**Лекция Introduction off internet**

1. **Какво е мрежови протокол** – система от правила, команди, заявки които позволяват на две системи да комуникират помежду си – ТСР протокол, IP протокол
2. **Какво е DNS** – домейн нейм систем, гигантска база данни която мапва домейн имена към IP адреси, като самата система има дървовидна структура
3. **Какво е turing complete машина** – всички съвременни компютри
4. **Обяснете OSI модела**
5. **Разлика между ТСР и UDP –** при ТСР се залага на надеждност на прехвърлените данни, а UDP е сент и форгет и там не е важно за изпращача дали получателя е получил някой пакет
6. **Кой е първия програмист –** Ада Ловлейс

**Лекция HTTP protocol**

1. **Обяснете кодовете, които започват с 1хх** информативни кодове за смяна на протокол, дава информация на клиента че може да продължи нататък**, 2хх** успешна операция **, 3хх** редайрект**, 4хх** клиентска грешка **и 5хх –** вътрешна грешка
2. **HTTP/2 - какво е multiplexing**  възможността да се пратят много заявки по една единствена връзка request – respons по една единствена ТСР връзка **и server push** когато сървъра връща множество файлове наведнъж без да са му поискани
3. **Как работи TLS/SSL** криптиран канал на 6 ниво в ОСИ модела, разменя се един ключ за криптиране на връзката
4. **Кавкви HTTP глаголи/операции познавате?** Гет за вземане на ресурс, пост за създаване, , пут за ъпдейтване, пач за парциален ъпдейт
5. **Опишете структурата на едно URL.** Започва с протокол :// пост: порт:път , куери параметърс и най накрая фрагмент
6. **Каква е разликата между URI и URL?** Юрл е подмножество на юрай вътре съдържат механизъм за вземане на ресурс
7. **Разлика между PUT/POST/PATCH.** Пост –създаване, пут – ъпдейтване, пач – парциален ъпдейт
8. **Какво представлява модела POST-redirect-GET и защо се използва?** Когато направим пост заявка за създаването на някакъв ресурс, той се създава на сървъра и не е много правилно да върнем директно риспонс, защото ако се рефрешне браузъра той ще се опита да създаде формата наново. Затова редиректва към новия ЮРЛ и браузъра прави гет заявка

**Лекция Spring Boot Introduction**

1. **Какво представляват Beans в Spring?** Различни обекти общото между тях е че се менажират от спринг т.е. не използваме оператора new, за достъп до тях използваме инвършън от контрол и инжекшън
2. **PostConstruct анотация?** Слага се върху метод който да се извика след инициализиране на конструктора
3. **PreDestroy анотация?** Слага се на метод който се извиква преди дестройва/унищожава/ шът даун
4. **Има ли разлика между application.properties и application.yaml (отнася се за предназначението).** не
5. **Какви bean scopes познавате, дайте примери и какви са разликите между тях?** Основните са сингълтон и прототайп, сингълтон се инициализара един единствен път, прототайп се създава всеки път, сешън бийнове, рекуест бийнове, къстъм скоупове
6. **Какво е Spring Boot?** Начин по който можем да си сетъпнем нашето спринг приложение, предоставя удобен депендънси мениджмънт, автоконфигурация и т.н
7. **Само на Java ли се пишат спринг приложения?** Не и на други JBM езици

**Лекция Spring MVC**

1. **Как анотираме контролер класовете си?** Чрез анотация Контролер също когато връщаме рест АПИ
2. **Как използваме @RequestMapping анотацията?** Върху клас левъл тя дава префикс на всичките методи (/user) това означава че за да се извика гет заявката трябва в пътя юрл да има този префикс
3. **Има ли разлики между @GetMapping и @RequestMapping с метод Get върху метод?** Не няма, едно и също нещо са
4. **Как може да вземем POST параметър?** Със помоща на ДТО на байнинг модел или с рекуест парам анотацията
5. **Как предаваме обект съм Thymeleaf view?** С помоща на методите които спринг ни позволява да инжектираме Модел, Модел Мап и Модел енд Вю
6. **Как може да вземем optional параметър?** В рекуест парам анотацията има опшънал който може да има стойност тру, фолс
7. **Как може да зададем default стойност на параметър?** В рикуест парам анотацията има възможност да сложим дефолт велю така че ако не дойде такъв параметър в заявката вместо грешка 400 да инжектираме дефолтната стойност
8. **Каква разликата между Path variable и Query parameter?** Едната е стойност която се кодира в пътя на ЮРЛ а другото е в куери стринга

**Лекция State Management**

1. **Какво е Cookie?** – малък текстов файл, който се изпраща от сървъра на клиента, като клиента е например web браузъра, този текстов файл не съдържа изпълним код
2. **За какво служат cookies?** – служат за персонализация, за юзър проследяване, и най-вече се използва за session management
3. **Какво е third party cookie?** – използват се за юзър тракинг и реклами
4. **Какво е HTTP сесия? –** област в паметта на сървъра или ДБ, където се пази информация за даден юзър, тя е свързана с даден номер – сешън ID, което се предава между сървъра и клиента, като това става с куки
5. **Как може в Spring до добавим атрибут в HTTP session? –** като инжектираме НТТР сешън като аргумент и го използваме с метода ADD ATRIBUTE
6. **Как може да се четат кукитата в контролер?** – с анотация за куки
7. **Как може сървърът да изтрие едно Cookie? :-)** – като сетне същото куки със същото име и пропъртите с мax age 0

**Лекция Spring Essentials**

1. **Какво е thymeleaf? –** представлява html темплийт енджин, който ни дава възможност да създаваме html темплийски и с помоща на него да създаваме динамични страници и да попълваме дадени стойности в тези темплийти
2. **Какво представляват Thymeleaf фрагментите?** – реюзъбъл парченца от нашите темплийти , неща които се срещат на няколко места
3. **Защо използваме link expressions? –** за да може с помоща им да генерираме правилни линкове, дори с пътища в тях и куери параметри, или ако имаме сървлет контейнер и на него могат да вървят няколко апликейшъна и всеки един има контекстен път localhost8080:/app1/ , и линк експрешъните ще се усетят кои са правилните апликейшъни и да стартира правилния
4. **Обяснете за какво служи @Primary анотацията. –** когато имаме няколко бийна за един и същ тип и искаме да инжектнем един от тях, за тази цел използваме тази анотация , а ако се използва LIST тогава всички бийнове ще се инджектнат
5. **Какво е Same Origin и CORS?** – сейм ориджин е полисито което всички браузъри имплементират и то забранява да се изпълняват JS които идват не от сървъра който хоства сайта, корс – съвкупност от правила които позволяват чужди за нашия сайт скриптове да бъдат изпълнявани
6. **С коя анотация управляваме CORS? -** @CrossOrigin
7. **Пример за употреба на CORS. –** когато имаме спринг приложение, което експоузва Spring API и имаме друго приложение което искаме да консумира това REST API и то се намира на друг сървър, тогава използваме КОРС
8. **За какво се използва @ModelAttribute върху метод? –** начин да инициализираме някаква стойност в нашите модели

**Лекция Thymeleaf and Validations**

1. **За какво служи стартера spring-boot-starter-validation? –** има две основни неща, има депендънси към JavaX API съдържа валидации нот нъл, сайз за ДТО валидациата, за да сработи всичко ни трябва и валидатор и по подразбиране използва Hibernate валидатора за да може да разапознава тези анотиции които използваме и
2. **Как се използва анотацията @Valid? –** може да се сложи на аргумент в нашия контролер, взема имплементацията и му дава байдинг модела за да извърши валидацията, тази валидация се основава на анотациите в ДТОто + къстъм валидаторите
3. **Как се имплементира custom валидатор? –** трябва да имаме една анотация и тя има мета анотация върху нея и в мета анотацията даваме класа който върши работата по валидацията, например че паролата е еднаква с потвърдената парола
4. **За какво се използва BindingResult? –** контейнер за грешките които са постъпили при процеса на валидация
5. **Какво става ако имаме аргумент на метод анотиран с @Valid и следващия аргумент не е BindingResult? –** хвърля грешка 400 рекуеста на клиента не е валиден, която се връща на клиента
6. **Обяснете за какво служи POST-redirect-GET патърна? –** когато постнем нещо за да създадем обект и имаме грешка при валидацията ако не направим редирект а върнем юзъра на същата страница и той натисне рефреш бутон, то същите данни ще бъдат изпратени отново, също така ако е успешна цялата валидация и клиента успешно е направил поръчка и сме му заредили нова страница с номер на поръчката, ако рефрешне ще направи още една поръчка. Докато при този метод постваме заявката и редиректваме към посочената страница и ако натисне рефреш му се показва същата страница без да се изпращат данни

**Лекция REST API**

1. **За какво е анотацията @RequestBody? –** служи за да кажем на спринг че трябва да сериализира бодито на НТТР рекуеста към някакво ДТО
2. **Какво е @ResponseBody?** Дава някаква инструкция на Спринг да сериализира дифолтно към JSON обект който се връща от контролера
3. **Каква е разликата между @Controller и @RestController? –** Реста е комбинация от Контролер + респонс боди
4. **Как може да контролираме HTTP response status-a на response-a, който връщаме на клиента? –** с ResponsEntity той ни позволяза да моделираме статуса, също така @ResponsStatus
5. **Какво е RestTemplate? –** клас с който можем да консумираме Рест сървиси, например в нашия сайт ако искаме да вмъкнем актуална прогноза за времето и някой сървър ни дава такава с РЕСТ Темплийт можем да дръпнем информацията
6. **С какво ще бъде заменен RestTemplate в бъдещи версии на Spring? –** с реактивния уеб клиент
7. **Защо да ползваме REST API?** За да имаме удобен достъп до данни и функционалност, която се намира на някой сървър, така че да можем да консумираме тези данни с различни клиенти
8. **6 принципа на REST архитектурата? 1.**Данните да могат да се свалят по определен начин **2.** Клиента и сървъра могат да се развиват независимо един от друг **3.**всяка заявка се третира като нова, не се помнят предните **4.**url-a се рестартира на по-голям период от време за да не се изпращат постоянни заявки **5.** Различни компоненти на системата могат да са на различни машини **6.** В отговора към клиента се качва код който е изпълним при клиента
9. **Какви НТТР глаголи познавате? –** Get, PUT, POST, PATCH, Delete
10. **Разлика между PUT/POST/PATCH. – POST-**създава обект на сървъра, **РUT –** update на целия обект на всички негови пропъртитар **PATCH** – маршъл ъпдейт, ще промени само полетата които са включени в тази заявка
11. **За какво служи fetch API? –** скрепваща лепенка, издърпва от програмата или базата информация под формата на json и я представя във view което сме направили
12. **Как анотираме контролерите, които служат за имплементиране на REST API?** @RestControler
13. **Как анотираме методите за Get/Post/Put/Patch и т.н.? -** @GetMapping, @PostMapping
14. **Кога връщаме следните HTTP статуси:**
    * **400** клиентска грешка, например ако искаме книга по нейното Id и не сме изпратили правилно id-то връща тази грешка
    * **409** конфликт, когато един ресурс който искаме да модифицираме се намира в неподходящ стейт, например добавяне на продукт към ордер, ако ордера вече има статус изпратен, той вече не може да приема повече продукти
    * **404** ресурса го няма
    * **401** нее ауторизиран
    * **403** не е аутентикиран
    * **201** връща се при пост когато криейтваме ресурс и е минало успешно
    * **200** генерален код – Всичко е наред със заявката
    * **204** no content, когато стартираме бекграунд таск който ще се изпълнява, или ако изтрием нещо

**Лекция Spring Security**

1. **Какво e Filter?** Част от Сървлет Апито, използва се от сървлет контейнерите, които генерират HTML, с негова помощ модифицираме http request-a и http respons-a
2. **Каква е разликата между authentication –** отговаря на въпроса кой съм аз **и authorization –** би дала отговор на въпроса Какво мога да правя**?**
3. **Какво е Principal?** Представлява текущо аутентицирания юзър, т.е. обект, който го представлява
4. **Какво е CSRF? Пример.** Неприятна атака. Ако сме логнати на сайт на който имаме доверие, и отворим един гаден сайт, той може да ни накара да изпълним заявка към добрия сайт и пътувайки тази заявка /например преведи пари от мен към друг акаунт/ тя пътува с всички кукита и добрия сайт може да се заблуди че заявката е автентична, лошия сайт използва сешън кукитата ни. За да се предпазим използваме CSRF Token , той няма как дъ бъде разпознат от лошия сайт
5. **Няколко думи за интеграцията между Spring Security и Thymeleaf?** Представлява стартер Spring Thymeleaf Security дава възможност да използваме някой от фичерите на Security в нашите Thymeleaf темплийти. Например дали юзъра има някаква роля , за да има право на достъп до даден обект.
6. **Какво е UserDetails и за какво се използва?** Интерфейс и е част от юзър фреймуърка, и в клас който имплементира **UserDetails** даваме на спринг всички детайли за юзъра от които той би имал нужда за да работи успешно примерно нашия **login**
7. **Каква е ролята на UserDetailsService?** Имплементацията на този интерфейс от юзърнейм да ни върне детайлите на юзъра
8. **Какво е PaswordEncoder и каква е ролята му при form Login?** Клас който може да хешира паролата на юзъра, и дава възможност да съхраняваме хеша, като после дава възможност да бъде сравнявана
9. **Как се представят permission-s на юзърите в контекста на спринг секюрити?** Част от **UserDetails** интерфейса, имплементират Granted Authority и всеки един негов инстанс представлява пермишън.

**Лекция HATEOAS**

1. **Кавкво е HATEOAS?** Последното ниво, т.е. глори ъф рест, където заедно с Json респонса се връщат и линкове, те могат да се навигират, и да изпълняваме различни операции върху ресурса
2. **Какво е HAL?** Полустандарт, който е замръзен, но **HATEOAS API** го използва, стандартен начин да се опишат линковете и ресурсите които идват в респонса
3. **Предимства на HATEOAS API?** Най-голямото предимство е че сме независими от структурата на URL-ите и че можем да го навигираме
4. **Недостатъци на HATEOAS API?** Изисква доста повече усилие разработката му
5. **Какво е spring data rest и каква е връзката му с HATEOAS?** Експоузваме директно ентитатата си през REST, връзката му е че **spring data rest** използва **HATEOAS** респонси за навигиране по ресурсите му

**Лекция Error handling**

**1. Защо е необходим error handling?** Първо за да може нашият сайт да е юзър френдли, да имаме по голяма сигурност, по лесно да може да обработим грешката и да бъдем нотифицирани за нея

**2. Какво представлява whitelabel error page и от къде идва?** Идва от спринг буут, дефолтна страница за грешка, заменя страницата за грешка на tomcat

**3. Как може да обработваме exception-s локално за контролер?** Със анотацията Ексепшън хендлър, в контролера да имаме метод който да обработва грешката и да е анотиран така

**4. Как може да обработваме exception-s глобално за приложението?** Със Контролер Адвайс, клас който се анотира със тази анотация в него можем да слагаме ексепшън хендлъри

**5. Как може да определяме HTTP status кода на респонса спрямо exception-a?** Когато анотираме самия ексепшън с респонс статус анотация

**6. Как може да сложим свой error page**? Когато сложим Таймлийф темплийт с име error, тогава спринг ще го хване автоматично и ще го покаже вместо white label page, 2. Спрямо статус кода на отговора, Във фолдър errors който можем да го създадем като дете на темплийтс, да сложим страници които са именувани статус код с имената на грешките пример 404.html

**7. Как може да сложим error page спрямо HTTP status кода?**

**8. Как може да върнем custom response при грешка в REST котролер?** със локален ексепшън хендлър който е специфичен за контролера, когато го обработим и в respons body-то върнем някакъв обект, който е къстъм полета

**Лекция Events**

**1. Опишете механизма на изпращане и получаване на events в Spring.** Обзървър патърна, имаме клас който енкапсулира данните на ивента , имаме един енджин на който подаваме ивента и той се изпълнява, имаме лисънъри които обработват и нотифицират за получаването на ивента и го обработват

**2. Какво е необходимо за да направим custom event?** 1. Имплементираме в клас AplicationEvent и да добавим методи със свои данни за които искаме да слушаме, трябва да инжектираме компонента който файърва ивента и трябва да анотираме един метод с EventLisener и да дадем в скоби типа на къстъм ивента, и в него метод можем да обработим този къстъм ивент

**3. Как се обработват events когато имаме няколко listeners - паралелно или последователно?** Последователно , т.е. изчакват се

**4. Какво е необходимо за да се ползва task scheduling в проект?** Да се сложи EnableSheduling върху RUN класа на апликейшъна , върху някой методи да бъдат анотирани с Shedule анотации, и те се пускат със **scheduled task** във бекграунда

**5. Как може да се зададе време за стартиране на scheduled task?** Crom, fixedRate и fixedDelay

**6. Разлика между fixedRate и fixedDelay.** В това че Delay изчаква определено време между стартиразнията а другото не изчаква може да стартира веднага ако се застъпват интервалите

**7. Какво означава крон израза "5 0 \* 8 \*"?** да стартира на 5 минути след полунощ през целия месец август

**8. Колко полета имат крон изразите в Spring?** 6 като има поле за секунди

**9. За какво служи Cacheable анотацията?** За кеширане на резултата който връща някакъв метод, тя идва като използваме спринг кеш стартера

**10. Как се трие кеш?** Със кеш евикт анотация

**11. Как се ъпдейтва кеш?** Със кеш пит анотация

**Лекция AOP**

1. **Какво означава AOP (концепция)?** Аспект ориентирано програмиране, подход при който някакви функционалности се отделят от останалия код и могат да се прилагат на различни места в нашете приложение, тези места **са join point-ове**
2. **Защо ползваме AOP?** За да може някакви неща /**крос кътинг**/ които са общи за приложението да ги отделим и да намалим мейтенънса и да улесним писането на приложението, изкарваме неща които не са типични за метода извън него
3. **Какво означава терминът "cross cutting concern"?**  парченце код чиято логика е валидна за много места в програмата
4. **Какво означава Join Point?** Някакво действие в нашата програма, изпълнение на метод
5. **Какво означава Advice?** Парченце код
6. **Какво означава Aspect?** Мястото на което можем да поставим парченцето код
7. **Какво означава Weaving?** Процеса който извършва за да може АОР да работи, т.е. създаването на проксита, закача крос кътингите към джойн пойнтовете
8. **Какво означава AOP Proxy?** Когато един клас обгръща друг клас, излага същия интерфейс за външния свят, само че прихващаме изпълненията на някой методи и правим някой неща преди или след тях.
9. **Как се генерират проксита в спринг?** С една библиотека глип, която е за манипулация на байт код и в рънтайм генерира нови проксита
10. **Стъпки за имплементацията на примерен advice/aspect.**  Депендънси спринг стартер АОР с негова помощ се дърпа аспект J анотациите, да анотираме един клас със аспект, класа трябва да е менидж бийн, @Component и вътре слагаме един пойнт кът експрешън, като език с който се селектват методи, в контекста на спринг и след това си дефенираме самия адвайс (араунд, бифор, автър ....) и вътре в метода слагаме парченцето код

**Лекция Testing**

1. **Какво представляват юнит тестовете?** Програмка която тества нашите класове в изолация от всичко останало, без депендънсита и без инфраструктура
2. **Какво е е мок? За какво се използва?** Фалшиви дъми класове, които бихме могли да инжектираме вместо дипендънсита, така че да изпълним един тест в изолация, те не правят нищо и могат да се обучават да връщат елементарни отговори
3. **Как се създава мок (популярна библиотека)?** Mockito
4. **Какво е интегрейшън тест?** Тества цялото ни приложение не в изолация а сглобена с всички дипендънсита и ако има връзка с външна система
5. **Какво е in-memory-db и приложение при тестовете.** База данни като всяка друга, тя работи в паметта и замества истинската база по време на тестовете. Използва се защото тестовете се изпълняват на кантиниус интегрейшън сървис. Продукт който се стартира и стои в паметта и по време на тестовете се връзваме към тази база. След като приключи тестването тази база изчезва със всичките данни в нея
6. **Какви други in-memory продукти познавате?** HSQL DB, H2, Мария ДБ, JMS
7. **Какво е MockMVC и как се сетъпва то?** Клас който можем да използваме за да изпраща фалшиви заявки към сървъра, но ние ги мокваме и билдваме все едно истински юзър от истински браузър изпраща тези заявки
8. **Пример за използване на MockMVC.** Гет заявка към контролер и да проверим дали модела е попълнен правилно
9. **Как можем да сетъпнем принципал когато тестваме контролер защитен от спринг секюрити?** Със мокюзър анотацията

**Containerization and documentation**

1. **Какво е VM? -** Виртуална машина (VM) е софтуерна програма или операционна система, която не само показва поведението на отделен компютър, но също така е способна да изпълнява задачи като изпълнение на приложения и програми като отделен компютър.
2. **Защо може да искаме да ползваме VM?** Виртуалните машини често са създадени за изпълнение на определени задачи, които са различни от задачите, изпълнявани в хост среда.
3. **Какво е hypervisor? -** Хипервайзорът е софтуер, който съществува извън гостната операционна система, за да прихване командите, изпратени до компютърен хардуер.
4. **Какво е container runtime?**
5. **Какво е docker?** Да го кажем с най-прости думи, това е набор от свързани софтуер-като-услуга (SaaS) и платформа-като-услуга (PaaS) продукти, които използват виртуализация на ниво операционна система за разработване и доставяне на софтуер в пакети, наречени контейнери.
6. **Разликата между Container и Image?** Популярен начин за обяснение на изображения и контейнери е сравняването на изображение с клас и контейнер с екземпляра на този клас. Слоевият подход на изображенията и контейнерите на docker помага за поддържане на малкия размер на изображенията и контейнерите.

Контейнерите се създават от изображения. Когато контейнер е създаден от изображение, върху него се поставя тънък слой за четене/запис

1. **От какво се състои един docker image?** Докер изображение се създава от множество слоеве. Ако вземем пример за Dockerfile, всяка инструкция се преобразува в слой
2. **Какво е container registry? -**  е едно място за вашия екип за управление на изображения на Docker, извършване на анализ на уязвимостта и решаване кой да има достъп до какво с фино контролиран достъп.
3. **Защо се налага да си документираме REST API?**
4. **Какво е Swagger? Swagger UI** е сървър, чрез който се визуализира разработен API така, че да може да бъде изследван и тестван.
5. **Какво е Open API?** Отвореният API не е нищо друго, освен API, който е бил публично достъпен, така че приложенията на трети страни да имат достъп до бек-офис услуги / приложения, без каквото и да е средно ниво. Това е начин за опростяване на интеграцията между външните приложения и основните функционалности на бизнеса.
6. **Как може да документираме REST API в Spring среда, популярни решения.**