

# Postman2Confluence

- 0. Глоссарий
- 2. Использование
  - 2.1 Добавление дополнительной информации в метод
  - 2.2 Добавить страницу в игнор
- 3. Подготовка
- 4. Создание

## 0. Глоссарий

- **Postman** - это инструмент, который позволяет создавать и выполнять запросы, документировать и мониторить сервисы в одном месте.
- **JSON** (англ. JavaScript Object Notation) - текстовый формат обмена данными, основанный на JavaScript.
- **Template Toolkit** (TT) - мощная «легковесная» perl-библиотека для работы с шаблонами, позволяющая разделять код, данные и представление.
- **Репозиторий** — место, где хранятся и поддерживаются какие-либо данные.

## 1. Ссылки

- [Postman - Цифровой Университет](#)

Разработчик самостоятельно тестирует API методы в процессе работы. Для этого можно использовать Postman.

После завершения работы над задачей, обновлённые методы из Postman можно выгрузить в виде json. Скрипт берёт json и по заданному шаблону (TT) загоняет данные в confluence.

Данный способ позволяет сэкономить время и упростить работу:

- разработчику
- тестировщику
- фронтенд разработчику

Из-за простоты создания документации бекенд-разработчик с большей вероятностью будет следить за актуализацией документации. Благодаря шаблонам, документация будет всегда в одном формате. При изменении формата можно будет повторить операцию со скриптом, чтобы не вносить ручную правки в confluence.

Выгруженные данные из Postman, для сокращения времени на актуализацию методов, можно передать коллеге по работе (см. [Экспорт /Импорт](#) коллекций)

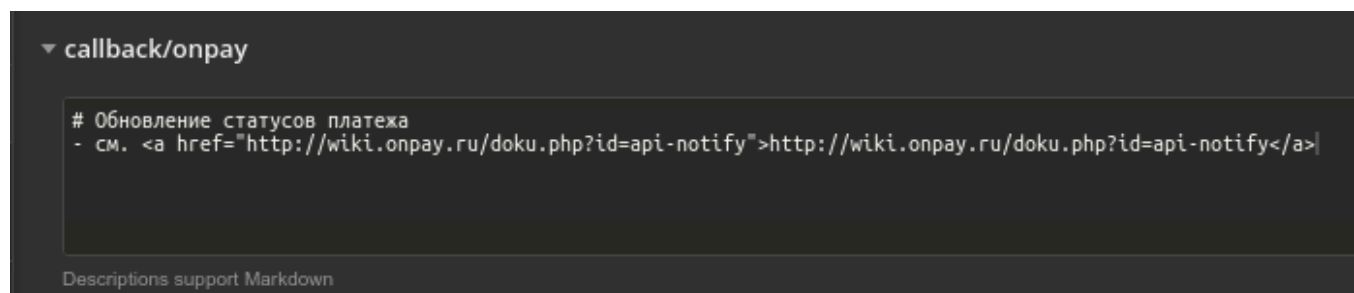
## 2. Использование

### 2.1 Добавление дополнительной информации в метод

Если нужно добавить доп. инфу в метод, так чтобы ее было видно, нужно

1. В описание к методу, добавляется
  - a. В описание, первым символом - "#"
  - b. Каждая след строка через символ "-"

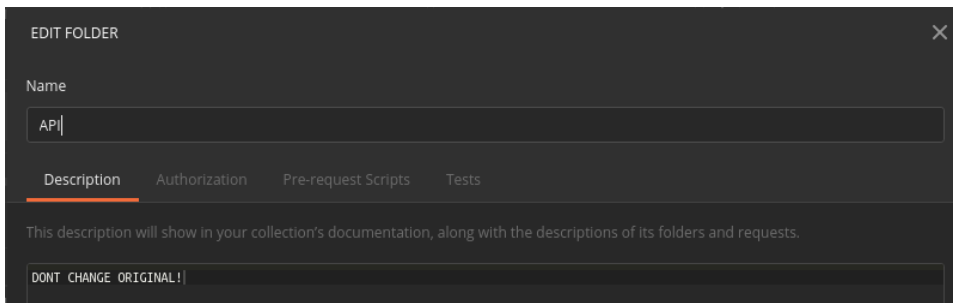
Пример - [Callback/OnPay - Обновление статусов платежа](#)



### 2.2 Добавить страницу в игнор

При заливке страницы в confluence, она создается заново ( создается новая версия), при этом внесенные вручную изменения на странице затрутсся. Во избежание этого можно добавить страницу в игнор. Например:

1. В описание к странице добавляется строка "DONT CHANGE ORIGINAL!". Пример - [API](#)



EDIT FOLDER

Name

API

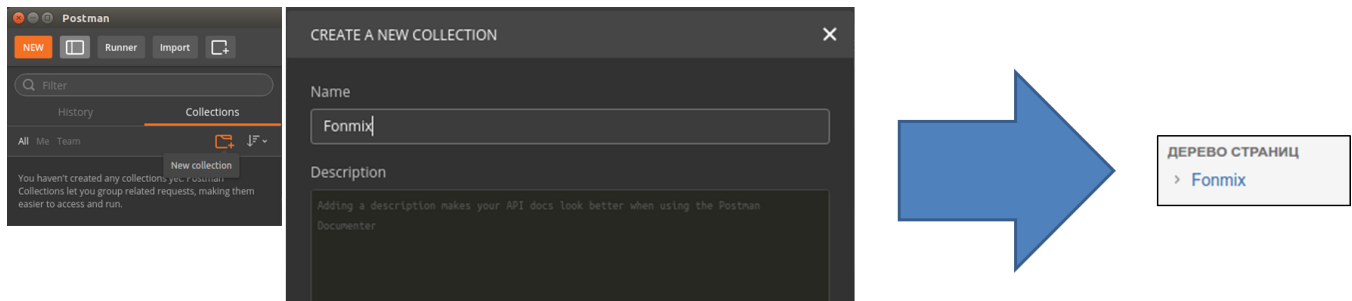
Description Authorization Pre-request Scripts Tests

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

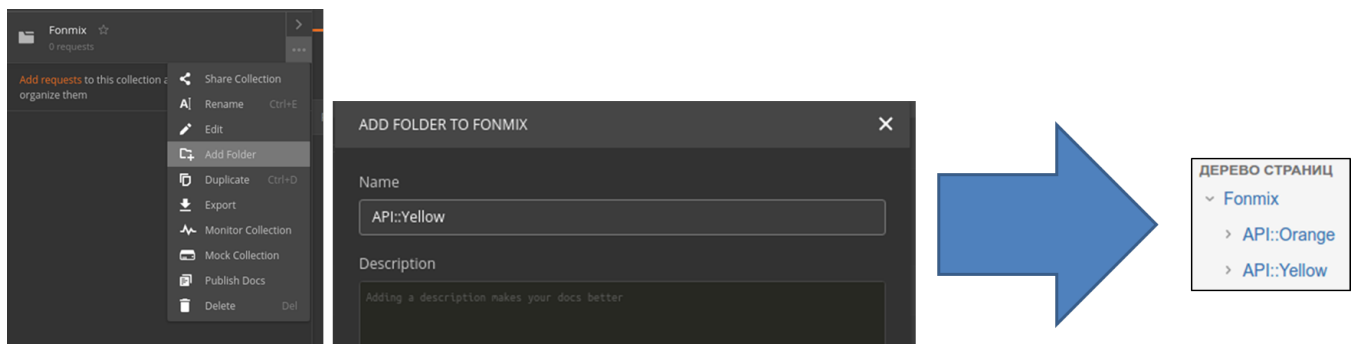
DONT CHANGE ORIGINAL!

### 3. Подготовка

- 3.1. Создать корневую страницу от которой будут наследоваться остальные



- 3.2. Создание структуры



- 3.3. Создание запросов

- 3.3.1 Сохранить запрос и добавить описание



POST

http://dev.qa.fonmix.ru/api/yellow/base/contract/add

Params

Send

Save

SAVE REQUEST

×

Requests in Postman are saved in collections (a group of requests).  
[Learn more about creating collections](#)

Request name

http://dev.qa.fonmix.ru//api/yellow/base/contract/add

Request description (Optional)

Adding a description makes your docs better

Descriptions support Markdown

Select a collection or folder to save to:

Q Search for a collection or folder

< API::Yellow::Base::Contract + Create Folder

add

•

+

...

▼ add ✎

Создание нового договора

POST ▼ http://dev.qa.fonmix.ru//api/yellow/base/contract/add

Если нужно указать дополнительные параметры, то можно воспользоваться разметкой markdown

add

+
...

▼ add

```
# Создание нового договора
- у роли пользователя должен быть доступ
- что-то еще
```

Descriptions support Markdown

POST ▼

http://dev.qa.fonmix.ru//api/yellow/base/contract/add

### 3.3.2. Указать параметры запроса

При GET запросе параметры указываются во вкладке Params, при POST запросе - во вкладке Body/form-data

POST ▼

http://dev.qa.fonmix.ru//api/yellow/base/contract/add

Params

Send ▼

Authorization
Headers (1)
Body ●
Pre-request Script
Tests

☒ form-data
☐ x-www-form-urlencoded
☐ raw
☐ binary

	Key	Value	Description
<input checked="" type="checkbox"/>	number	test	text Номер договора да
<input checked="" type="checkbox"/>	type_id	1	int ID типа договора да
<input checked="" type="checkbox"/>	client_id	4	int ID юр. лица да

в поле description с разделителем pipe - | указываются:

- тип
- описание
- параметр обязательный - да/нет

### 3.3.3 Если нужен пример запроса, необходимо:

#### 3.3.3.1 Выполнить запрос

POST http://dev.qa.fonmix.ru/api/yellow/base/contract/add Params Send

Authorization Headers (1) Body Pre-request Script Tests

Key	Value	Description
<input checked="" type="checkbox"/> X-Secret-Key	mlkSXn6qfjN7EzzY3ajpCSveowdFji0E	
New key	Value	Description

Body Cookies (7) Headers (13) Test Results Status: 200 OK

Pretty Raw Preview JSON

```

1 {
2   "success": 1,
3   "data": {
4     "item": {
5       "json_custom": "",
6       "rate": "",
7       "client_id": "4",
8       "number": "test",
9       "ts_create": 1514551814,
10      "count_filials": "",
11      "id": "5a463a06-0000-0000-0000-000051481a61",
12      "type_id": "1",
13      "fonmix_crm_id": "",
14      "date_prepare": "",
15      "ts_modify": 1514551814,
16      "date_end": "",
17      "date_block": "",
18      "date_start": "",
19      "date_activation": "",
20      "sphinx_id": ""
21    }
22  },
23   "params": {
24     "number": "test",
25     "client_id": "4",
26     "type_id": "1"
27   }
28 }

```

### 3.3.3.2 Добавить в примеры

Examples (0)

No examples added

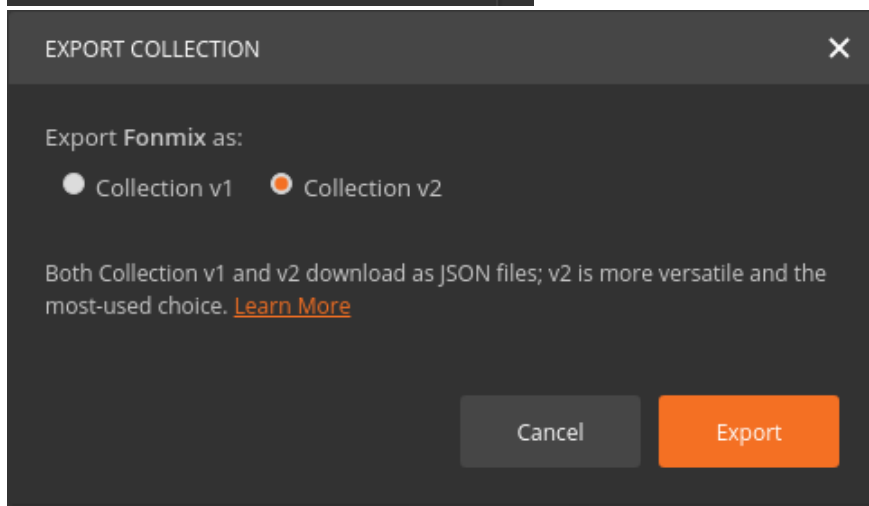
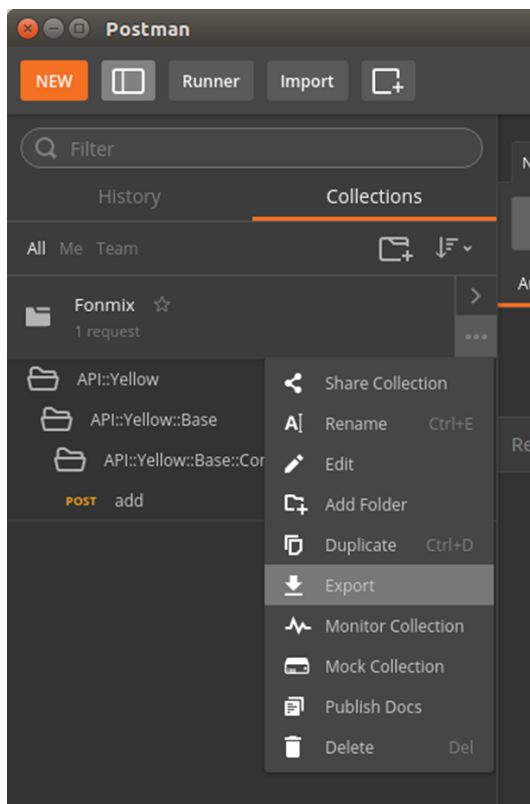
Save responses and associated requests as Examples. [Learn More](#)

Add Example

Save Example

Params

### 3.3.4 Экспортировать структуру во временную папку (нужно выбрать 2-ю версию)



## 4. Создание

Скрипт находится в репозитории - <https://gitlab.id-network.ru/backend/internal/lib-external>

```
perl projects/fonmix.ru/lib/External/script/postman2confluence.pl -login='<your login>' -password=<your password> -file=/home/orlov/doc/Fonmix.postman_collection.json -space='~y.orlov' -host='http://fonmix.ru'
```

где:

1. Обязательные параметры
  - perl - программа запуска
  - projects/fonmix.ru/lib/External/script/postman2confluence.pl - скрипт
  - -login='<your login>' - логин от confluence
  - -password=<your password> - пароль от confluence
  - -space='~y.orlov' - пространство в котором будут созданы страницы
  - -file=/home/orlov/doc/Fonmix.postman\_collection.json - путь до json файла, экспортированного из postman
2. Не обязательные параметры
  - -start\_page='[auto] API::Yellow::Cabinet' - раздел документации который нужно обновить
  - -template='fonmix' - шаблон который использует скрипт, по умолчанию fonmix. Шаблоны лежат в lib/External/script/template
  - -host='http://fonmix.ru' - заменяет в примерах хост на который был запрос



Важно использовать параметр `start_page` т.к. иначе скрипт будет обновлять всю документацию

PS. Для частого использования, все параметры скрипта можно перенести в конфиг, для этого

1. Создается yaml конфиг, пример

```
$ cat ./config-dir/dev/postman.yaml

Confluence:
  host: https://confluence.id-network.ru/
  # space: ~y.orlov
  space: FONMIX

Sign_in:
  login: <you login>
  password: <you pass>

host_replace: fonmix.ru
file: /home/orlov/doc/Tech.Doc.FonMix.1.0.postman_collection.json
# host_replace: crm.fonmix.ru
# file: /home/orlov/doc/Tech.Doc.FonMix.CRM-1.0.postman_collection.json
```

2. Устанавливается переменные окружения

```
export CONFIG_PROJECT=postman;
export CONFIG_DIR=/home/orlov/config-dir/dev;
```

По окончании работы скрипта будут созданы страницы в указанном пространстве.

**Пример:**

# API::Yellow::Base::Contract

Создатель Орлов Ярослав Андреевич, отредактировано 19 мин. назад

- [Создание нового договора](#)

## Создание нового договора

- метод: **POST**
- путь: **//api/yellow/base/contract/add**
- пользователь должен быть **авторизован** ([Доступ к сервису](#))
- у роли пользователя должен быть доступ
- что-то еще
- [входящие параметры](#)

Параметр	Тип	Описание	Обязательный
number	text	Номер договора	да
type_id	int	ID типа договора	да
client_id	int	ID юр. лица	да

- [пример](#)

### request

```
POST: http://fonmix.ru//api/yellow/base/contract/add
number: test
type_id: 1
client_id: 4
```

### response

```
1 {
2   "success" : 1,
3   "params" : {
4     "type_id" : "1",
```

## 5. Postman desktop (не готово)

Для выполнения авторизованных запросов, нужно авторизоваться на сайте и добавить в cookies - rsid



track/list/from/csv

dev

track/list/from/csv

POST

{{host}}/api/yellow/base/track/list/from/csv

Params

Send

Save

Authorization

Headers

Body

Pre-request Script

Tests

Cookies

Code

Key	Value	Description		Bulk Edit	Presets
New key	Value	Description			

Body

Cookies (2)

Headers (13)

Test Results

Status: 200 OK

Time: 80 ms

Size: 2.8 KB

Pretty

Raw

Preview

JSON

Save Response

MANAGE COOKIES

Type a domain name

Add

dev.qa.fonmix.ru 2 cookies

uid

Cookie\_2

+ Add Cookie

rsid=2091071f72f8ef115ed99fccf10fa692; path=/; domain=.dev.qa.fonmix.ru;

Cancel

Save