



# Lesson - 9

**Client-server architecture. HTTP protocol.**

SkillUp, by Vitali Cernomschi

# План занятия



1. Вопросы по домашнему заданию.
2. HTTP: [http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/http\\_basics.html](http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/http_basics.html)
  - a. Клиент
  - b. Сервер
  - c. Архитектура клиент-сервер
  - d. HTTP methods
3. JSON Server:
  - a. <https://www.npmjs.com/package/json-server>
4. Programs to create and test custom HTTP requests, view responses:
  - a. [Postman](#)
  - b. [Advanced Rest Client](#)
  - c. [Curl from CLI](#)

# Клиент

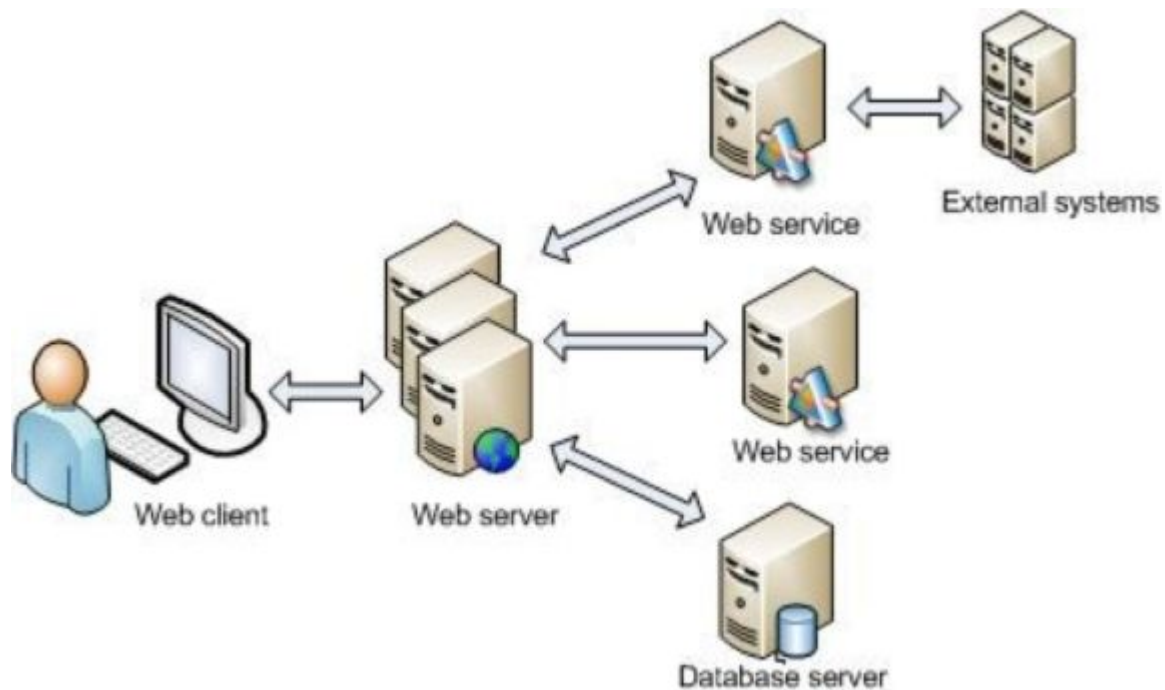


**Клиент - аппаратный или программный компонент вычислительной системы, который направляет запросы серверу.**

**Программа-клиент взаимодействует с сервером, используя определенный протокол. Она может запрашивать с сервера любые данные, манипулировать данными непосредственно на сервере, запускать на сервере новые процессы и т. п.**

**Полученные от сервера данные клиентская программа может предоставлять пользователю или использовать как-нибудь иначе, в зависимости от назначения программы. Программа-клиент и программа-сервер могут работать как на одном и том же компьютере, так и на разных. Во втором случае для обмена информацией между ними используется сетевое соединение.**

# Клиент



# Сервер



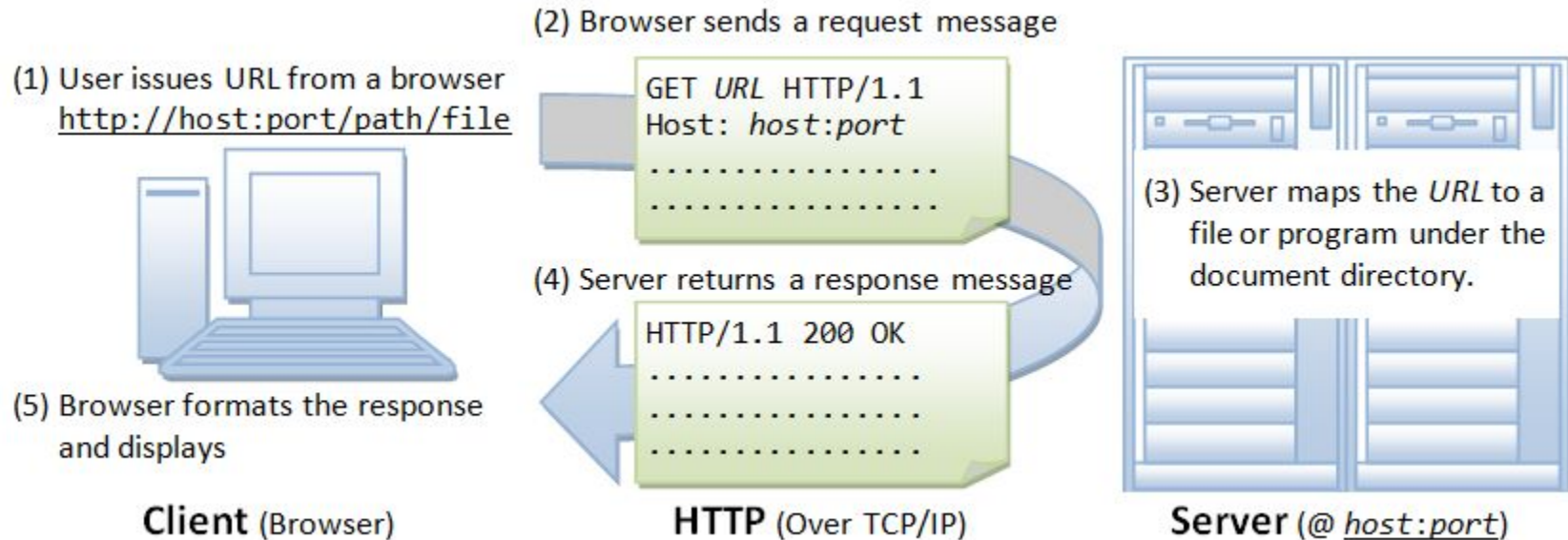
Веб-сервер (англ. Web Server) - это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, выдает им HTTP-ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-потокom или другими данными.

Веб-сервер - основа Всемирной паутины.

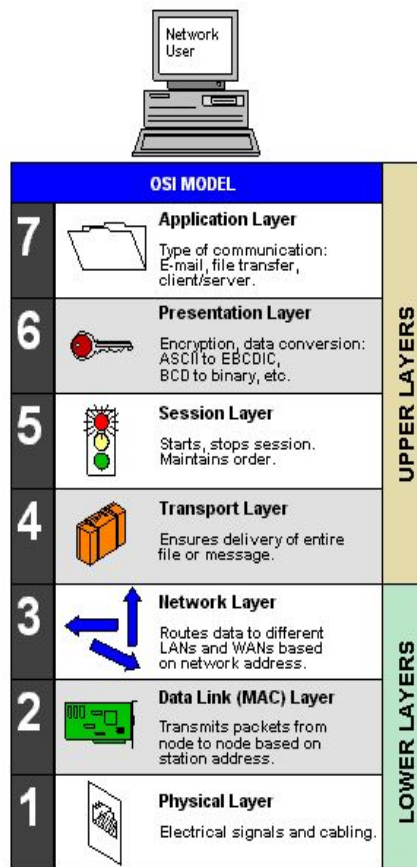
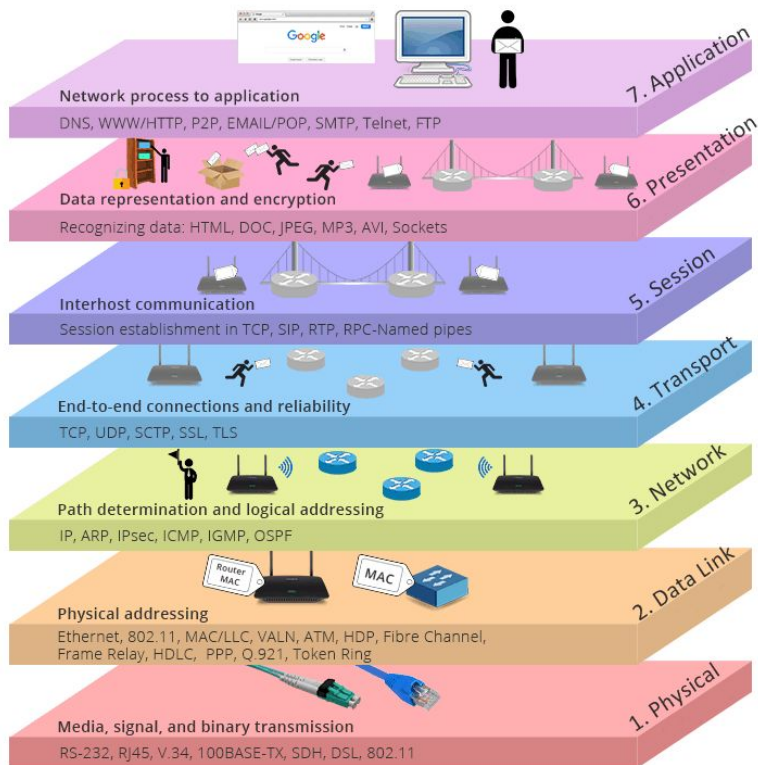
# Сервер



# Клиент-серверная архитектура



# 7 layers of OSI model





# HTTP протокол



HTTP - протокол передачи данных, используемый в компьютерных сетях. Название сокращенное от Hyper Text Transfer Protocol, протокол передачи гипертекстовых документов

Основным назначением протокола HTTP является передача веб-страниц (текстовых файлов с разметкой HTML), хотя с помощью него успешно передаются и другие файлы, связанные с веб-страницами (изображения и приложения), так и не связанные с ними ( в этом HTTP конкурирует с более сложным FTP).

HTTP предполагает, что клиентская программа - веб-браузер - способна отображать гипертекстовые веб-страницы и файлы других типов в удобной для пользователя форме. Для правильного отображения HTTP позволяет клиенту узнать язык и кодирования веб-страницы и / или спросить версию страницы в нужном языке / кодировке, используя обозначения из стандарта MIME.

# HTTP methods



1. OPTIONS
2. GET
3. HEAD
4. POST
5. PUT
6. PATCH
7. DELETE
8. TRACE
9. CONNECT

В основном используются методы GET, POST

Метод HTTP (англ. HTTP Method) - последовательность любых символов, кроме управляющих и разделителей, что указывает на основную операцию над ресурсом. Обычно метод представляет собой короткое английское слово, записанное большими буквами и чувствительно к регистру.

# GET



- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests should be used only to retrieve data

# POST



- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length

# POST vs GET

	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL  Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

# JSON Server



Install JSON Server: `npm install -g json-server`

Create a `db.json` file with some data

```
{  
  "posts": [  
    { "id": 1, "title": "json-server", "author": "typicode" }  
  ],  
}
```

Start JSON Server: `json-server --watch db.json`

Now if you go to <http://localhost:3000/posts/1>, you'll get

# Домашнее задание



1. Посмотреть видео: <https://www.youtube.com/watch?v=SzSXHv8RKdM>
2. Прочитать: [http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/http\\_basics.html](http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/http_basics.html)
3. Опционально\*: <https://learn.freecodecamp.org/>