

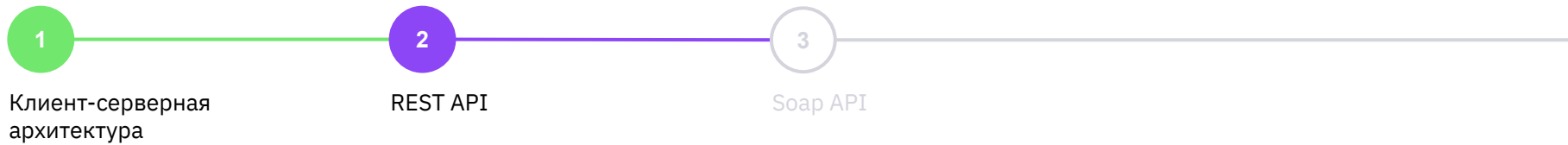
# REST API

Урок 2



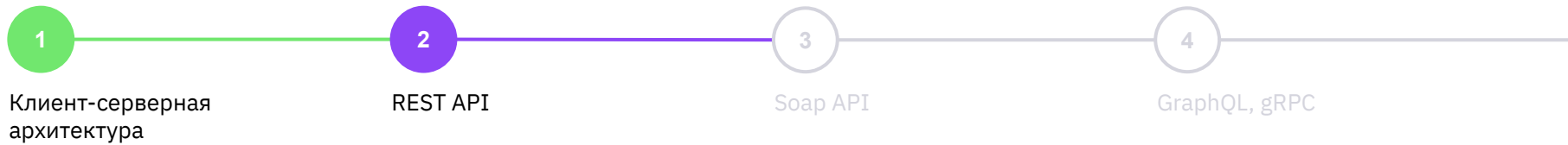


## План курса





## План курса





## План урока

-  Что такое REST API
-  HTTP-методы
-  HTTP-коды
-  Headers
-  Request и Response
-  Кэширование данных
-  Postman
-  Автоматизация запросов в Postman на примере готовых сниппетов





# Что такое REST API



# REST API

**REST API** — это способ взаимодействия сайтов и веб-приложений с сервером. Его также называют RESTful.

 **REST** (Representational State Transfer) — это архитектурный стиль взаимодействия компонентов распределённой системы в компьютерной сети.

 **API** (Application Programming Interface) — это совокупность инструментов и функций в виде интерфейса для создания новых приложений, благодаря которому одна программа будет взаимодействовать с другой.



# HTTP-методы



# HTTP-методы

**METHOD** **URI** **HTTP/VERSION**

где:

- **METHOD** — метод HTTP-запроса,
- **URI** — идентификатор ресурса,
- **VERSION** — версия протокола.





## Пример взаимодействия

### Запрос:

GET /index.php HTTP/1.1

Host: example.com

User-Agent: Mozilla/5.0 (X11; U; Linux i686; ru; rv:1.9b5) Gecko/2008050509 Firefox/3.0b5

Accept: text/html

Connection: close

### Ответ:

HTTP/1.0 200 OK

Server: nginx/0.6.31

Content-Language: ru

Content-Type: text/html; charset=utf-8

Content-Length: 1234

Connection: close

... далее HTML-страница в ответе ...



## Ресурсы и методы

METHOD **URI** HTTP/VERSION

**URI** (Uniform Resource Identifier) — единообразный идентификатор ресурса.

### Пример:

METHOD <https://news.com/new/527648>

**Ресурс** — файл на сервере. Также ресурсом может являться какой-либо абстрактный объект.



## Ресурсы и методы

METHOD:



**GET** — получение ресурса



**POST** — создание ресурса



**PUT** — обновление ресурса








**DELETE** — удаление ресурса



## Ресурсы и методы

Ещё существуют методы:

-  **HEAD** – запрашивает ресурс так же, как и метод GET, но без тела ответа.
-  **CONNECT** – устанавливает «туннель» к серверу, определённого по ресурсу.
-  **OPTIONS** – используется для описания параметров соединения с ресурсом.
-  **TRACE** – выполняет вызов возвращаемого тестового сообщения с ресурса.
-  **PATCH** – используется для частичного изменения ресурса.








# HTTP-коды



## Коды ответов

Код ответа HTTP показывает, был ли выполнен HTTP-запрос.

**Есть 5 классов таких кодов:**

	Информационные	100 - 199
	Успешные	200 - 299
	Перенаправления	300 - 399
	Клиентские ошибки	400 - 499
	Серверные ошибки	500 - 599



## Информационные: 100 - 199

100	Continue	<b>Продолжить</b> — промежуточный ответ, указывает, что запрос успешно принят, и клиент может продолжать присылать запросы либо проигнорировать этот ответ, если запрос был завершён.
101	Switching Protocol	<b>Переключение протокола</b> — ответ на запрос клиента, содержащий заголовок Upgrade. Указывает, что сервер переключился на протокол, который был указан в заголовке. Эта возможность позволяет перейти на несовместимую версию протокола и обычно не используется.
102	Processing	<b>В обработке</b> — сервер получил запрос и обрабатывает его, но обработка ещё не завершена.
103	Early Hints	<b>Ранние подсказки</b> — в ответе сообщаются ресурсы, которые могут быть загружены заранее, пока сервер будет подготавливать основной ответ.



## Успешные: 200 - 299

200	OK	<p><b>Успешно</b> — запрос успешно обработан. «Успешность» зависит от запрошенного метода HTTP:</p> <ul style="list-style-type: none"><li>• GET (получить) — запрошенный ресурс найден и передан в теле ответа.</li><li>• HEAD (заголовок) — заголовки переданы в ответе.</li><li>• POST (посылка) — ресурс, описывающий результат действия сервера на запрос, передан в теле ответа.</li><li>• TRACE (отслеживать) — тело ответа содержит тело запроса, полученного сервером.</li></ul>
201	Created	<b>Создано</b> — запрос успешно выполнен, в результате был создан ресурс.
202	Accepted	<b>Принято</b> — запрос принят, но ещё не обработан.
203	Non-Authoritative Information	<b>Информация не авторитетна.</b>
204	No Content	<b>Нет содержимого</b> для ответа на запрос.
205	Reset Content	<b>Сбросить содержимое</b> — код присылается, когда запрос обработан.





## Перенаправления: 300 - 399

300	Multiple Choice	<b>Множественный выбор</b> — присылается, когда у запроса более одного из возможных ответов.
301	Moved Permanently	<b>Перемещён на постоянной основе</b> — URI запрашиваемого ресурса был изменён.
303	See Other	<b>Просмотр других ресурсов</b> — код присылается, чтобы направлять клиента для получения запрашиваемого ресурса в другой URI с запросом GET.
305	Use Proxy	<b>Использовать прокси</b> — запрошенный ресурс должен быть доступен через прокси.



## Клиентские ошибки: 400 - 499

400	Bad Request	Плохой запрос — сервер не понимает запрос из-за неверного синтаксиса.
401	Unauthorized	Неавторизованно — для получения ответа нужна аутентификация.
403	Forbidden	Запрещено — у клиента нет права доступа к содержимому, поэтому сервер отказывается дать ответ.
404	Not Found	Не найден — сервер не может найти запрашиваемый ресурс. Этот код — самый известный из-за частоты его появления в вебе.



## Серверные ошибки: 500 - 599

500	Internal Server Error	<b>Внутренняя ошибка сервера</b> — сервер не знает, как обработать ситуацию, с которой он столкнулся.
501	Not Implemented	<b>Не выполнено</b> — метод запроса не поддерживается сервером и не может быть обработан.
502	Bad Gateway	<b>Плохой шлюз</b> — сервер, во время работы в качестве шлюза для получения ответа, нужного для обработки запроса, получил недействительный (недопустимый) ответ.
503	Service Unavailable	<b>Сервис недоступен</b> — сервер не готов обрабатывать запрос. Частые причины: отключение сервера или его перегруженность.



# Headers



# Headers

Заголовки HTTP позволяют клиенту и серверу отправлять дополнительную информацию с HTTP-запросом или ответом.





## ▼ Request Headers

```
GET /api/v1/products HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Cache-Control: max-age=0
Connection: keep-alive
Cookie: _ga=GA1.2.1016557436.1604261537; REMEMBERME=U3ltZm9ueVxDb21wb25lbnRcU2VjdXJpdHlcQ29yZVxVc2VyXFVzZXI6Y3k1NVlXdHZkbXhsZGtCamIzSndMbTfoYVd3dWVuVT06MTY10Tc4MjkzMjpmZWUxMWNhNGY.
2ZKjYjVhYTMwZTk1NmJmMGRmZjQ5; PHPSESSID=d6ofr2kqvvd5qudfp59tdjvd4v; pAuth=K6qwjCEVNmFcaDssl1J85Z4u7Thkp7UkF2pNa6jndhzkeCog7fQ8KkGV2PoSwnE
Host: api.youla.io
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="102", "Google Chrome";v="102"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "macOS"
```



# Headers




Заголовки могут группироваться по контекстам:

-  **Основные заголовки** применяются как к запросам, так и к ответам, но не имеют отношения к данным, передаваемым в теле.
-  **Заголовки запроса** содержат больше информации о ресурсе, который нужно получить, или о клиенте, который запрашивает ресурс.
-  **Заголовки ответа** (en-US) содержат дополнительную информацию об ответе (например, о местонахождении) или о сервере, предоставившем его.
-  **Заголовки сущности** содержат информацию о теле ресурса (например, длину содержимого или тип MIME).



# Headers

## Группы заголовков запроса:

-  основные заголовки (General headers),
-  заголовки запроса (Request headers),
-  заголовки сущности.



# Request и Response





## Request

GET /api/v1/products HTTP/1.1

Accept: text/html

Accept-Encoding: gzip, deflate, br

Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7

Cache-Control: max-age=0

Connection: keep-alive

Host: api.youla.io

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/102.0.5005.61 Safari/537.36



## Response

HTTP/1.1 200 OK

Server: nginx/1.16.1

Content-Type: application/json

Content-Language: ru-RU

Strict-Transport-Security: max-age=31536000; includeSubDomains; preload

Access-Control-Allow-Origin: https://youla.io

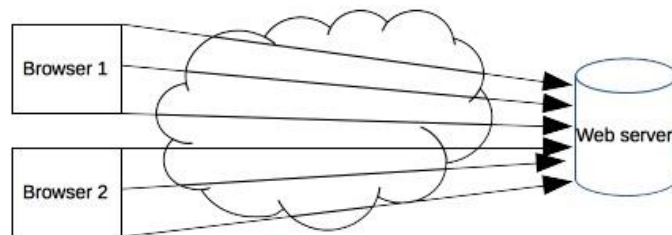
```
{
  "data": [],
  "status": 200,
  "detail": "Products list",
  "uri": "/products",
  "meta": {
    "search_id": "06aa4065c2ecc5fae6fff1540a00bcbd",
  }
}
```



# Кэширование данных

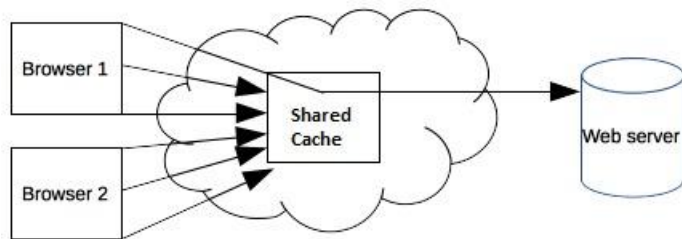
## Кэширование данных

No cache



All identical requests are going through to the server.

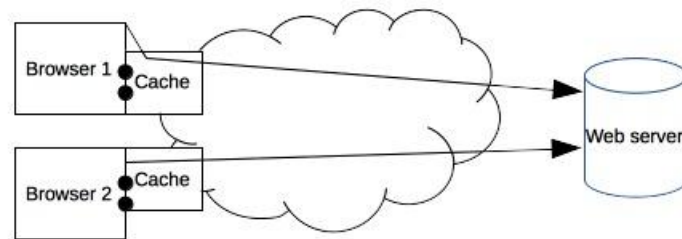
Shared cache



The first request is going through.

Subsequent identical requests are served by the shared cache.  
(more efficient)

Local (private) cache








The first request of each client is going through.

Subsequent identical requests are not even sent, but served by the local cache.  
(most efficient, except for first requests)



## Цели кэширования

-  **Успешно загруженные ресурсы:** ответ 200 OK на запрос методом GET HTML-документов, изображений или файлов.
-  **Постоянные перенаправления:** ответ 301 Moved Permanently (перемещено навсегда).
-  **Сообщения об ошибках:** ответ 404 Not Found (не найдено).
-  **Неполные результаты:** ответ 206 Partial Content (частичное содержимое).
-  **Ответы на запросы, отличные от GET,** если есть что-либо, подходящее для использования в качестве ключа кэша.

## stale / fresh





# Postman



## Postman

**Postman** — инструмент с открытым исходным кодом. Используется для тестирования REST API.







# Postman

## Auth - CreateToken

1.0.0 ▾

Creates a new auth token to use for access to the PUT and DELETE /booking

POST

`https://restful-booker.herokuapp.com/auth`

Example 1:

```
curl -X POST \  
  https://restful-booker.herokuapp.com/auth \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "username" : "admin",  
    "password" : "password123"  
  }'
```

Header

[Restful-booker](#)

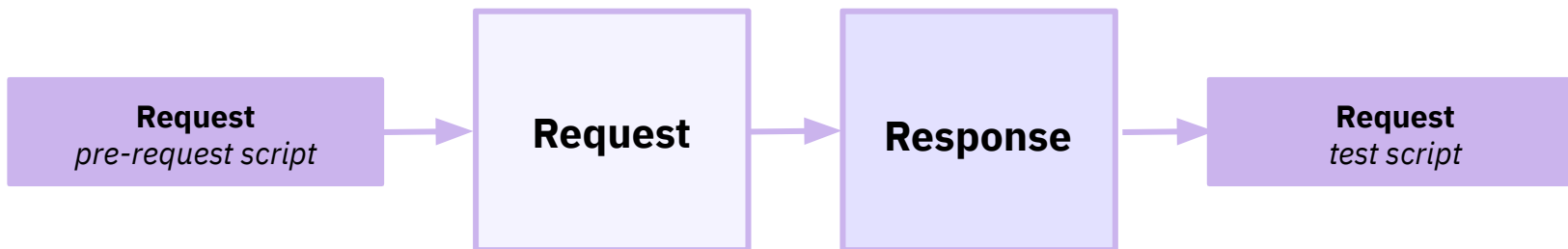


# Автоматизация запросов в Postman на примере готовых сниппетов



## Pre-request script

**Pre-request script** — это логика или часть кода, выполнение которой гарантировано до начала выполнения запроса.





## Tests

Раздел нужен для написания тестов, которые будут выполняться после получения ответа на запрос.

POST

▼

https://restful-booker.herokuapp.com/auth

Params

Authorization

Headers (10)

Body ●

Pre-request Script

Tests ●

Settings

```
1 var jsonData = pm.response.json();
2 pm.collectionVariables.set("token", jsonData.token);
3
4
5
```



Спасибо 😊  
за внимание