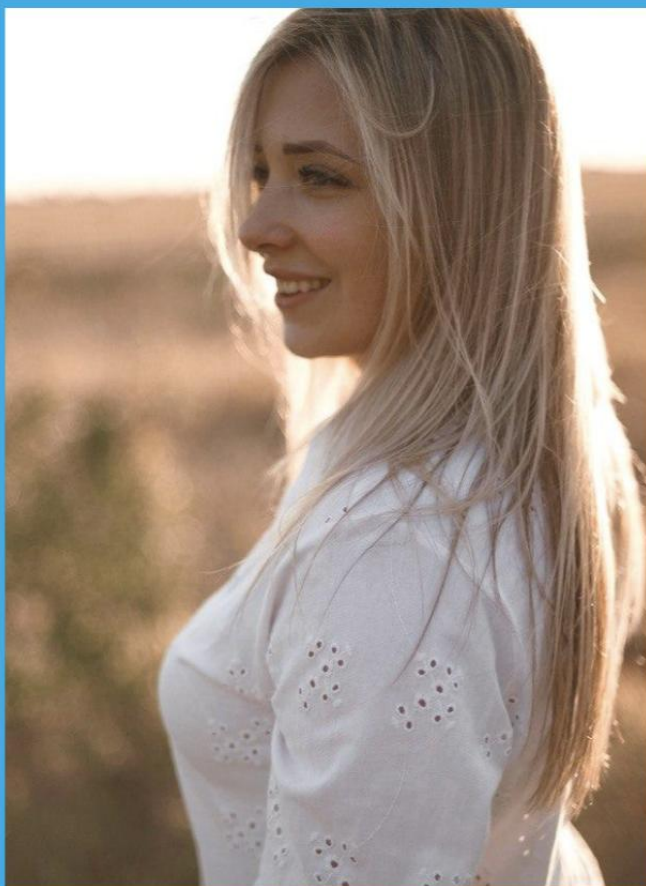




REST API

Основы и создание спецификации в Swagger

О преподавателе



**ШАШКОВА
ПОПИЦА**

**Бизнес- и системный
команды Dion.Видео
аналитик**
ХОЛДИНГ Т1

SIEMENS
energy

Atos

PRINZHORN
GROUP

VTB

БРИЛЛИАНТЫ
АЛРОСА

+T1

DJON

План



- Понятие API
- Структура REST API
- Методы REST API (основные POST, PATCH, PUT, GET, DELETE)
- Коды HTTP-ответов (1xx, 2xx, 3xx, 4xx, 5xx)
- Query-params
- Форматы передаваемых данных
- КВИЗ
- Знакомство с инструментом для проектирования спецификации openAPI в Swagger. Практика написания документации в YAML

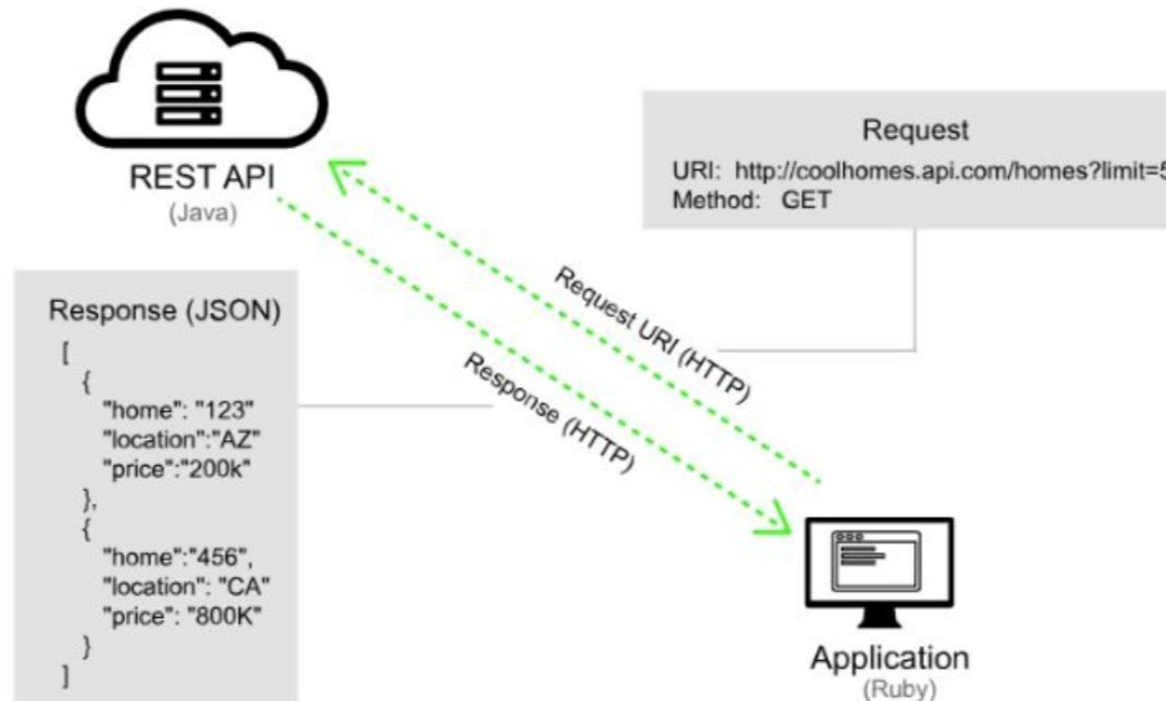


ОСНОВЫ REST API



+ API (Application Programming Interface) обеспечивает взаимодействие между двумя системами

+ REST API (Representational State Transfer Application Programming Interface) представляет собой **архитектурный** стиль для построения веб-сервисов, который базируется на принципах REST



Структура REST API методов



Запрос:

- + Header
- + Cookie
- + URL
- + Метод
- + Endpoint
- + Query params
- + Body

Ответ:

- + Header
- + Cookie
- + Body
- + Код HTTP ответа

▼ GET https://api.miro.com/v1/oauth-token

Request Headers

```
accept: "application/json"
Authorization: "Bearer eyJtaXJvLm9yaWdpbiI6ImV1MDEifQ_4Zf2tMCrAt55tUJZW0RYhprA_-Q"
User-Agent: "PostmanRuntime/7.36.1"
Cache-Control: "no-cache"
Postman-Token: "151de8f8-f555-495c-b01f-581e1b7dd647"
Host: "api.miro.com"
```

Cookie: "AWSALBTG=PoVGnitKstJR6g1tGqFpUPeM0vS/s5Z0m0MKLwowUFFq6LW5HYdFTv/LvqlWJFqRH+bi/jPNkktHXGgMvbzcUAXKhjvmIalVfJegL6mUpAgXL5I1/j+UFFq6LW5HYdFTv/LvqlWJFqRH+bi/jPNkktHXGgMvbzcUAXKhjvmIalVfJegL6mUpAgXL5I1/j+YyY1GFoi9swIFDA7nnstfqtAUE/0SUFUAzoY3IqMMJdSdh7vmGCP6iHwz"

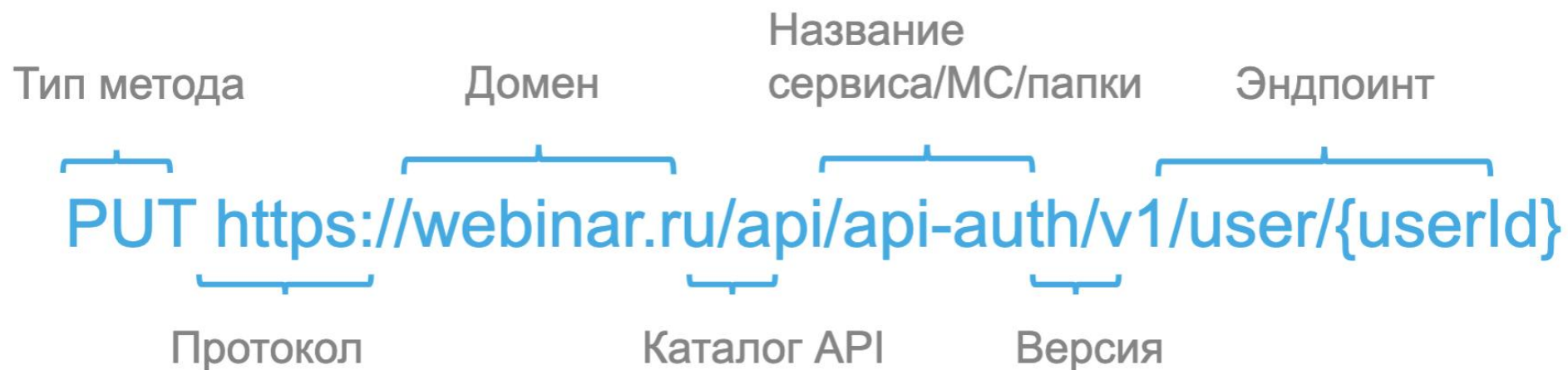
Response Headers

```
Content-Type: "application/json"
Transfer-Encoding: "chunked"
Connection: "keep-alive"
Date: "Sat, 27 Jan 2024 19:00:15 GMT"
Server: "nginx"
```

Response Body

```
{
  "status": 401,
  "code": "tokenNotProvided",
  "message": "No authorization data was found on the request",
  "type": "error"
}
```

Структура URL



Методы REST API



CRUD (Create, Read, Update, Delete) — это аббревиатура, которая отражает четыре основные функции, доступные для манипулирования данными в системе

GET

для **чтения** данных с сервера

- GET /users
- GET /users/123

POST

для **создания** нового ресурса

- POST /users

PUT

для **обновления** существующего ресурса

- PUT /users/123

PATCH

для **частичного** обновления ресурса

- PATCH /users/123

DELETE

для **удаления** ресурса с сервера

- DELETE /users/123

Категории HTTP ответов



REST API использует строку состояния в HTTP ответе (статус ответа), чтобы информировать Клиентов о результате запроса

HTTP определяет 40 стандартных кодов состояния (статусов ответа), которые делятся на пять категорий

Категория

Описание

1xx: Информация	Этот класс содержит информирующие заголовки о процессе передачи. Это обычно предварительный ответ, состоящий только из Status-Line и опциональных заголовков, и завершается пустой строкой. Нет обязательных заголовков. Серверы НЕ ДОЛЖНЫ посылать 1xx ответы HTTP/1.0 клиентам
2xx: Успех	Этот класс кодов состояния указывает, что запрос клиента был успешно получен, понят, и принят
3xx: Перенаправление	Коды этого класса сообщают клиенту, что для успешного выполнения операции необходимо сделать другой запрос, как правило, по другому URI
4xx: Ошибка клиента	Класс кодов 4xx предназначен для указания ошибок со стороны клиента
5xx: Ошибка сервера	Коды ответов, начинающиеся с "5" указывают на случаи, когда сервер знает, что произошла ошибка или он не может обработать запрос

Коды HTTP ответов



2xx

- 200 ОК
- 201 Создано
- 204 Нет контента
- 206 Частичное содержимое

4xx

- 400 Плохой запрос
- 401 Неавторизован
- 403 Доступ запрещен
- 404 Не найдено
- 409 Конфликт
- 429 Слишком много запросов

5xx

- 500 Внутренняя ошибка сервера
- 502 Плохой шлюз
- 503 Сервер не доступен
- 504 Таймаут шлюза

Query params



1. Фильтрация данных

GET /products?category=electronics

GET /products?price_greater_than=100

2. Пагинация

GET /products?page=2&per_page=20

3. Сортировка

GET /products?sort=price_asc

4. Поиск

GET /products?search=query

5. Авторизация и аутентификация

GET /products?token=abc123

Форматы передаваемых данных



```
{  
  "array": [  
    1,  
    2,  
    3  
  ],  
  "boolean": true,  
  "color": "gold",  
  "null": null,  
  "number": 123,  
  "object": {  
    "a": "b",  
    "c": "d"  
  },  
  "string": "Hello World"  
}
```

```
<real:car category="truck">  
  <model lang="en">Scania R 770</model>  
  <year>2005</year>  
  <price currency="US dollar">200000.00</price>  
</real:car>  
<toy:car category="sedan">  
  <title lang="en">Ford Focus</title>  
  <year>2012</year>  
  <price currency="US dollar">100.00</price>  
</toy:car>
```

+|т1

КВИЗ



Какой метод использовать для удаления документа из ресурса?



1 правильный ответ

1. GET

2. POST

3. DELETE

4. PATCH

5. PUT

Какой метод использовать для обновления информации о себе в профиле?



2 правильных ответа

1. GET

2. POST

3. DELETE

4. PATCH

5. PUT

Какой метод использовать для получения корзины на маркетплейсе?



1 правильный ответ

1. GET

2. POST

3. DELETE

4. PATCH

5. PUT

Какой метод использовать для получения корзины на маркетплейсе, где используется более 10 фильтров?



2 правильных ответа

1. GET

2. POST

3. DELETE

4. PATCH

5. PUT

Какой код ответа сервера обозначает успешное завершение запроса



3 правильных ответа

1. 404

2. 200

3. 201

4. 204

5. 206

Какой код ответа сервера указывает на ошибку клиента, связанную с отправленным запросом



1 правильный ответ

1. 200

2. 403

3. 400

4. 404

5. 401

Какой код ответа сервера обычно используется для ресурса, который был успешно удален?



1 правильный ответ

1. 200

2. 203

3. 206

4. 204

5. 401

Какой код ответа сервера обычно используется для клиента, если он не прошел аутентификацию и авторизацию?



1. 401/403

2. 400/403

3. 403/401

4. 401/404

5. 400/404



Создание спецификации в Swagger

Инструмент Swagger



Swagger – это набор инструментов для работы с OpenAPI Specification (OAS). Он помогает разработчикам и системным аналитикам проектировать, документировать и использовать RESTful API сервисы

Swagger включает две основные части:

Swagger Editor - с помощью этого инструмента создается в ручном режиме документация: OpenAPI спецификация

Swagger API-Documentation - результат того, что вы создали в Swagger Editor - API-документация, с помощью которой можно сразу проверять API

Полезные материалы



1. <https://ru.wikipedia.org/wiki/HTTP> и https://en.wikipedia.org/wiki/HTTP#Request_methods - об HTTP
2. https://ru.wikipedia.org/wiki/Список_кодов_состояния_HTTP - коды ответов HTTP
3. <https://jsoneditoronline.org/#right=local.hugewi&left=local.fexifo> - инструмент для валидации JSON
4. <https://www.astera.com/ru/type/blog/types-of-apis/> - типы API
5. <https://developer.mozilla.org/ru/docs/Glossary/Idempotent> - идемпотентность в API
6. <https://habr.com/ru/articles/729528/> - виды API и сравнение
7. <https://habr.com/ru/articles/134033/> - управление версиями