**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?**
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**
    * **409**
    * **404**
    * **401**
    * **403**
    * **201**
    * **200**
    * **204**

**Лекция Spring Security**

1. **Какво представлява Filter?**
2. **Каква е разликата между authentication и authorization?**
3. **Какво съдържа SecurityContextHolder?**
4. **Какво е principal?**
5. **Опишете една CSRF атака?**
6. **Как Спринг предпазва от CSRF?**
7. **Няколко думи за интеграцията между thymeleaf и spring security?**

**Лекция HATEOAS**

1. **Кавкво е HATEOAS?**
2. **Какво е HAL?**
3. **Предимства на HATEOAS API?**
4. **Недостатъци на HATEOAS API?**
5. **Какво е spring data rest и каква е връзката му с HATEOAS?**

**Лекция Error handling**

1. **Какво е whitelabel error page? Как може да я махнем или заменим?**
2. **Какво е поведението на custom exception (дефиниран от нас ексепшън) анотиран с @ResponseStatus?**
3. **Как може да имплементираме exception handling специфичен за контролер?**
4. **Как може да имплементираме глобален exception handling?**
5. **Каква е ролята на HTTP status кодовете връщани от REST API-тата?**
6. **Може ли да инжектираме Model или ModelView в exception handler?**

**Лекция Events**

1. **Какво е вграден (built-in event) в Spring? Пример.**
2. **Как може да слушаме за event?**
3. **Какво е необходимо за да създадем и използваме собствен event?**
4. **Какво е Scheduling?**
5. **Как може да се зададе времето за изпълнение на един бекграунд таск?**
6. **Какво е необходимо да се направи за да се активира scheduling функционалността на Спринг?**
7. **Как може да се кешира връщана стойност от Спринг компонент?**
8. **Какво е разликата между анотациите @CachePut и @CacheEvict?**

**Лекция AOP**

1. **Какво е AOP?**
2. **Какви са предимствата, които ни предоставя AOP?**
3. **Какво е cross-cutting concern?**
4. **Какви типове advice-и позвнавате?**
5. **Какво е JoinPoint?**
6. **Какво е PointCut expression?**
7. **Как може да се подредят аспекти?**

**Лекция Testing**

1. **Какво е Unit test?**
2. **Какво е Integration test?**
3. **Какво представлява mocking и защо се използва, пример.**
4. **Обяснете т.нар. triple A.**
5. **Code first vs Test first development.**
6. **Какво е тестова пирамида и кое се приема за правилна тестова пирамида?**
7. **Какво е параметризиран тест?**
8. **Пример за embedded решение за integration testing.**