

Skillbox

Проектирование API

HTTP-протокол

Модели OSI и TCP/IP

Чернухин Максим

Software Architect

АО «Альфа-Банк»

В прошлом модуле

- Научились управлять транзакциями
- Узнали, что такое ACID и BASE
- Поняли, что такое сага

В этом модуле

- Узнаем, как спроектировать API
- Что такое REST
- Как помогает в разработке API-first

В этом уроке мы узнаем

- Что такое HTTP
- Как используют HTTP
- Будущее протокола
- Модели OSI и TCP/IP

HTTP

Немного истории

- HTTP — HyperText Transfer Protocol

HTTP

Немного истории

- HTTP — HyperText Transfer Protocol
- HTTP 0.9 был предложен в марте 1991 года Тимом Бернерсом-Ли

HTTP

Немного истории

- HTTP — HyperText Transfer Protocol
- HTTP 0.9 был предложен в марте 1991 года Тимом Бернерсом-Ли
- Предназначался для клиент-серверного взаимодействия

HTTP

Немного истории

- HTTP — HyperText Transfer Protocol
- HTTP 0.9 был предложен в марте 1991 года Тимом Бернерсом-Ли
- Предназначался для клиент-серверного взаимодействия
- Занимает прикладной уровень в моделях OSI и TCP/IP

HTTP

Немного истории

- HTTP — HyperText Transfer Protocol
- HTTP 0.9 был предложен в марте 1991 года Тимом Бернерсом-Ли
- Предназначался для клиент-серверного взаимодействия
- Занимает прикладной уровень в моделях OSI и TCP/IP
- HTTP 1.0 описан в 1996 году в документе RFC 1945

HTTP

Немного истории

- HTTP — HyperText Transfer Protocol
- HTTP 0.9 был предложен в марте 1991 года Тимом Бернерсом-Ли
- Предназначался для клиент-серверного взаимодействия
- Занимает прикладной уровень в моделях OSI и TCP/IP
- HTTP 1.0 описан в 1996 году в документе RFC 1945
- HTTP 1.1 — июнь 1999 года, RFC 2068

HTTP

Немного истории

- HTTP — HyperText Transfer Protocol
- HTTP 0.9 был предложен в марте 1991 года Тимом Бернерсом-Ли
- Предназначался для клиент-серверного взаимодействия
- Занимает прикладной уровень в моделях OSI и TCP/IP
- HTTP 1.0 описан в 1996 году в документе RFC 1945
- HTTP 1.1 — июнь 1999 года, RFC 2068
- HTTP 2 — февраль 2015 года, RFC 7540

HTTP

Немного истории

- HTTP — HyperText Transfer Protocol
- HTTP 0.9 был предложен в марте 1991 года Тимом Бернерсом-Ли
- Предназначался для клиент-серверного взаимодействия
- Занимает прикладной уровень в моделях OSI и TCP/IP
- HTTP 1.0 описан в 1996 году в документе RFC 1945
- HTTP 1.1 — июнь 1999 года, RFC 2068
- HTTP 2 — февраль 2015 года, RFC 7540
- HTTP 3 — Draft 2019 года

Модели OSI и TCP/IP

OSI

Application layer

Presentation layer

Session layer

Transport layer

Network layer

Data link layer

Physical layer

TCP/IP

Application layer

Transport layer

Network layer

Network Access layer

Модели OSI и TCP/IP

OSI

- Передача битов по сети
- Представление сигналов в цифровую информацию

Data link layer

Physical layer

TCP/IP

Network Access layer

Модели OSI и TCP/IP

OSI

- Выделение потоков бит
- Обнаружение и коррекция ошибок
- Управление доступом к возможности отправлять сообщения в широковещательных сетях

Data link layer

Physical layer

TCP/IP

Network Access layer

Модели OSI и TCP/IP

OSI

- Адресация запросов
- Маршрутизация через транспортные узлы

Network layer

Data link layer

Physical layer

TCP/IP

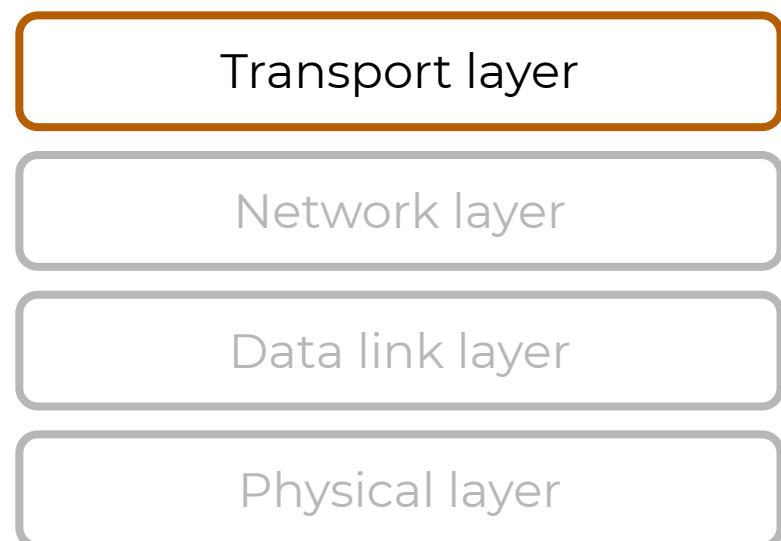
Network layer

Network Access layer

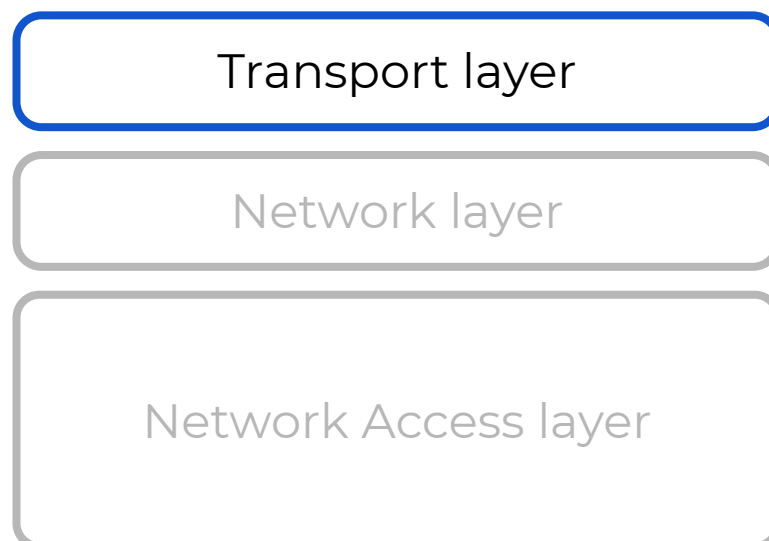
Модели OSI и TCP/IP

OSI

- Надёжность выше, чем у сети
- Позволяет абстрагироваться от нижних уровней

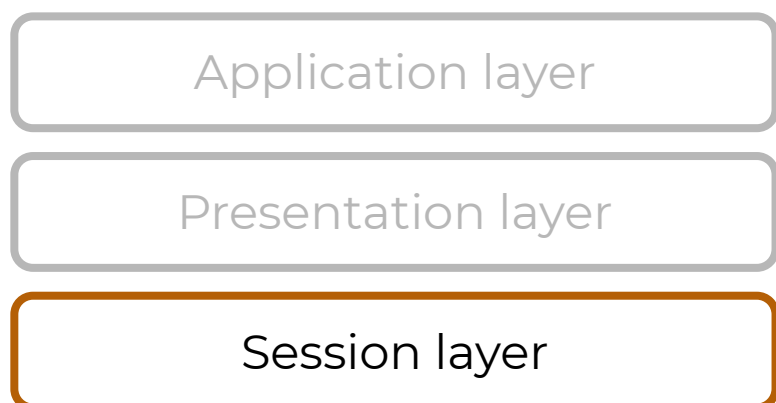


TCP/IP



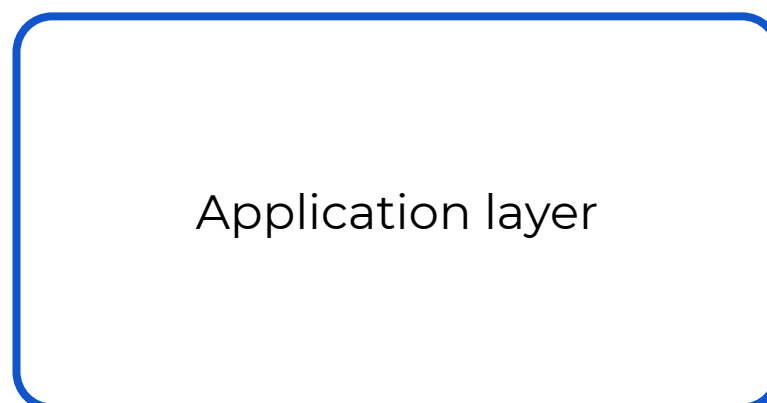
Модели OSI и TCP/IP

OSI



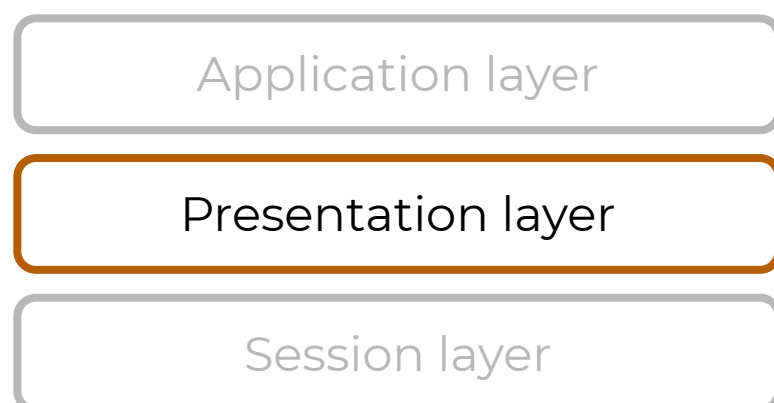
- Для поддержания сессии
- Определение последовательности доступа
- Определение доступа к ресурсам
- Защита от разрывов соединения
- Возобновление работы после разрывов

TCP/IP



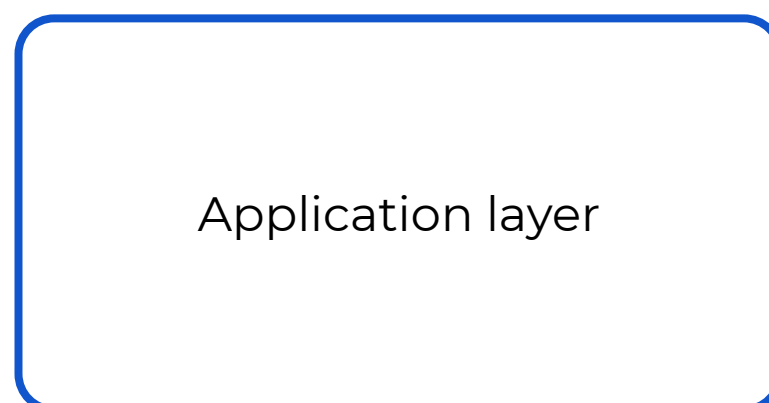
Модели OSI и TCP/IP

OSI



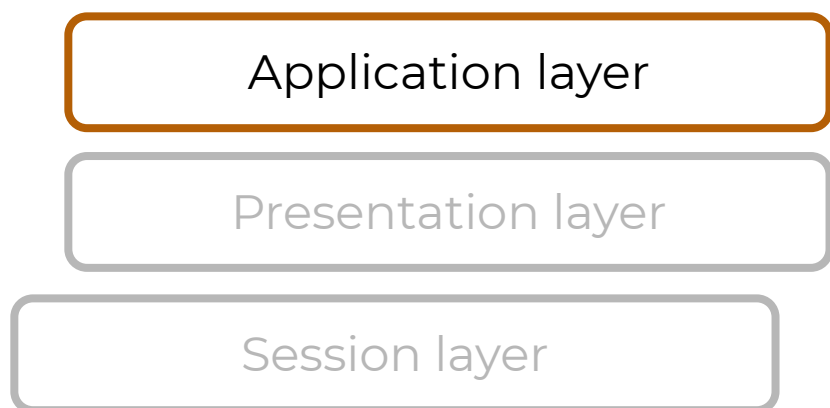
- Форматы представления символов
- Шифрование и дешифрование

TCP/IP



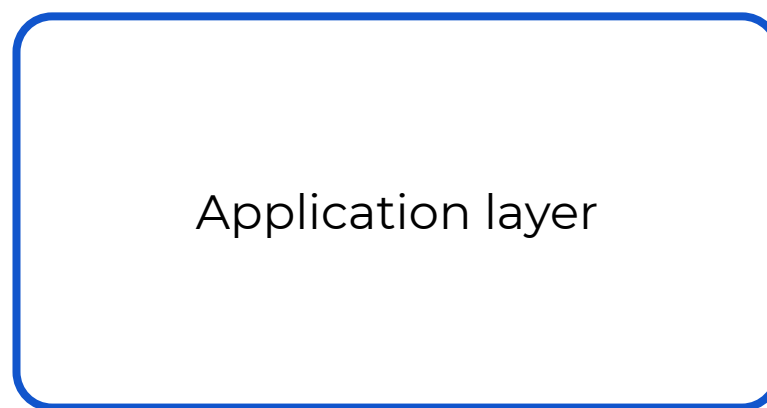
Модели OSI и TCP/IP

OSI



- Электронная почта
- Видео, аудио
- Доступ к файлам
- WEB и так далее

TCP/IP



HTTP-запрос

```
GET / HTTP/1.1  
Host: developer.mozilla.org  
Accept-Language: fr
```

ТСР/ІР для НТТР

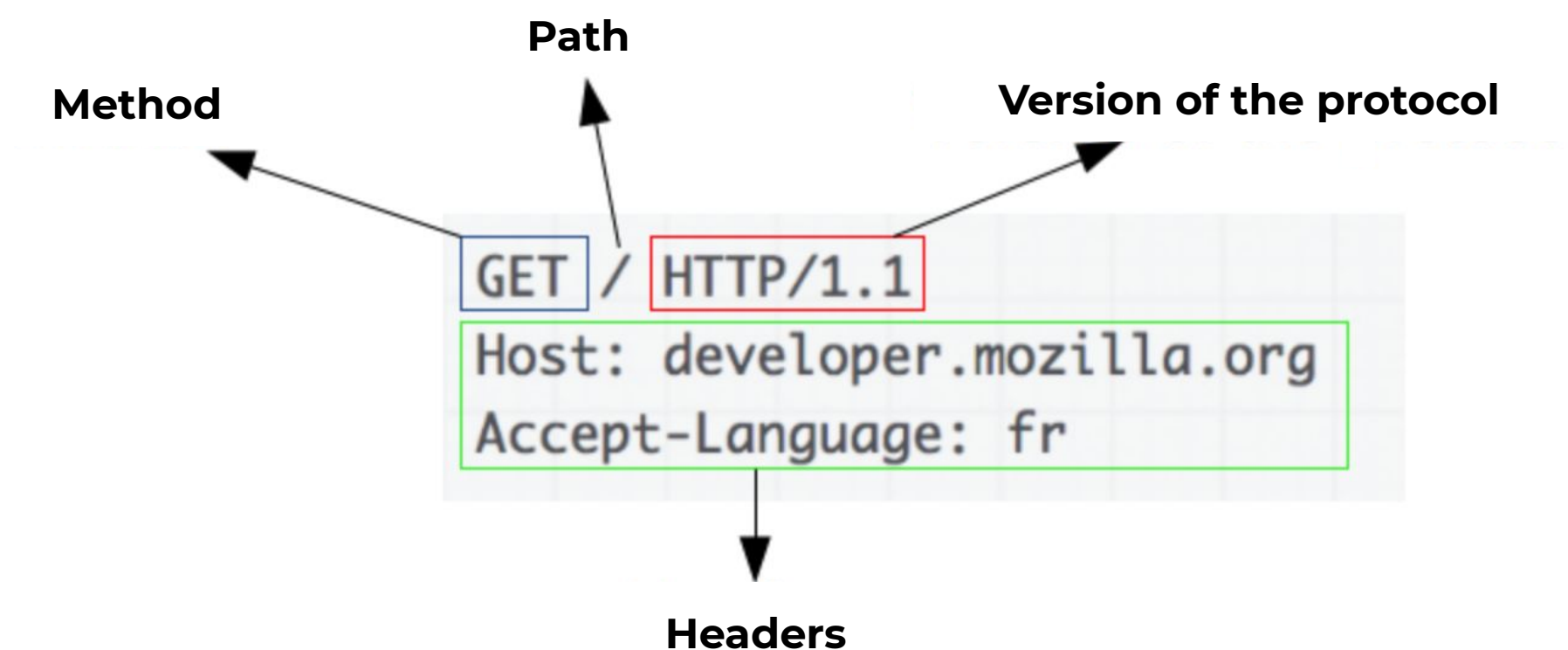
НТТР

ТСР/UDP

ІР v4/v6

Network Access layer

HTTP-запрос

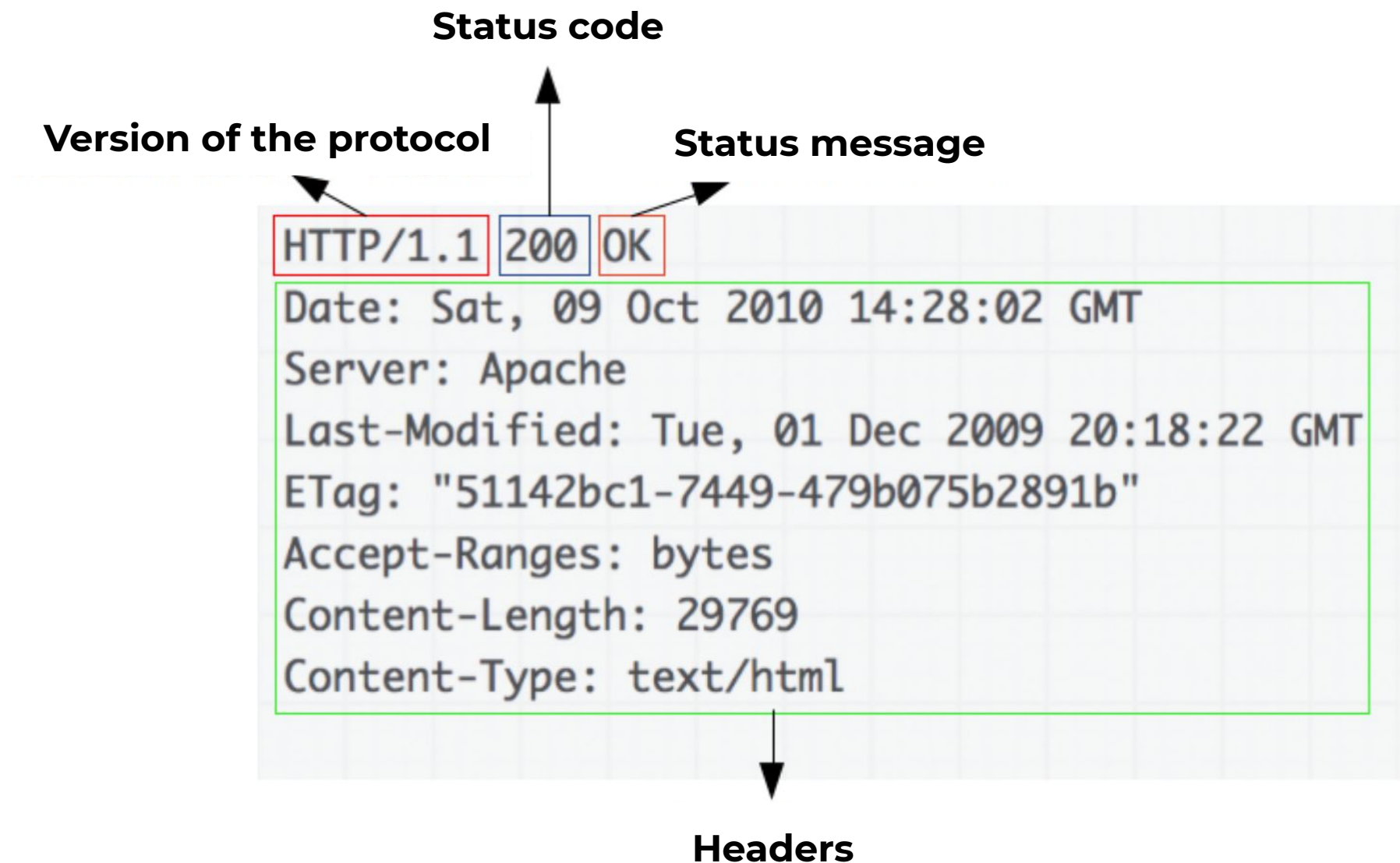


HTTP-OTBET

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html

<!DOCTYPE html... (here comes the 29769 bytes o
f the requested web page)
```

HTTP-OTBET



HTTP-методы

- Метод **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 определённому ресурсу. Часто вызывает изменение состояния или какие-то побочные эффекты на сервере
- **PUT** заменяет все текущие представления ресурса данными запроса
- **DELETE** удаляет указанный ресурс

HTTP-методы

- **CONNECT** устанавливает «туннель» к серверу, определённому 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 cookie



Set-Cookie: <имя-cookie>=<заголовок-cookie>

HTTP-коды ответов

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

Документация RFC 7231

HTTP-КОДЫ ОТВЕТОВ

418 — I'm a teapot

<https://www.google.com/teapot>



418. I'm a teapot.

The requested entity body is short and stout.
Tip me over and pour me out.



Итоги

- Мы разобрали протокол HTTP
- Поняли, как он встраивается в модели OSI и TCP/IP
- Разобрали содержание запросов и ответов
- Увидели, какие бывают коды ответов
- Узнали, что такое cookie

Не разобрали

- Авторизацию

Что дальше?

- Узнаем, что такое REST
- Поймём, почему REST ассоциируется с HTTP
- Узнаем, как строить RESTful-сервисы

Skillbox

**Спасибо
за внимание!**