Теория

1. Интернет – кабел, който е заровен в земята или океана

* Представлява една огромна мрежа от много малки или големи по рамер мрежи (адреси)
* Мрежата може да е само едно устройство свързано към рутер, който е свързан към голямата мрежа(интернета)
* Мрежата(интернета) позволява обмен на информация между различните адреси
* Една мрежа представява минимум 2 устройства, които могат да обменят информация помежду си – Пример: Телефон и рутер (комуникират чрез пакети)

1. Как работи Web Server – Web клиент (браузър) изпраща заявка към сървъра, той чрез бакенда взима и обработва информация и зависимост заявката връща дадена информация обратно на браузъра.

* **Сървър** – машина, която обработва и получава информация и зависимост информацията връща даден отговор на клиента, който го изисква
* **Клиент** – машини или браузър, който получава дадения отговор от сървъра и го реализира по лесен начин за човека, който използва браузъра
* **Мрежов протокол** – те са основната комуникация между сърър и клиент
* Http, TCP, UDP, **IP**

1. **Пакети** – съобщение, файл, голям обем информация, която се разделя на много на брой битове(1 и 0). Тези битове се съхраняват в пакети, за да могат да се предадат от изпращача на получателя. Пакетите не винаги пътуват по един и същи път, за да достигнат до получателя. Когато стигнат до получателя, те се сглобяват в съответното съобщение или файл.

* Всеки пакет съдържа важна информация (headers), чрез която се разпознава дали съответния пакет е валиден или не
* Например: Откъде идва(изпращач), къде отива(получател), какво съдържа пакета, колко е голям пакета
* **Пакетите винаги имат един и същи размер**
* Има специални проверки за пакети, които проверяват дали той е пълен, дали има грешка в него и тн.

1. **DNS – Domain Name System (сървър от много домейни)** – сървър, съдържащ кеш от тонове имена на домейни и техните съвпадащи IP адреси.

* **DNS**  представлява Key Value Pair от адрес – домейн
* Когато в клиента се въведе домейн, то търси в DNS съръра със съответващия адрес на домейна

1. **TCP (Transport Control Panel) –** той се грижи за проверка и правилно пътувабе на изпратени пакети

* Служи за правилен път на пакетите, без да се отклоняват и за последователна огранизация на пакетите, за да не стават аномалии
* TCP използва header на съответните пакети, за да провери валидността и сигурността на пакетите, защото в headera се съдържа всичката важна информация за проверката на пакет
* TCP проверява дали всички пакети са в правилен ред. Ако установи, че даден пакет не съответства на това, което трябва да има в него, той бива изхвърлен.
* Също проверява пакетите се сглобяват правилно за реализиране на файла или информацията

1. **TCP VS UDP –** TCP набляга на надежността, докато UDP набляга на скоростта

* UDP не прави проверка на пакетите
* UDP не установява сесия и не гарантира доставка на данни
* UDP не се интересува дали данните са получени
* UDP е много педерастки начин за комуникации в мрежата

1. **Какво е OSI (Open System Interconnect) Модел в мрежовите комуникации**

* Състои се от 7 слоя
* Всеки слой обслужва слоя над него и в замяна се обслужва от слоя под него
* Ако го разберем, ще ни помогне за отстраняване на много проблеми
* ОСИ Модела служи за проверка на рекуеста и проверка на респонса, тоест проверка на заявката и отговора, който се връща – връзка между клиента и сървъра
* **Application Layer 7 - Разрешава на различни приложения да използват мрежата и да я представят на крайния потребител. ПРИМЕР: DNS, HTTP, FTP; Браузърите**
* **Presentation Layer 6 – част от операционната система; Служи за да може програмата да разбира, компресира и криптира входящи и изходящи данни. ПРИМЕР: ASCII към EBCDIC**
* **Session Layer 5 – Служи за стартиране, прекратяване, терминиране на определена сесия; ПРИМЕР: Sockets**
* **Transport Layer 4 – сърцевината на слоевете; Служи за транспортиране на данни и логическа комуникация между програмата и мрежата. ПРИМЕР: TCP, UDP**
* **Network Layer 3 – Служи за правилен транспортиране на пакети от изпращач до получател. Транспортира пакетите зависимост дали е LAN или WAN. Пример: IP**
* **Data Link Layer 2 – Некъв много нисък слой не го схващам ма е важен lmao**
* **Physical Layer 1 – физически може да се пипне; ПРИМЕР: USB, Blootooth, рутер.**
* **ВСИЧКИ МОДЕЛИ СА ВЗАИМНОСВЪРЗАНИ И ЕДИНИЯ НЕ МОЖЕ БЕЗ ДРУГИЯ**

1. Middleware – n на брой пътища, през които URL адреса трябва да мине и да бъде проверявана информацията от него чрез вътрешна логика, преди да бъде обработвана от сървъра.
2. Cookies – малък файл с обикновен текст, което съдържа малка част от данни и информация за конкретен клиент или уеб сайт

* Използва се най-често за state – ПРИМЕР: Да се запази резултат от игра на потребителя, да се запази продукт в количка; В него може да се запази и предпочитание на потребителя за сайта ПРИМЕР: Потребителя иска сайта да му е черен, а не бял, затова в кукито се запазва това като предпочитание на потребителя и в сървъра с валидации в Метода Георги, които връща черен цвят се връща черен цвят на сайта.
* Също се използват за да се знае дали потребителя е влязал или не и с кой профил е влязал; За да се запази активността на сърфирането на потребителя;
* Изпраща се в хедъра на заявката; Използва се като JWT Token
* Когато имаме **СЕСИЯ** е задължително в кукито да има само name: и value: , като когато се прати заявка към сървъра той търси този name и value в базата и зависимост дали се съдържа в базата връща нужната информация ПРИМЕР: От заявката се праща JSON с name: sid value: 5 и в базата се съдържа name: sid value: Tom. Като резпонс ще се върне името Том.
* Сесията представлява куки, в което се съдържа само и единствено уникално ID на сесията

1. Аутентикация и ауторизация

* **Аутентикация** е процес на проверки на самоличността – Кой си ти?
* **Ауторизация** е процес на определяне на какво поже да прави потребителя – Имаш ли право да правиш това?

1. Разлика между **Сесия** и **JWT Token**

* Разликата е че Идто на сесията се генерира рандом, докато jwt token се генерира посредством скрита парола и се криптира зависимост параметрите, които съдържа ПРИМЕР: Сесията е куки което съдържа само session id, докато jwt token съдържа различни параметри като username and role. И двете се съхраняват в браузъра.
* Най-голямата разлика е че стейта на юзъра при jwt token не се съдържа в сървъра, а в самия jwt token
* Jwt token е по стабилно, Когато има много юзъри, които пращат заявки **при сесиите** тези заявки ще се приемат по бавно и по-натоварено
* Единствения проблем е че големината на jwt tokena е масивно по голяма от тази на сесията, защото в jwt tokena се съдържа не само много информация, а и метод на криптиране на самия jwt token.