ozon{tech

HTTP, REST и GRPC

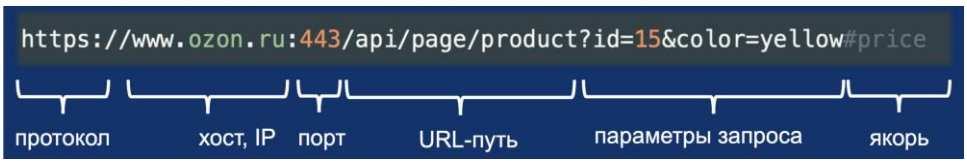
План занятия

HTTP

HyperText Transfer Protocol — протокол прикладного уровня передачи данных.

Сейчас HTTP используется для передачи произвольных данных.

URL (Uniform Resource Locator) определяет местонахождение ресурса, передает параметры в GET-запросы.



HTTP-методы

HTTP-метод указывает серверу на то, какое действие мы хотим произвести с ресурсом.

GET запрашивает представление ресурса. Запросы с использованием этого метода могут только извлекать данные.

HEAD запрашивает ресурс так же, как и метод GET, но без тела ответа.

POST используется для отправки сущностей к определенному ресурсу. Часто вызывает изменение состояния или какие-то побочные эффекты на сервере (не всегда).

PUT заменяет все текущие представления ресурса данными запроса.

DELETE удаляет указанный ресурс.

CONNECT устанавливает туннель к серверу, определенному по ресурсу.

OPTIONS используется для описания параметров соединения с ресурсом.

TRACE выполняет вызов возвращаемого тестового сообщения с ресурса.

PATCH используется для частичного изменения ресурса.

Заголовки

Заголовки позволяют клиенту и серверу отправлять дополнительную информацию с HTTP-запросом или ответом. В HTTP-заголовке содержится нечувствительное к регистру название, а затем после (:) непосредственно значение. Пробелы перед значением игнорируются.

```
Content-Type: application/json
Pragma: no-cache
Accept: application/json
Accept-Encoding: gzip, deflate, br
Cache-Control: no-cache
Accept-Language: en-us
Host: www.ozon.ru
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko)
Connection: keep-alive
x-o3-app-name: dweb_client
x-o3-app-version: release_22-8'-'2021_bec6de4a
```

```
Content-Type: application/json; charset=UTF-8
Server-Timing:
FirstByte;dur=26.496786,Widgets;dur=19.850209,Resolve;dur=4.602773,Total;dur=26.498882,Internal;dur=2.0459
Content-Encoding: br
Date: Wed, 22 Sep 2021 13:23:44 GMT
X-Content-Type-Options: nosniff
Vary: Accept-Encoding, Origin
X-Frame-Options: SAMEORIGIN
x-o3-trace-id: 23061db461746b02
Server: nginx
x-b3-traceid: 23061db461746b02
x-o3-page-type: search_suggestions
x-o3-platform: desktop
x-recruiting: Like web development? Write us: https://job.ozon.ru/
```

Тело

Основные данные запроса или ответа. Могут быть в любом формате, для REST API чаще всего используется JSON.

```
{
    "layout": [
        {
            "component": "searchHistory",
            "params": "{}",
            "stateId": "searchHistory-372553-default-1",
            "version": 3,
            "vertical": "catalog",
            "widgetTrackingInfo": "
{\"name\":\"catalog.searchHistory\",\"vertical\":\"catalog\",\"component\":\"searchHistory\",\"version\":3
,\"id\":372553,\"revisionId\":382053,\"index\":1,\"timeSpent\":18192213}",
            "timeSpent": 18192213,
            "name": "catalog.searchHistory"
        }
    ]
}
```

Код ответа

Показывает, был ли запрос успешно выполнен.

```
1xx: Informational (информационные):
    100 Continue («продолжай»);
    101 Switching Protocols («переключение протоколов»);
    102 Processing («идёт обработка»);
    103 Early Hints («ранняя метаинформация»);
```

```
2xx: Success (успешно):
    200 OK («хорошо»);
    201 Created («создано»);
    202 Accepted («принято»);
    203 Non-Authoritative Information («информация не авторитетна»);
    204 No Content («нет содержимого»);
    205 Reset Content («сбросить содержимое»);
    206 Partial Content («частичное содержимое»);
    207 Multi-Status («многостатусный»);
    208 Already Reported («уже сообщалось»);
    226 IM Used («использовано IM»)
```

```
3xx: Redirection (перенаправление):
    300 Multiple Choices («множество выборов»);
    301 Moved Permanently («перемещено навсегда»);
    302 Moved Temporarily («перемещено временно»),
    302 Found («найдено»);
    303 See Other («смотреть другое»);
    304 Not Modified («не изменялось»);
    305 Use Proxy («использовать прокси»);
    306 — зарезервировано (код использовался только в ранних спецификациях);
    307 Temporary Redirect («временное перенаправление»);
    308 Permanent Redirect («постоянное перенаправление»)
```

```
4xx: Client Error (ошибка клиента):
    400 Bad Request («неправильный, некорректный запрос»);
    401 Unauthorized («не авторизован (не представился)»);
    402 Payment Required («необходима оплата»);
    403 Forbidden («запрещено (не уполномочен)»);
    404 Not Found («не найдено»);
    405 Method Not Allowed («метод не поддерживается»);
    406 Not Acceptable («неприемлемо»);
    407 Proxy Authentication Required («необходима аутентификация прокси»);
    408 Request Timeout («истекло время ожидания»);
    409 Conflict («конфликт»);
    410 Gone («удалён»);
    411 Length Required («необходима длина»);
    412 Precondition Failed («условие ложно»);
    413 Payload Too Large («полезная нагрузка слишком велика»);
    414 URI Too Long («URI слишком длинный»);
    415 Unsupported Media Type («неподдерживаемый тип данных»);
    416 Range Not Satisfiable («диапазон не достижим»);
    417 Expectation Failed («ожидание не удалось»);
    418 I'm a teapot («я — чайник»);
    419 Authentication Timeout (not in RFC 2616) («обычно ошибка проверки CSRF»);
    421 Misdirected Request ;
    422 Unprocessable Entity («необрабатываемый экземпляр»);
    423 Locked («заблокировано»);
    424 Failed Dependency («невыполненная зависимость»);
    425 Too Early («слишком рано»);
    426 Upgrade Required («необходимо обновление»);
    428 Precondition Required («необходимо предусловие»);
    429 Too Many Requests («слишком много запросов»);
    431 Request Header Fields Too Large («поля заголовка запроса слишком большие»);
    449 Retry With («повторить с»);
    451 Unavailable For Legal Reasons («недоступно по юридическим причинам»).
    499 Client Closed Request (клиент закрыл соединение);
```

```
5xx: Server Error (ошибка сервера):
    500 Internal Server Error («внутренняя ошибка сервера»);
    501 Not Implemented («не реализовано»);
    502 Bad Gateway («плохой, ошибочный шлюз»);
    503 Service Unavailable («сервис недоступен»);
    504 Gateway Timeout («шлюз не отвечает»);
    505 HTTP Version Not Supported («версия HTTP не поддерживается»);
    506 Variant Also Negotiates («вариант тоже проводит согласование»);
    507 Insufficient Storage («переполнение хранилища»);
    508 Loop Detected («обнаружено бесконечное перенаправление»);
    509 Bandwidth Limit Exceeded («исчерпана пропускная ширина канала»);
    510 Not Extended («не расширено»);
    511 Network Authentication Required («требуется сетевая аутентификация»);
    520 Unknown Error («неизвестная ошибка»);
    521 Web Server Is Down («веб-сервер не работает»);
    522 Connection Timed Out («соединение не отвечает»);
    523 Origin Is Unreachable («источник недоступен»);
    524 A Timeout Occurred («время ожидания истекло»);
    525 SSL Handshake Failed («квитирование SSL не удалось»);
    526 Invalid SSL Certificate («недействительный сертификат SSL»).
```

Куки

Небольшой фрагмент данных, который сервер отправляет браузеру пользователя. Браузер может сохранить этот фрагмент у себя и отправлять на сервер с каждым последующим запросом. Это, в частности, позволяет узнать, с одного ли браузера пришли несколько запросов (например, для аутентификации пользователя). С помощью кук можно сохранить любую информацию о состоянии, HTTP-протокол сам по себе этого делать не умеет.

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: yummy_cookie=choco
Set-Cookie: tasty_cookie=strawberry
```

```
GET /sample_page.html HTTP/1.1
Host: www.example.org
Cookie: yummy_cookie=choco;
tasty_cookie=strawberry
```

REST API

Representational State Transfer (REST) — передача состояния представления.

Технология позволяет получать и модифицировать данные и состояния удаленных приложений, передавая HTTP-вызовы через интернет или любую другую сеть.

Application Programming Interface (API), или программный интерфейс приложения, — набор инструментов, который позволяет одним программам работать с другими.

CRUD

Четыре базовые функции, используемые при работе с БД: создание (create), чтение (read), модификация (update), удаление (delete).

В API,соответственно, HTTP-методы — POST, GET, PUT, DELETE.

Более подробно об HTTP: https://developer.mozilla.org/ru/docs/Web/HTTP

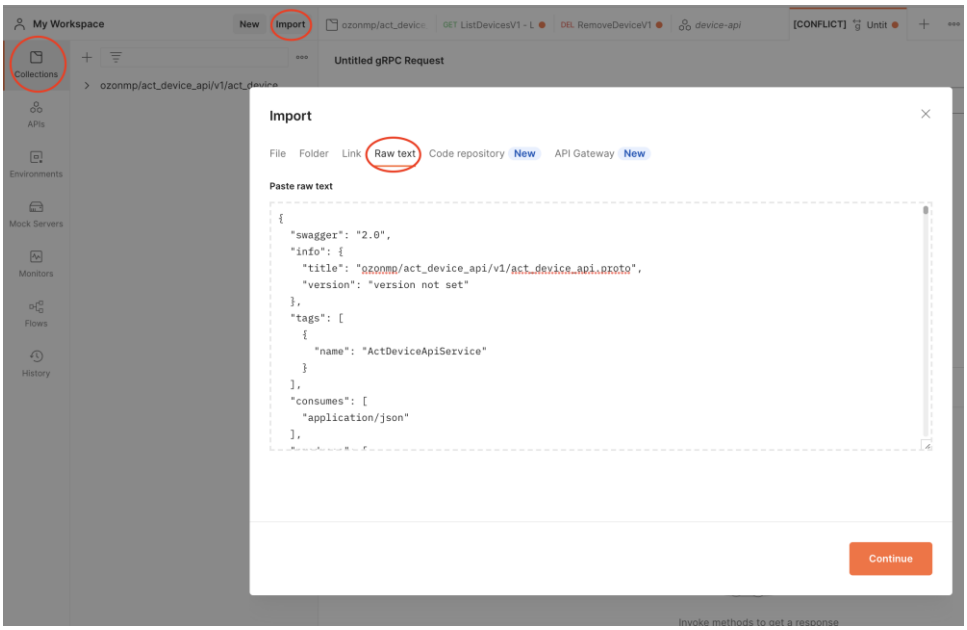
Инструменты для работы REST API

SWAGGER
Интерактивное описание REST API, помогает понять структуру API без доступа к коду: https://swagger.io/

POSTMAN

Платформа для работы с API. Упрощает каждый этап жизненного цикла API, оптимизирует тестирование и совместную работу: https://www.postman.com/

Импорт коллекции (Raw text — swagger.json):



CURL

Консольная утилита для взаимодействия с сервером по различным протоколам (в том числе HTTP): https://curl.se/

GRPC

Система удаленного вызова процедур (RPC), разработанная в Google. В качестве транспорта используется HTTP/2, в качестве языка описания интерфейса — Protocol Buffers: https://grpc.io/

Protocol Buffers

Язык описания интерфейса, описывает структуру вызовов (файлы с расширением **.proto**).

Stub

Локальный объект на стороне клиента (по сути клиент), содержащий в себе все методы сервиса.

Unary RPC

Самый простой тип удаленного вызова процедуры: один запрос — один ответ.

Server streaming RPC

Сервер отправляет стрим (поток, множество) ответов на один запрос клиента, данные о выполненном запросе приходят после полной отправки всех данных стрима.

Client streaming RPC

Клиент отправляет стрим запросов серверу и получает один ответ.

Bidirectional streaming RPC

Двунаправленный стриминг между клиентом и сервисом.

Deadlines/Timeouts

Клиент может установить время, которое он готов потратить на получение ответа.

RPC termination

Клиент и сервер могут оценивать успешность вызова со своей стороны по-разному.

Cancelling an RPC

Вызов может быть закончен в любое время по инициативе как сервиса, так и клиента.

Metadata

Данные об определенном вызове (можно сравнить с HEADERS в HTTP).

Channels

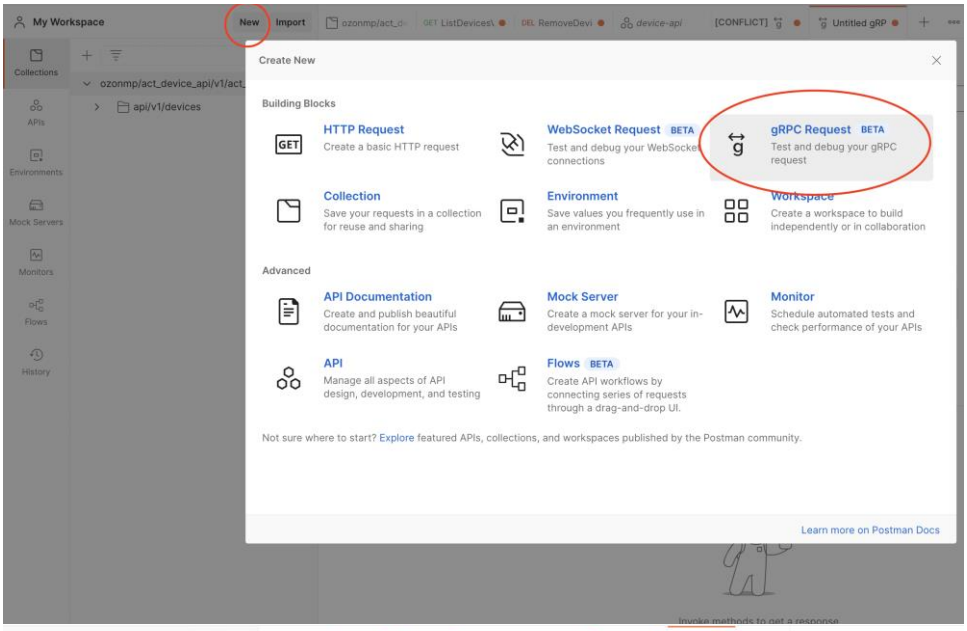
Каналы осуществляют подключение к серверу во время инициализации стаба.

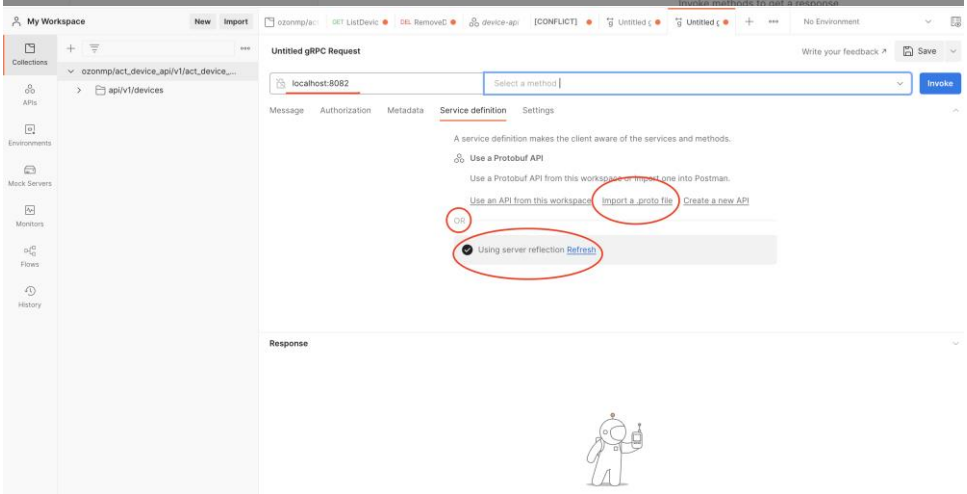
Инструменты для работы GRPC

POSTMAN

Платформа для работы с API. Упрощает каждый этап жизненного цикла API, оптимизирует тестирование и совместную работу: https://www.postman.com/

Создание коллекции (импорт протофайла или использование рефлексии):





BLOOMRPC

Легкое приложения для работы только с GRPC: https://github.com/bloomrpc/bloomrpc

grpc_cli

Стандартная утилита из репозитория gRPC: https://github.com/grpc/grpc/blob/master/doc/command_line_tool.md

grpcurl

Сторонняя утилита для взаимодействия с gRPC-сервером: https://github.com/fullstorydev/grpcurl

grpc_cli, grpcurl cheatsheet https://gitlab.ozon.dev/-/snippets/9

Evans

Консольная утилита с широким функционалом (автокомплит, скрипты): https://github.com/ktr0731/evans