

Въпроси за интервю - Мрежи

1. OSI модел: мрежови протоколи

#	Слой	Описание	Протоколи
7	Application	Позволява на по потребителските приложения да заявяват услуги или информация, а на сървър приложенията – да се регистрират и предоставят услуги в мрежата	DNS, FTP, HTTP, NFS, NTP, DHCP, SMTP, Telnet
6	Presentation	Конвертиране, компресиране и криптиране на данни	TLS/SSL
5	Session	Създаването, поддържането и терминирането на сесии. Сигурност. Логически портове	Sockets
4	Transport	Грижи се за целостта на съобщенията, за пристигането им в точна последователност, потвърждаване за пристигане, проверка за загуба и дублиращи се съобщения	TCP, UDP
3	Network	Управлява пакетите в мрежата. Рутинане. Фрагментация на данните. Логически адреси	IPv4, IPv6, IPX, ICMP
2	Data Link	Предаване на фреймове от един възел на друг. Управление на последователността на фреймовете. Потвърждения. Проверка за грешки. MAC	ATM, X.25, DSL, IEEE 802.11
1	Physical	Отговаря за предаването и приемането на неструктурирани потоци от данни по физическия носител. Кодиране/декодиране на данните. Съвързване на физическия носител	IEEE 802.11, IEEE 1394, Bluetooth

2. **Какво е мрежов адаптор?** – Осъществява връзката между компютърна система и мрежа (NetworkInterface)
3. **Какво е IP адрес?** – Всеки компютър, свързан към мрежа, се идентифицира с логически адрес (InetAddress)
4. **Какво е порт?** – Крайна точка на една комуникация. Използват се от TCP/UDP протоколите, за да идентифицират за кое приложение са предназначени данните
5. **Каква е разликата между TCP и UDP протокола?**

Характеристика	TCP	UDP
Connection	Connection-based	Connection-less
Надеждност	висока	ниска
Ред на пакетите	гарантиран	не гарантиран
Скорост на доставка	по-ниска от UDP	по-висока от TCP
Проверка за грешки	да, с възстановяване	да, без възстан.
Потвърждаване при получаване	да	не
Имплементация	ServerSocket/Socket	DatagramSocket

6. **Какво е сокет (socket)?** – Сокетите представляват крайните точки на двупосочна мрежова връзка (connection) между две приложения. Всеки сокет се идентифицира чрез комбинация от IP адрес и номер на порт
7. **Какво е сесия?** – Сесията е група от взаимодействия на потребител с уебсайт в рамките на определен период от време
8. **Характеристики на HTTP:**
 - Модел заявка-отговор(**request-response**) – служи за комуникационен канал в клиент-сървър архитектура
 - Не пази състояния (**stateless**) – всяка клиентска заявка е независима сама по себе си
 - Клиентът и сървърът съдържат информация един за друг, единствено по време на комуникацията (**connectionless**)
 - Всеки тип данни могат да бъдат изпращани, докато и клиентът и сървърът знаят типът (**media independent**)
 - HTTP транзакция
 1. Клиентът отваря комуникационен канал (TCP сокет)
 2. Клиентът изпраща заявка към сървъра
 3. Сървърът връща отговор на клиента
 4. Сървърът затваря сокета

- **HTTP: Заявка** – инициатор е клиентът – подава информация на сървъра, достъп до кой ресурс иска да получи и каква операция иска да извърши с него:

GET en.wikipedia.org/w/index.php HTTP/1.1

Connection: Keep-Alive

Host: en.wikipedia.org

1. Начален ред:

- **HTTP Метод** – указва типа операция, която клиентът иска да извърши със заявления ресурс
- **URL** – уникален локатор на заявления ресурс
- **Версия на HTTP** – версията на протокола, която ще се използва за комуникацията

2. Хедъри – Предоставят информация за заявката

3. Данни (Тяло) – опционални (GET Request няма тяло)

- **Основни HTTP методи**

1. GET – за зареждане на ресурс от сървъра

2. HEAD – идентичен с GET, с разликата, че отговорът няма да върне тяло, само хедъри

3. POST – изпраща данни (например от HTML форма) за обработка от сървъра. Данните се съдържат в тялото на заявката

4. PUT – ъплоудва даден ресурс

5. DELETE – трие даден ресурс

- **HTTP: Отговор** – изпраща се от уеб сървъра, като резултат от изпълнението на клиентска заявка: ***HTTP/1.1 200 OK***

Date: Tue, 18 Oct 2018 19:08:15 GMT

Server: Apache

1. **Начален ред:**
 - **Версия на HTTP**
 - **Статус код – обяснява резултата от изпълнението на заявката**
 - **Причина – кратко обяснение на статус кода**
2. **Хедъри – Предоставят информация за отговора**
3. **Данни (Тяло) – отговорите обикновено връщат данни, като тук най-често се съдържа HTML документът, получен на базата на клиентската заявка**
- **HTTP статус кодова – трицифрени кодове, идентифициращи какъв е резултатът от обработката на клиентските заявка, групирани в 5 категории, на базата на първата цифра**
 1. **Група 100 (Information) – Служат за „временни“ кодове, т.е. заявката е пристигнала, но сървърът все още не е готов с резултата**
 - **100 Continue**
 - **101 Switching protocols**
 2. **Група 200 (Successful) – Сървърът е обработил успешно клиентската заявка**
 - **200 OK**
 - **206 Partial content**
 3. **Група 300 (Redirection) – Ресурсът е наличен, но е разположен на друго място**
 - **301 Moved permanently**
 - **304 Not Modified**
 - **307 Temporary redirect**
 4. **Група 400 (Client Error) – Клиентска грешка**

- 400 Bad Request
- 401 Not Authorized
- 404 Not Found
- 408 Request Timeout
- 5. **Група 500 (Server Error) – Сървърна грешка**
 - 500 Internal Server Error
 - 501 Not implemented
 - 503 Service Unavailable
- **HTTP Хедъри:**
 1. **Основни (General Headers) – могат да се ползват едновременно и в заявки, и в отговори. Съдържат информация (мета-данни) за самото съобщение или за метода на комуникация**
 - Connection: keep-alive
 - Data: Sat, 17 Nov 2018 16:08:15 GMT
 2. **Завка (Request Headers) – специфични са само за заявката и могат да съдържат данни за самата заявка или за клиента**
 - Accept: text/html
 - Accept-Charset: utf-8
 - Accept-Language: en-US
 - User-Agent : Mozilla/4.0
 3. **Отговор (Response Headers) – съдържат информация (мета-данни) за сървъра и формата на съобщението**
 - Server: Apache
 - Allow: GET, HEAD

4. **Същински (Entity Headers)** – информация за самото съдържание на данни (тяло) и/или за ресурса, заявен от клиента
 5. **User Agent** – софтуер, който извършва действие от името на потребителя: E-mail клиенти, Web Browser-и, Месинджъри: Skype, WhatsApp
- **HTTP сесии** – За да може сървърът да корелира заявките, идващи от един и същ клиент като логически принадлежности към една последователност, се използват:
 1. **Бисквитки (Cookies)** – Cookie-тата са малки текстови файлове, генерирани от сървъра и изпратени на клиента в header-ите. Чрез информацията в тях може да се идентифицира сесията
 2. **Hidden fields в HTML forms** - HTML страницата съдържа форма с поле, което е скрито и не се визуализира в брауъра, но стойността му се праща като част от заявката и стойността му идентифицира сесията
 3. **URL rewriting** – добавяме в края на всяко URL данни, които да идентифицират сесията
 - **HTTP Port: 80 | HTTPs Port: 443**
 - **HTTP/2:**
 1. **Двоичен, вместо текстови**
 2. **Постига паралелизъм само с една TCP връзка между клиента и сървъра**
9. **Принципи на Representational State Transfer (REST)** – Стил софтуерна архитектура за реализация на уеб услуги, има за цел да подобри и улесни комуникацията между две системи
 - **Клиент-Сървър архитектура**
 - Клиентът и сървърът имат различни задачи

- Сървърът най-често съдържа бизнес логиката и се грижи за съхранението на данните
- Клиентът най-често извлича данни от сървъра

2. Предоставя единен интерфейс (конкрет)

3. Не пази състояния (stateless)

4. Предлага възможности за кеширане

5. Многослойна система

6. Предоставя code on demand

10. **Какво е JSON?** – JavaScript Object Notation (JSON) е един от най-популярните формати за обмен на данни и се използва за предаване на структурирани данни по мрежова комуникация

Java библиотека за работа с JSON: [google/gson](https://google.github.io/gson/)

Gson gson = new Gson(); String json = gson.toJson(object);

11. **Каква е разликата между:**

- **Encoding** – Процес, който променя данните в нов формат чрез схема, като данните не са защитени
- **Hashing** – Процес, който изчислява фиксиран размер математическо обобщение на данни. Хашингът не може да се 'reverse-не'. Предпочитана техника в удостоверявателния процес
- **Encryption** – Процес на защитене 'encoding', по такъв начин, че само удостоверени клиенти имат ключ или парола и могат да декриптират данните, за да получат оригинала