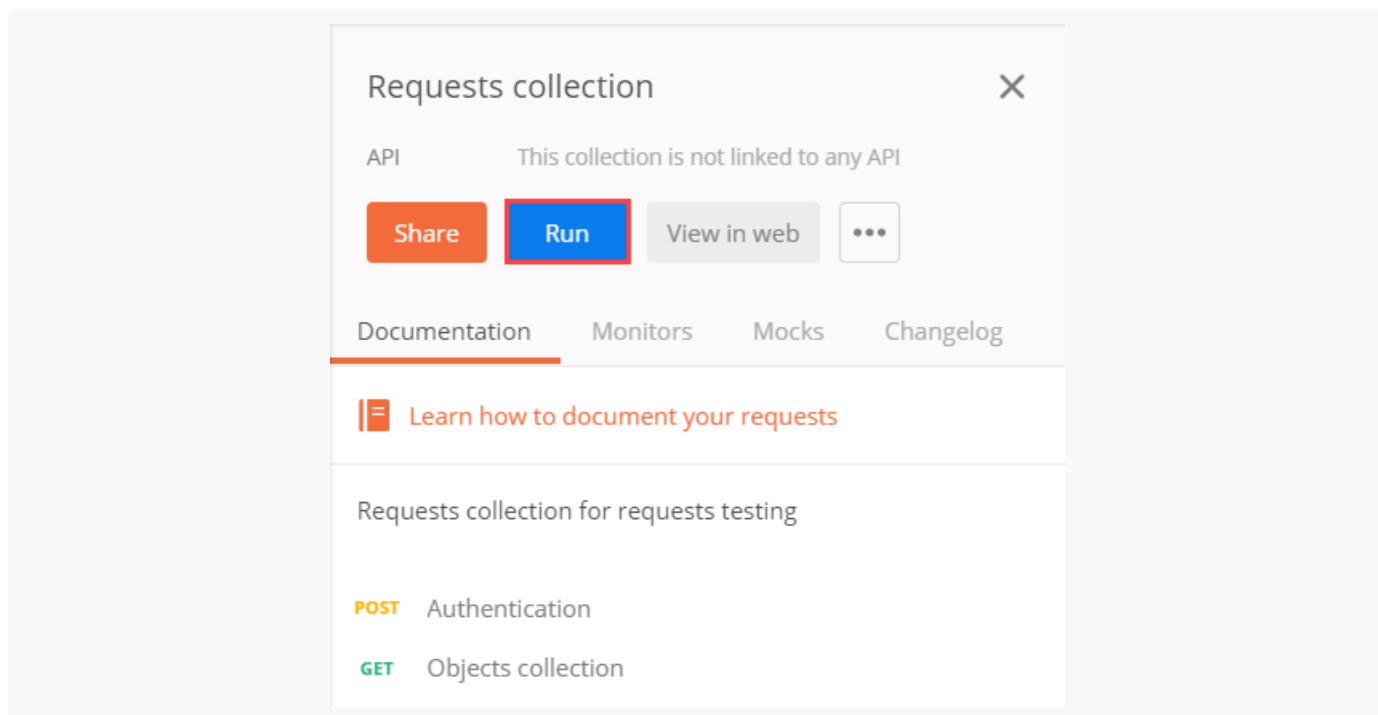
Переменные коллекции

Имя переменной	Описание
<code>BaseURI</code>	Адрес приложения Creatio.
<code>UserName</code>	Логин пользователя Creatio.
<code>UserPassword</code>	Пароль пользователя Creatio.
<code>BPMCSRF</code>	Токен для защиты от CSRF-атак.
<code>CollectionName</code>	Имя коллекции объектов (таблицы базы данных), с которой необходимо работать.

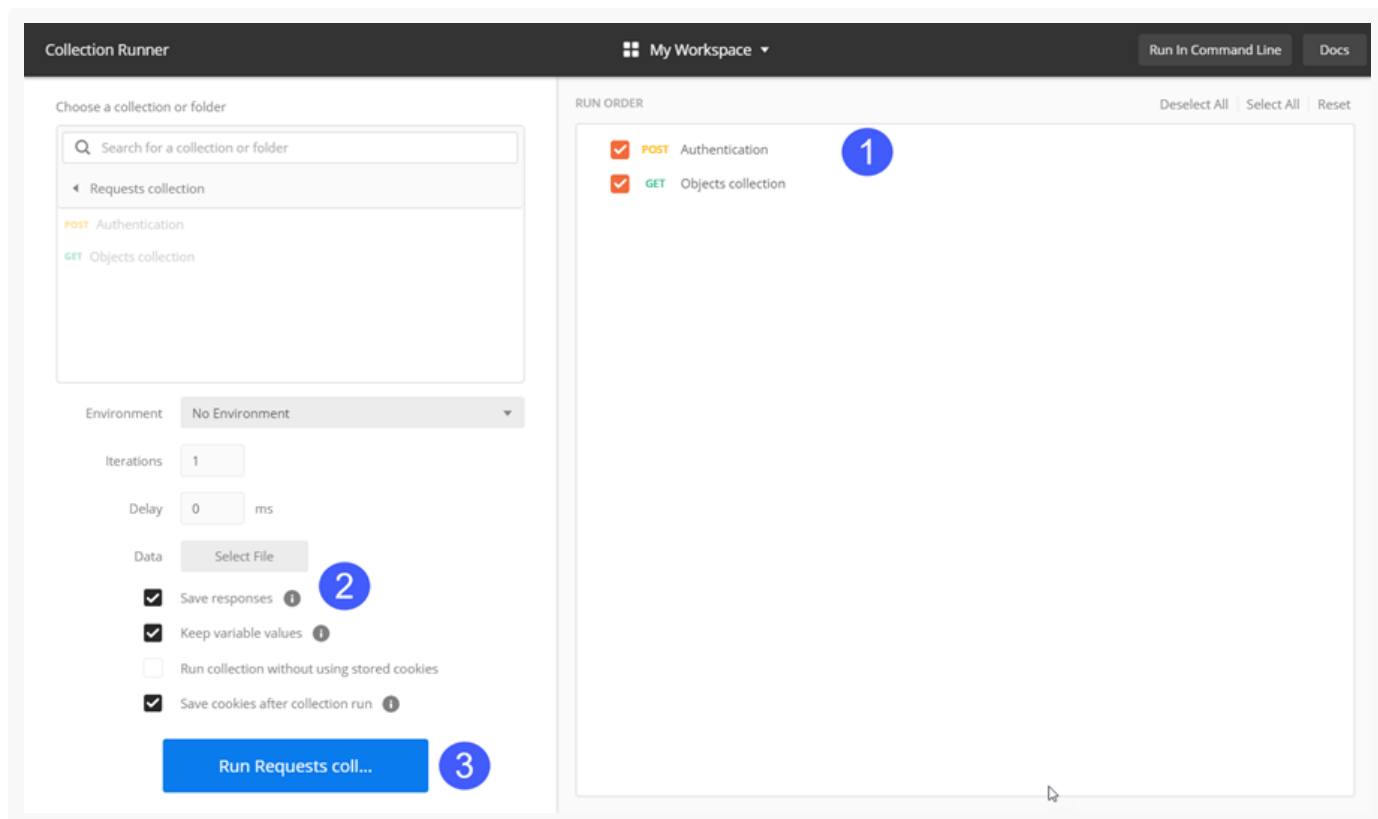
### 4. Продублируйте значения переменных в колонках [ *Initial value* ] и [ *Current value* ].

## 4. Выполнить коллекцию запросов

1. На панели работы с запросами в Postman рядом с именем коллекции нажмите кнопку .
2. Нажмите кнопку [ *Run* ].



3. В блоке [ *Run order* ] выберите запросы, которые необходимо запустить, и настройте порядок их выполнения (1).
4. Установите признак [ *Save responses* ] (2).
5. Нажмите [ *Run* ] (3).



После выполнения коллекции запросов Postman позволяет посмотреть структурные элементы каждого

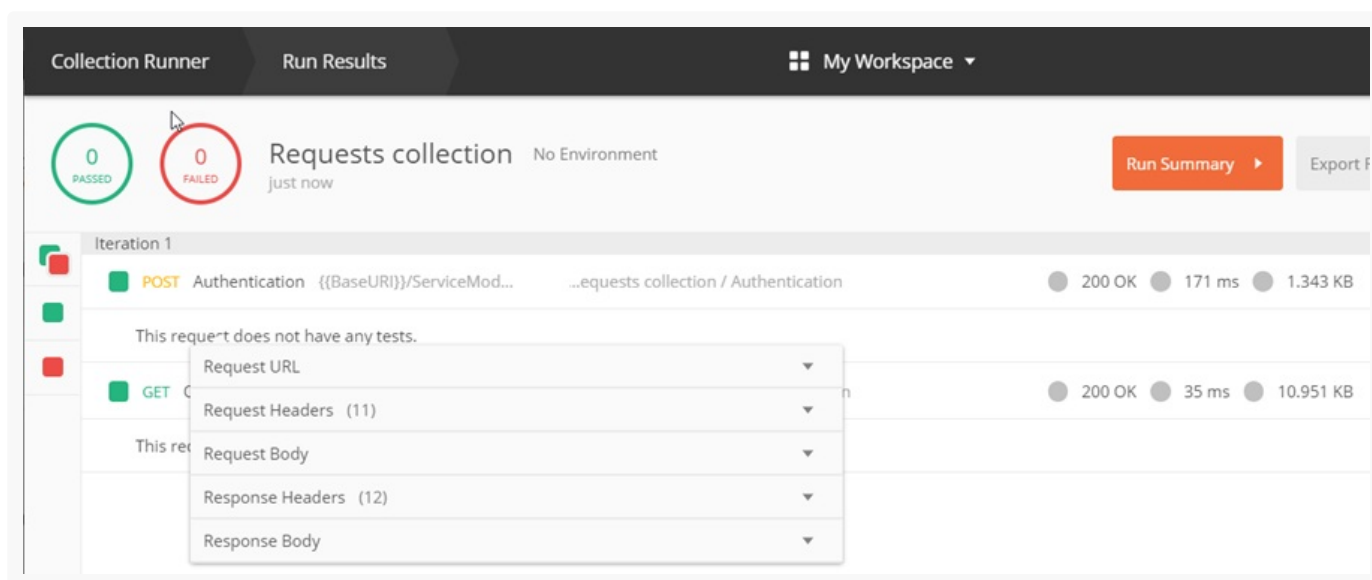
запроса коллекции.

Структурные **элементы запроса**:

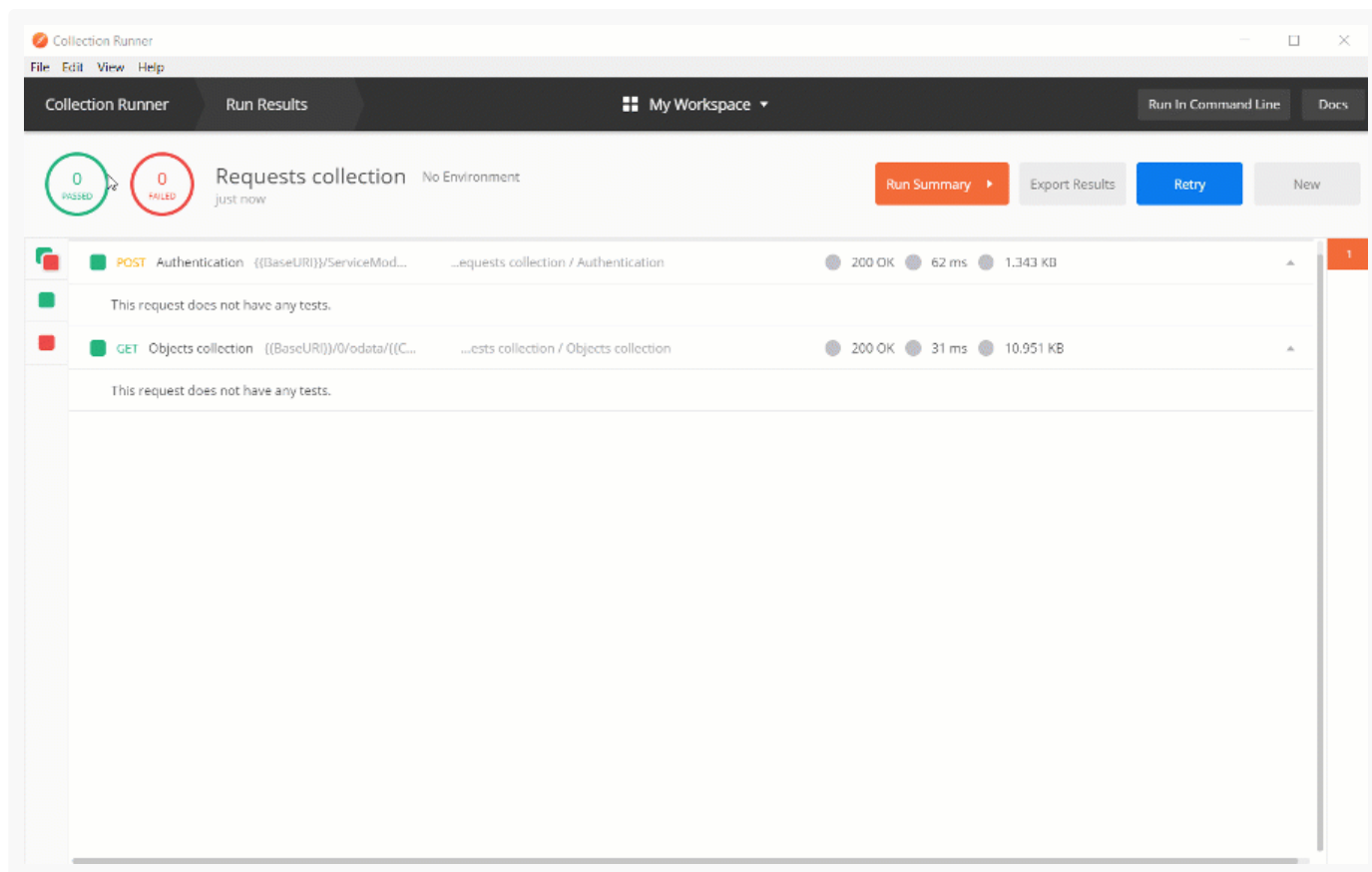
- Строка запроса.
- Заголовки запроса.
- Тело запроса (опционально).
- Заголовки ответа на запрос.
- Тело ответа на запрос (опционально).

**Способы** отображения информации структурных элементов запроса:

- Кликнуть по имени запроса.



- В меню [ View ] выбрать [ Show Postman Console ]. Затем выбрать структурный элемент запроса, информацию по которому необходимо посмотреть.



- Нажать комбинацию клавиш `Alt+Ctrl+C`.

Работа с Postman подробно описана в официальной [документации Postman](#).

## Fiddler

 Средний

**Fiddler** — это приложение прокси-сервера для регистрации, проверки и изменения трафика HTTP и HTTPS между компьютером и веб-сервером или серверами. **Назначение** Fiddler — предоставлять разработчикам возможность проверки запросов и ответов, включая HTTP-заголовки и метаданные (например, файлы cookie, кэширование и кодирование информации).

Fiddler рекомендуется использовать для тестирования запросов при интеграции с Creatio по протоколу [OData 4](#) и [OData 3](#).

**Этапы** работы с запросами в Fiddler:

- Формирование запроса.
- Выполнение запроса.

**Коллекция запросов** позволяет выполнять несколько запросов один за другим. Использование коллекций значительно сокращает время тестирования запросов.

Коллекция запросов к Creatio должна включать:

1. `POST`-запрос к сервису аутентификации `AuthService.svc`. Работа с сервисом аутентификации подробно описана в статье [Аутентификация](#).

2. Пользовательский запрос для работы с данными.

# Выполнить запрос к Creatio в Fiddler


 Средний

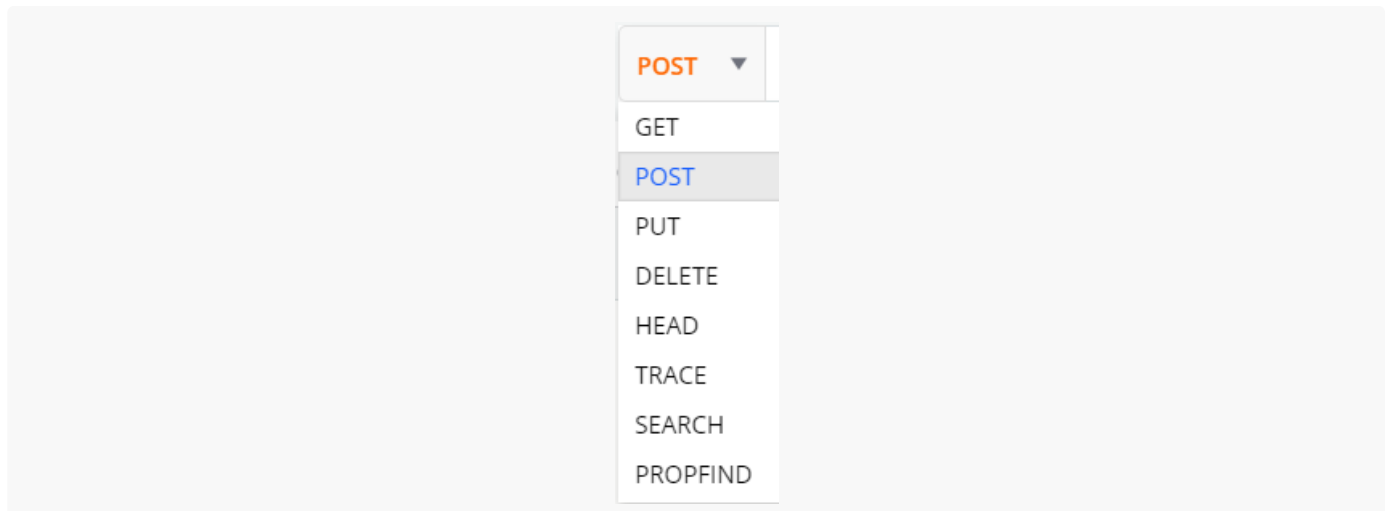
**Пример.** Используя сервис odata, добавить новую активность с колонками:

- [ Заголовок ] ([ Title ]) — "Process the incoming website form request".
- [ Ответственный ] ([ Owner ]) — [ Id ] текущего пользователя.
- [ Заметки ] ([ Notes ])—  
"Please, email to client@gmail.com and process the following request: clients request".

## 1. Сформировать запрос

Для формирования **запроса на добавление данных** в Fiddler:

1. Создайте новый запрос, нажав кнопку  [ New Request ].
2. В выпадающем списке панели инструментов рабочей области Fiddler выберите метод запроса `POST`.



3. В поле запроса панели инструментов рабочей области Fiddler введите строку запроса:

### Пример строки запроса

`http(s)://mycreatio.com/0/odata/Activity/`



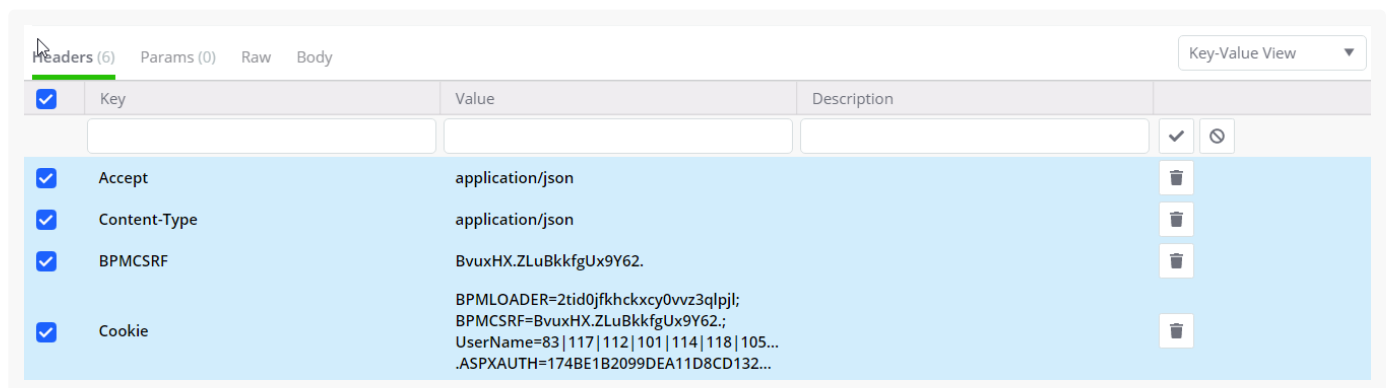


4. Выберите версию 1.1 протокола HTTP.
5. Перейдите на вкладку [ *Headers* ] и заполните заголовки запроса:

- `Content-Type` : `application/json` .
- `Accept` : `application/json` .
- `Cookie` : `BPMLOADER` , `.ASPXAUTH` , `BPMSESSIONID` и `BPMCSRF` — cookie получены из запроса к сервису аутентификации.
- `BPMCSRF` : "CSRF-токен BPMCSRF" — содержит значение одноименной cookie.

### Пример HTTP-заголовка запроса

```
Accept:application/json
Content-Type:application/json
Cookie: BPMSESSIONID=cxa54p2dsb4wnqbbzvgyxcoo; BPMCSRF=6yCmyILSlIE8/toyQm9Ca.; BPMLOADER=rqqj
BPMCSRF: 6yCmyILSlIE8/toyQm9Ca.
```



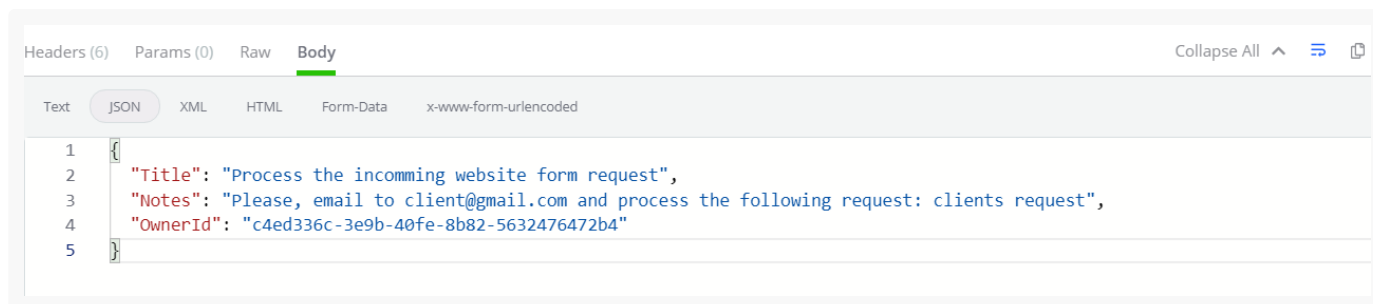
The screenshot shows the 'Headers' tab in Fiddler. It displays a table with columns: Key, Value, and Description. There are four headers listed: Accept, Content-Type, BPMCSRF, and Cookie. Each header has a checkbox on the left and a trash icon on the right. The values are as specified in the example above.

Key	Value	Description
Accept	application/json	
Content-Type	application/json	
BPMCSRF	BvuxHX.ZLuBkkfgUx9Y62.	
Cookie	BPMLOADER=2tid0jfkxckxycy0vvz3qlpjl; BPMCSRF=BvuxHX.ZLuBkkfgUx9Y62.; UserName=83 117 112 101 114 118 105... .ASPXAUTH=174BE1B2099DEA11D8CD132...	

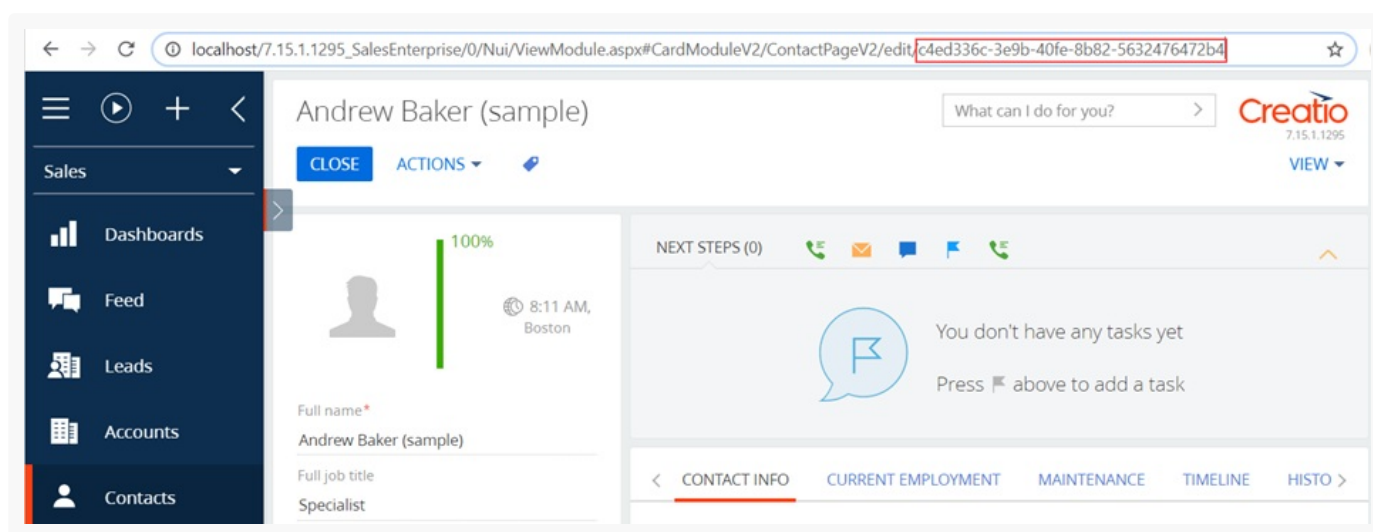
6. Перейдите на вкладку [ *Body* ] и заполните тело запроса.

### Пример тела запроса

```
{
  "Title": "Process the incoming website form request",
  "Notes": "Please, email to client@gmail.com and process the following request: clients requ",
  "OwnerId": "c4ed336c-3e9b-40fe-8b82-5632476472b4"
}
```

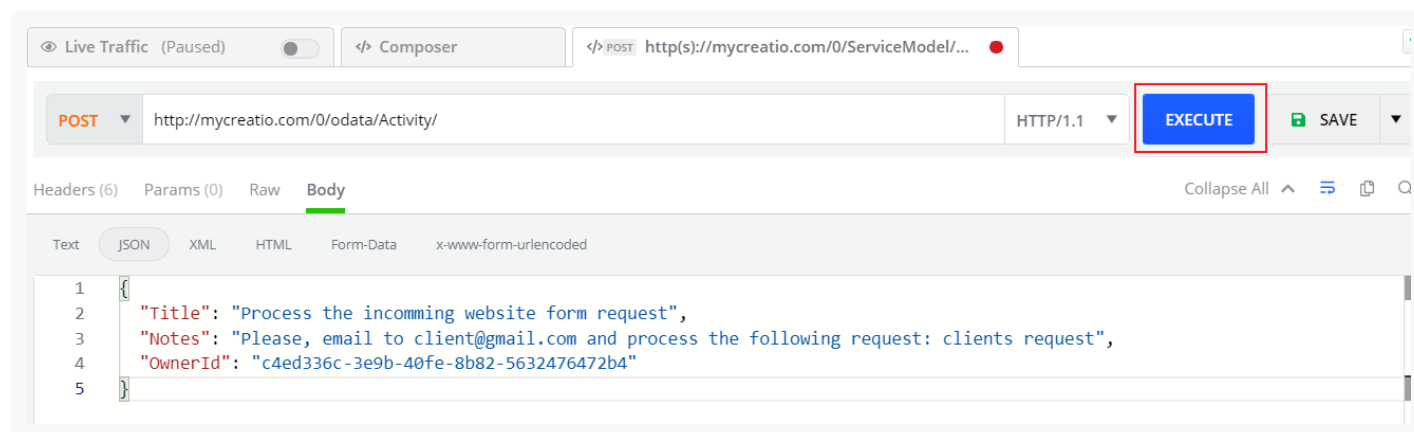


Для справочной колонки в запросе необходимо к названию колонки добавить суффикс Id. В качестве значения указать идентификатор из таблицы базы данных. В приведенном примере используется справочная колонка [ Ответственный ] ([ Owner ]), для которой в запросе указывается идентификатор `OwnerId`. Значение идентификатора ответственного можно посмотреть в браузере, открыв соответствующую страницу контакта.



## 2. Выполнить запрос

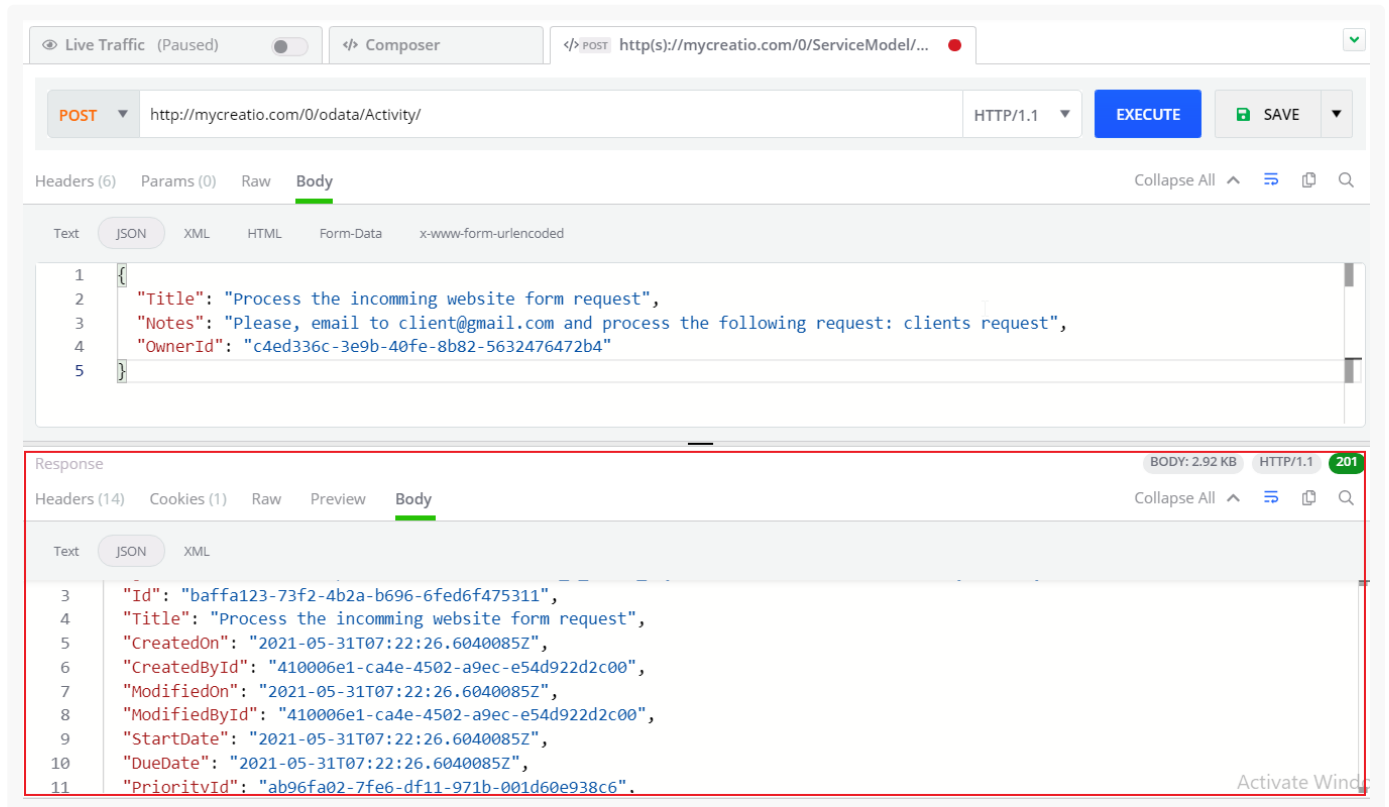
Чтобы выполнить запрос в Fiddler, на панели инструментов рабочей области нажмите кнопку [ Execute ].



## Результат выполнения

В результате выполнения запроса:

- В блоке [ *Response* ] отобразится код состояния ответа. На вкладке [ *Body* ] отобразится ответ от сервиса `odata`.



В теле ответа содержится добавленная запись в формате JSON. Элемент `"Id"` содержит идентификатор добавленной активности, который можно использовать при составлении других запросов, например, на редактирование.

- В раздел [ *Активности* ] ([ *Activities* ]) будет добавлена новая запись.

The screenshot displays the Creatio user interface. On the left is a dark blue navigation sidebar with icons and labels for 'Dashboards', 'Employees', 'Contacts', 'Accounts', 'Activities' (highlighted with a red bar), and 'Feed'. The main content area is titled 'Activities' and features a search bar with the placeholder text 'What can I do for you?'. Below the search bar are buttons for 'NEW TASK' (green) and 'ACTIONS' (blue with a dropdown arrow). A filter bar shows a date range '5/25/2021 till 5/25/2021', a user filter 'Employee', and options for 'Filters/folders' and 'Tag'. The task list contains one entry: 'process the incoming website form request'. This entry is enclosed in a red rectangular box. Below the task title, the details are organized into columns: 'Owner' (Andrew Baker (sample)), 'Due' (5/25/2021 2:14 AM), and 'Status' (Not started). The 'Category' is listed as 'To do'. On the right side of the interface, there is a vertical toolbar with icons for user profile, settings, help, email, chat, and notifications (indicated by a red circle with the number 3).

Owner	Due	Status	Category
Andrew Baker (sample)	5/25/2021 2:14 AM	Not started	To do