46.

Основы

работы с API

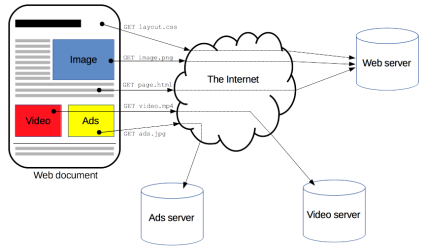
Цель:

Познакомиться со следующими основами работы с API:

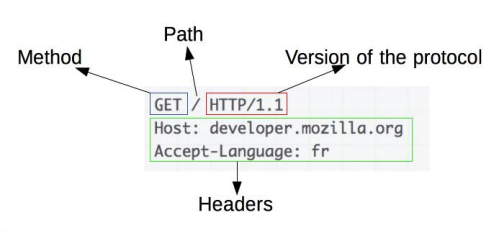
- обзор протокола HTTP

- HTTP запросы

HTTP:



HTTP запросы:



HTTP запросы:

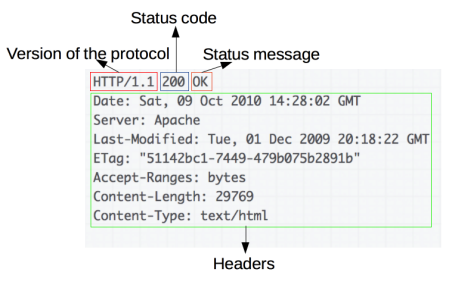
Запросы содержат следующие элементы:

● HTTP-метод

● Путь к ресурсу: (например developer.mozilla.org) ● Версию HTTP-протокола.

● Заголовки (опционально), предоставляющие дополнительную информацию для сервера. ● Тело, для некоторых методов, таких как POST, которое содержит отправленные данные.

HTTP ответы:



HTTP ответы:

Ответы содержат следующие элементы:

● Версию HTTP-протокола.

● HTTP код состояния, сообщающий об успешности запроса или причине неудачи.

● Сообщение состояния — краткое описание кода состояния. ● HTTP заголовки, подобно заголовкам в запросах. ● Опционально: тело, содержащее пересылаемый ресурс.

HTTP методы:

● **GET:** получить доступ к существующему ресурсу. В URL перечислена вся необходимая информация, чтобы сервер смог найти и вернуть в качестве ответа искомый ресурс.

● **POST:** используется для создания нового ресурса. POST запрос обычно содержит в себе всю нужную информацию для создания нового ресурса.

HTTP методы:

● **PUT:** обновить текущий ресурс. PUT запрос содержит обновляемые данные.

● **DELETE:** служит для удаления существующего ресурса.

● **HEAD:** аналогичен GET. Разница в том, что при данном виде запроса не передаётся сообщение. Сервер получает только заголовки. Используется, к примеру, для того чтобы определить, был ли изменен ресурс.

HTTP методы:

● **TRACE:** во время передачи запрос проходит через множество точек доступа и прокси серверов, каждый из которых вносит свою информацию: IP, DNS. С помощью данного метода, можно увидеть всю промежуточную информацию.

● **OPTIONS:** используется для определения возможностей сервера, его параметров и конфигурации для конкретного ресурса.

HTTP headers:

● **Accept-Language:** en-us,en;q=0.5

● **User-Agent:** Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729) Этот заголовок может содержать несколько частей информации, таких как: ○ Имя и версия браузера.

○ Название и версия операционной системы.

○ Язык по умолчанию.

● **Cookie:** PHPSESSID=r2t5690jko5r4q7ib3vtdjq120; foo=bar ● **Authorization:** Basic bXl8jk00yOm15cGFzcw==

Коды ответа HTTP:

1. Информационные **100 - 199**

2. Успешные **200 - 299**

3. Перенаправления **300 - 399**

Своеобразное сообщение клиенту о необходимости совершить ещё одно действие. Самый распространённый вариант применения: перенаправить клиент на другой адрес.

4. Клиентские ошибки **400 - 499**

Данный класс сообщений используется сервером, если он решил, что запрос был отправлен с ошибкой.

5. Серверные ошибки **500 - 599**

Коды ответа HTTP. Примеры:

● **200 OK**

● **204 No Content:** в теле ответа нет сообщения.

● **400 Bad Request**: вопрос был сформирован неверно.

● **401 Unauthorized**: для совершения запроса нужна аутентификация. Информация передается через заголовок Authorization.

● **403 Forbidden**: сервер не открыл доступ к ресурсу.

● **404 Not Found:** означает, что ресурс не найден на сервере. ● **503 Service Unavailable**: это может случиться, если на сервере произошла ошибка или он перегружен. Обычно в этом случае, сервер не отвечает, а время, данное на ответ, истекает.

HTTP/HTTPS:

● HTTPS не является отдельным протоколом передачи данных, а представляет собой расширение протокола HTTP с надстройкой шифрования;

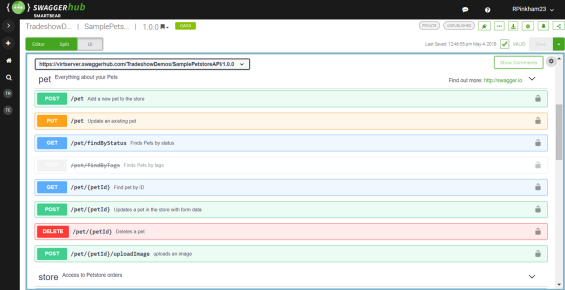
● передаваемые по протоколу HTTP данные не защищены, HTTPS обеспечивает конфиденциальность информации путем ее шифрования;

● HTTP использует порт 80, HTTPS — порт 443.

Postman & Swagger:



Swagger:



Postman: