**SPRING**

50. Significance of springboot over spring.

51. Different annotations (Authorized Requested Parma etc).

52. Difference b/w @ component and @ service.

53. @ Qualifier.

54. Define MVC architecture of SpringBoot.

55. Spring Initializer .

56. Spring Actuator.

57. SpringBoot Scope of Bear.

58. Default scope of bear.

**Hibernate**

59. What is Hibernate?

60. What is ORM?

61. What is HQL?

62. Difference between JPA and Hibernate?

63. Difference between JDBC and Hibernate?

64. What is JPA?

65. What is persistent in Hibernate?

66. What is session in Hibernate?

67. What is session factory?

68. What is session?

69. Difference between **get() method and load() method**.

70. What is lazy loading in Hibernate?

71. How many types of association mapping is there in Hibernate?

72. What are the key components are in Hibernate?

What is transaction in Hibernate

73. What is the significance of Hibernate over JDBC.

74. What is Session and Sessionfactory in Hibernate.

75. What is Dialect in Hibernate.

76. Mappings in Hiberate.

77. What is lazy Eager Loading in Hibernate in Hibernate.

78. How do we declare multiple primarykey in Hibernate.

79. What is a POJO Class.

80. What Parameter that Repository accepts.

**Advanced Java/ J2 EE**

172. Write down the code for JDBC.

173. Difference between **doGet() and doPost() method.**

174. Explain Servlet life cycle.

175. What is session in servlet?

176. What is request dispatcher?

177. What are the features available in Spring Boot?

178. What is spring initializer?

179. Difference between spring and spring boot.

180. What is spring IOC container?

181. Explain bean life cycle.

182. What are those spring starter available in spring boot?

183. What do you mean by Dependency Injection?

184. What is bean factory?

185. Bean scope?

186. What all annotations you know in spring boot?

187. What is spring MVC?

188. What is spring profile?

189. What is spring security?

190. What is application , property file in spring boot?

191. What is spring actuator?

192. How to connect spring boot with the database?

193. Difference between @Controller and @RestController.

194. How to use two database in spring boot?

195. What is web services? Types of web services?

196. Difference between Monolithic and Micro services?

197. How you tested web services? Tool you used.

198. What is idempotent?

199. Status codes 201, 501, 500, 203, and 404.

200. What is PACT and OAUTH?

201. What is post man?

202. What all HTTP method you used in your project?

203. How to expose web service and consume web service.

204. How to send email and write code? Which dependency you added for this?

205. What all micro services you know?

206. How you managed you codes or How to share your codes with seniors?

207. Which design pattern you follow in your project?

208. What all configuration you have done in your project?

**209. Explain project flow.**

210. Difference between **JPA and Hibernate.**

211. What is **Get and Post method.**

6

212. What is ORM technique.

213. Annotations used in Hibernate.

214. What is HQL.

215. @Value annotation.

216. Difference between Spring and SpringBoot.

217. @Qualifier,@Autowired ,@Component, @SpringBoot.

218. Which annotation starts the SpringBoot Project.

219. What is Bean Life Cycle and what is Spring IOC.

220. Difference Between @Controller and @RestController.

221. What are Stream API’s.

222. What are Webservices and Microservices.

223. REST and SOAP API’s.

**SPRING BOOT**

337. Spring Vs Spring Boot (Advantages, Disadvantages, Which one is better).

338. Spring Initializer.

339. The Spring Initializr is ultimately a web application that can generate a Spring Boot

project structure so that we can import that into the IDEs like eclipse…

340. Spring Acuator.

341. Spring Boot Actuator is **a sub-project of the Spring Boot Framework**. It uses

HTTP endpoints to expose operational information about any running application.

342. The main benefit of using this library is that we get health and monitoring metrics

from production-ready applications.

343. Spring Boot starter.

344. Spring Boot Starters are **dependency descriptors that can be added under the**

**<dependencies> section in pom. xml**. There are around 50+ Spring Boot Starters for

different Spring and related technologies. These starters give all the dependencies under

a single name.

345. Spring MVC.

346. A Spring MVC is a Java framework which is used to build web applications. It follows

the Model-View- Controller design pattern. It implements all the basic features of a core

spring framework like Inversion of Control, Dependency Injection.

347. Flow of Spring MVC.

348. Advantages of Spring MVC over other framework.

349. Excellent support for developing Restful web services.

350. **Separate roles** - The Spring MVC separates each role, where the model object,

controller, view resolver, Dispatcher Served, validate, etc. can be fulfilled by a specialized

object.

351. **Light-weight** - It uses light-weight served container to develop and deploy your

application.

352. **Rapid development** - The Spring MVC facilitates fast and parallel development.

353. **Flexible Mapping** - It provides the specific annotations that easily redirect the page.

354. Declaration of different layer in Spring Boot.

355. How to connect Spring Boot application to database using JDBC.

356. What is spring starter parent dependency tag.

357. 6.How you exposing data and how do you consuming it.

358. Bean factory vs Application context .

11

359. One difference between the bean factory and application context is that the former

only instantiates bean when you call get Bean() method while

Application Context instantiates Singleton bean when the container is started, It doesn't

wait for the get Bean to be called.

What is ORM ?

**JDBC Hibernate**

It is database connectivity technology

It is a framework,

It does not support lazy loading

Hibernate support lazy loading

We need to maintain explicitly database

connection and transaction.

Hibernate itself manage all transaction

Low performance

High Performance

360. Session and Session Factory in hibernate.

361. Session Factory is a class for Session objects. It is available for the whole application

while a Session is only available for particular transaction.

362. Session is short-lived while Session Factory objects are long-lived.

363. Session Factory provides a second level cache and Session provides a first level

cache.

364. Lazy Loading in hibernate.

**Lazy loading is a fetching technique used for all the entities in Hibernate. It**

**decides whether to load a child class object while loading the parent class object.**

**365.** Caching in hibernate.

Caching is a mechanism to enhance the performance of a system. It is a buffer memory

that lies between the application and the database. Cache memory stores recently used

data items in order to reduce the number of database hits as much as possible.

**366. How to create immutable class in hibernate.**

**Avoid providing any methods which modify object state.** Obvious candidates are

property setters as well as any other methods adjusting existing properties.

367. **Make all fields private** – to avoid modifying them directly, especially if they are

reference variables.

368. **Make all fields final** – to explicitly express intent that their values should not change.

This also means all the properties need to be assigned at the moment of creation in

constructor.

369. **Ensure class cannot be extended** – this eliminates a possibility to expose its

variables indirectly through a child class. It can be accomplished by making the class

final or by providing a private constructor. Static factory method or a builder class is used

to instantiate objects in such a scenario.

**Hibernate Architecture.**

370. The Hibernate architecture is categorized in four layers.

371. Java application layer

372. Hibernate framework layer

12

373. Backhand api layer

374. Database layer

375. IOC Container.

376. The IoC container is responsible to instantiate, configure and assemble the objects.

377. There are two types of IoC containers. They are:

**Bean Factory Application Context**

378. Get () Vs load().

**Get ( ) Load ()**

It is used to fetch data from the

database for the given identifier

It is also used to fetch data from the

database for the given identifier

If object not found for the given

identifier then it will return null object

It will throw object not found exception

It returns fully initialized object so this

method eager load the object

It always returns proxy object so this method

is lazy load the object

It is slower than load() because it return

fully initialized object which impact the

performance of the application

It is slightly faster.

If you are not sure that object exist then

use get() method

If you are sure that object exist then use

load() method

379. Hibernate vs JDBC. Why Hibernate over JDBC?

380. Lazy Loding vs Eager Loading.

**Lazy Eager**

In Lazy loading, associated data

loads only when we explicitly call

getter or size method.

Eager loading, data loading happens at the

time of their parent is fetched

Many-To-Many and One-To-Many

associations used lazy loading

strategy by default.

Many-To-One and One-To-One associations

used eager loading strategy by default.

It can be enabled by using the

annotation parameter :

fetch = Fetch Type. LAZY

It can be enabled by using the annotation

parameter :

fetch = Fetch Type. EAGER

Initial load time much smaller than

Eager loading

Loading too much unnecessary data might

impact performance

381. States of Object in Hibernate.

**Transient State**

**An object we haven't attached to any** *session* **is in the transient state.** Since it was

never persisted, it doesn't have any representation in the database. Because no

*session* is aware of it, it won't be saved automatically.

**Persistamce State-**

An object that we've associated with a ***session*** is in the persistent state. **We either**

**saved it or read it from a persistence context, so it represents some row in the**

**database.**

**Detached State-**

When we close the *session*, all objects inside it become detached. Although they still

represent rows in the database, they're no longer managed by any *session*.

382. First Level Cache Vs Second Level Cache.

**OTHER QUESTIONS**

383. Is it possible to change default port number.

384. Microservices And Webservices .

385. How to integrate email API in project.

386. Which all API’S you used in your project.

387. Why would you prefer STS on other applications.

**What is SOLID principle with advantages?**

35. Difference between micro services and monolithic applications?

In monolithic application all module based on one server. In micro services bigger application break

into smaller mini project and interact by web services

36. What is Maven?

Maven is a build tool which is based on the concept of a project object model (POM),. Using maven,

we can build and manage any Java-based project.

37. What is Spring Initializer?

Spring initializer gives spring boot project structure which we can further use to make spring boot

project in eclipse.

38. What is Unit testing?

It is also known as White Box testing where testing done by code level (programming level). Ex.:-

jUnit , testing.

39. What is Junit?

Junit used for unit testing. Annotation used in junit are:- @Test, @Before, @After, @BeforeTest,

@AfterTest

40. What is JPA?

Java persistence API(JPA) is a concept of taking the object content and map it into the Database by

ORM (object relationship mapping).

41. What is Hibernate JPA?

The implementation of JPA can be done by using Hibernate.

42. What are Spring IOC and Its type?

Inside Spring IOC(inversion of control) dependency injection logic written inside it. There are 2 types

of Ioc available

I)BeanFactory ,

II)ApplicationContext

43. What is spring security?

It is a framework that focuses on providing authentication and authorization mechanism to spring

application.

44. What are JPA annotations used in your project explain each ?

JPA annotation are:-

**(i)@Entity :-**It defines which java class is map to DB table.

**(ii)@Table :-**When DB table and Entity class name isn’t same we use this annotation for

mapping.

**(iii)@Column:**- This is used to map entity class variable into DB column **(iv)@Id :**- It maps entity

class variable with primary key column of DB.

**(v)@Generated Value:**- It helps us to auto-increment value of variable in entity class while saving

record in DB.

45. What are spring boot annotation used in your project explain each?

46. Spring boot annotation are:-

@Autowired, @controller, @service , @component, @RequestMappin,

DeleteMapping, @pathvariable, @modelAtteribute, @RequestBody

@RequestParam, @RestController ,@GetMapping ,@PostMapping,,@PutMapping,

,,@Qualifier, @springBootApplication

47. What is Dependency injection?

48. It is core of spring framework. Used to inject bean inside Given reference variable

Types of Autowired and explain each?

Default By name By type constructor

49. Difference between @Controller and @RestController?

@controller define controller layer in spring boot , @Restcontroller defines webservices layer in our

project.

50. Difference between @Autowiring and @Bean?

When interface implement only one class @Autowiring used When interface implement more than

one class @Bean used.

51. What is @Qualifier and @Services and @Component and @Value and @Query ?

@Qualifier :- if one interface implement into 2 different classes ,so we’ve to tell spring which

class object should create.

@Service :- Define the service layer of spring boot. @component :- It tells spring boot to maintain

object life cycle.

@Query :- It use to declare query directly on repository method.

52. Explain @SpringBootApplication?

It defines the starting point of execution in spring boot project.

53. Difference between BeanFactory and ApplicationContext?

The applicationContext comes with advanced features,while beanFactory comes with basic features.

Application context implement from bean factory interface.

54. Difference between JPA and Hibernate?

JPA is a concept where we taking object content and map it into the DB and implementation done by

hibernate.

55. Explain advantage of Hibernate over JDBC?

Jdbc is a connectivity ,hibernate is a framework

Jdbc has lower performance than hibernate We can create database table by using hibernate

56. What is Session in hibernate?

Session interface(API) is the main tool used to communicate with hibernate It used to get a physical

connection with Database.

57. What are roll back and commit?

Once rollback is execute ,the database would reach its previous state. Once the commit statement

execute ,the data cannot be rolled back

58. Session management in spring boot?

1. create spring boot project from spring initializer.

2. add spring session jdbc dependency in pom.xml

3. add spring jdbc properties in application

59. How to create session object?

HttpSession mySession=request.getSession( );

60. Difference between session and sessionFactory?

Session factory is a factory class for session objects.it available for whole application while session

only available for particular transaction.

61. Difference between get and load method of hibernate?

Both are used to fetch data .

Get( ) return null if no row is available on the database

Load( ) throws object not found exception.

62. What is caching and hibernate and explain 1st level and 2nd level caching?

Caching is the mechanism to enhance the performance of the system.

1st level maintained at session level and accessible only to the session, while 2nd level maintain at

the session factory level and available to all sessions.

63. What are Core interface of hibernate?

Configuration interface

Transaction interface

Session interface – Session Factory interface

**What is Jenkins?** Jenkins is used for testing and reporting.

64. What is JIRA? Jira is a software application used for issue tracking and project

management.

65. How to read application. Properties in any class?

Read a property from application. Properties file using @ Configuration Properties & @value

66. What are response codes of POSTMAN?

**200/201/204/400/401/403/404/405/500/503**

a. 200= ok

b. 201=file created

c. 400= bad request

d. 404=file not found

e. 500=internal server error

f. 503=service unavailable

g. 203=Non- authorized

h. 501=HTTP server error

Explain Session Tracking System

Session Tracking is a way to maintain state (data) of an user.

67. Explain types of spring bean scope?

Singleton , prototype ,request ,session, global session

68. What is Actuator in spring boot?

Actuator is used to expose operational information .It is the subproject of spring boot.

69. What is Starter tag in spring boot?

It is the default configuration of all hibernate dependency.

70. Types of hibernate mapping explain each?

Primitive type mapping (integer ,long , short ,Boolean ) Date and time mapping (Date , time

,timestamp ,calendar

Binary and Large Object Types(binary ,text ) Jdk related type (class,timezone)

71. Types of inheritance in hibernate?

1. Mappings:Table per class hierarchy

2. Table per sub-class hierarchy

3. Table per concrete class hierarchy

**Advantages of Hibernate over JDBC:**

104. Hibernate is an ORM tool

105. Hibernate is an open source framework.

106. Better than JBDC.

107. Hibernate supports inheritance and polymorphism.

108. With hibernate we can manage the data stored across multiple tables, by applying

relations(association)

Hibernate has its own query language called Hibernate Query Language. With this HQL

hibernate became database independent.

Hibernate supports relationships like One-To-One, One-To-Many, Many-To-One ,Many-

To-Many.

Hibernate supports lot of databases.

. Hibernate supported databases List.

Hibernate also supports annotations along with XML.

. Hibernate supports Lazy loading.

. Hibernate is easy to learn it is developers friendly.

116. Hibernate maintains database connection pool.

. Using Hibernate its Easy to maintain and it will increases productivity

118. What is JPA can you directly implement it?

**JSP Interview Questions**

119. JSP lifecycle?

120. What are the JSP implicit objects?

121. What are the different tags in JSTL?

The JSTL tags can be classified, according to their functions, into the following JSTL tag library

groups that can be used when creating a JSP page −

**Core Tags Formatting tags SQL tags XML tags JSTL Functions**

122. Explain the **jspDestroy()** method.

Methods of Hibernate?

Methods of Hibernate Session

a) **Save():** Save() method generates the primary key and inserts the record in the

database. It is similar to the persist() method in JPA but it behaves differently in a

detached instance by creating the duplicate record upon database commit.

b) **Update():** Update() is used to update the existing database record. It returns an

exception if the record is not found or called in a transient instance.

c) **saveOrUpdate():** It saves or updates the database based on the entity passed. It

does not return an exception in the transient state but it makes the state to persistent

during a database operation.

d) **merge():** Values from a detached entity are updated to the database when the

merge() is used by changing the detached entity to the persistent state.

e) **delete():** Delete method works in persistent mode to remove the entity from the

database. An exception is returned if no record is found in the database.

163. Xml v/s JSON?

What is spring boot starter?

**Spring Boot** provides a number of **starters** that allow us to add jars in the classpath.

Spring Boot built-in **starters** make development easier and rapid. **Spring**

**Boot Starters** are the **dependency descriptors**.

In the Spring Boot Framework, all the starters follow a similar naming pattern: **spring-bootstarter-**

**\***, where **\*** denotes a particular type of application. For example, if we want to use Spring and

JPA for database access, we need to include the **spring-boot-starter-data-jpa**

dependency in our **pom.xml** file of the project.

174. Spring Boot Actuator Features

175. There are **three** main features of Spring Boot Actuator:

**Endpoints**

**Metrics**

**Audit**

Session, session factory?

**SessionFactory** is an interface. SessionFactory can be created by providing

Configuration object, which will contain all DB related property details pulled from either

hibernate.cfg.xml file or hibernate.properties file. SessionFactory is a factory for Session

objects.

We can create one SessionFactory implementation per database in any application. If

your application is referring to multiple databases, then you need to create one

SessionFactory per database.

The SessionFactory is a heavyweight object; it is usually created during application

start up and kept for later use. The SessionFactory is a thread safe object and used by

all the threads of an application.

**A Session** is used to get a physical connection with a database. The Session object is

lightweight and designed to be instantiated each time an interaction is needed with the

database. Persistent objects are saved and retrieved through a Session object.

The session objects should not be kept open for a long time because they are not

usually thread safe and they should be created and destroyed them as needed. The

main function of the Session is to offer, create, read, and delete operations for instances

of mapped entity classes.

181. Spring MVC?

A Spring MVC is a Java framework which is used to build web applications. It follows

the Model- View-Controller design pattern. It implements all the basic features of a

core spring framework like Inversion of Control, Dependency Injection.

A Spring MVC provides an elegant solution to use MVC in spring framework

by the help of **DispatcherServlet**. Here, **DispatcherServlet** is a class that receives

the incoming request and maps it to the right resource such as controllers, models,

and views.

182. JPA anotations?

@Entity, @Table ,@Column ,@Id ,@GeneratedValue.

183. Stream API with example?

184. Microservices v/s Monolithic architecture?

244. Benefits of Spring Boot -----Transient keyword?

**Advance Java & All**

245. Difference between JPA & Hibernate

246. What is singleton?

247. What is spring security?

248. All JPA annotation & spring boot Annotation explain?

249. Spring boot application (IMP)

250. What is session?

251. What is session factory?

252. What is spring IOC? Types

253. Explain DI?

254. Difference between primary key, foreign key & unique key.

255. What is served & its life cycle?

256. MVC Architecture?

257. Web services?

258. What is boxing?

259. What if two null store in HashMap?

260. Why we use spring boot framework?

261. How to integrate hibernate in your project?

262. Mapping in hibernate?

263. How to configure database in spring boot project?

264. Diff between @controller and @restcontroller ?

265. What are the types of Autowire?

266. Explain Dependency Injection?

267. What is spring IOC ?

268. Spring boot project flow?

269. What are the HTTP methods and explain each one?

**SPRING BOOT->**

289. Spring Vs Spring Boot (Advantages, Disadvantages, Which one is better).

290. Spring Initializer.

291. Spring Acuator.

292. Spring Boot starter.

293. Spring MVC.

294. Flow of Spring MVC.

295. Advantages of Spring MVC over other framework.

296. Declaration of different layer in Spring Boot.

297. How to connect Spring Boot application to database using JDBC.

Design patterns (singleton, prototype)

How to achieve singleton pattern if we can break them using clone and reference

What is Junit testing

How to test controller service layer and data base

What is mocking

Write test case in Mockito , mock MVC, integrity testing

Explain REST API

298. How to implement REST API

299. How to explore JSON object

300. Difference between @RestController and @Controller

301. @SpringBootRun explain

302. Difference between @Configuration and @Autoconfiguration

303. @Component and @Service Difference

304. @Mock annotataion explain

305. @Qualifier annotation

306. How to consume query parameter

307. Classes used to consume JSON object

308. Why to go with Microservices Architecture

309. Stream API explain in brief

312. Predicate, function, consumer, supplier syntax, parameter and return type for

these

313. Functional interface, predefined fuctional interface names.

**HIBERNATE->**

365. What is ORM ?

366. Hibernate Vs JDBC.

367. Session and SessionFactory in hibernate.

368. Caching in hibernate (Indepth).

369. How to create immutable class in hibernate.

370. Hibernate Architecture.

371. IOC Container.

**OTHER QUESTIONS ->**

372. Is it possible to change default port number.

373. Microservices And Webservices (All the questions which is provided by Sir).

374. How to integrate email API in project.

375. Which all API’S you used in your project.

376. Why would you prefer STS on other applications.

377. NOTE-> (Project explanation is very important)

Explain about spring , spring bean , Spring bean scope ?

397. Code snippet for string object ?

398. Actuators ?

399. Stators in spring boot ?

400. Design patterns : Singleton , prototype

401. Java bi predicate

402. API flow , Rest API

403. JBOSS architecture

404. Stream methods

405. Copy constructor

406. Types of constructor

407. Executor service

408. Call procedure from .java

409. J unit

410. Login framework

411. SOAP vs REST

412. Expose and consume of web services

413. Hibernate JPA vs JPA repository

414. Prepared statements vs statements

415. JDBC template

416. What is SDLC ?

417. What are present in object class ?

418. Spring security

419. Email API

420. Parallel stream

421. Types of design pattern ?

422. How to optimize code ?

423. What actuator in spring ?

424. Bean factory and application context ?

425. How to exclude in spring boot ?

426. Types of inheritance in hibernate ?

427. Types of bean scope

428. Types of Auto wiring

429. Key components of spring boot

Modes of auto wired

432. Spring security

433. Spring IOC

434. Streams in java

435. Aggregation composition

436. Fail fast , fail safe

437. @component

438. What is server validation and Clint validation

HTTP methods in spring boot ?

442. Agile methodology ?

443. Difference between monolithic architecture and micro service ?

444. OAUTH2, pact tool ?

445. Semantic monitory

446. Clint certificate

447. What you will do if micro service fail ?

448. Hibernate architecture

449. Spring initializer

450. Spring security

451. Difference between bean factory and application context

452. Different springBoot scopes

453. Different hibernate objects

454. Difference between mapping and configuration classes in hibernate ?

455. Load balancing

456. Circuit breaker

457. Jeera

458. GIT HUB repository

459. Front controller

460. @qualifier , @path variable

461. Count number of objects in java ?

462. Functional interface sub methods and types

463. How will you pass xml in application Properties ?

464. What is postman ?

465. Repository layer and how it will work ?

466. Where we store java code ?

467. Jenkins

468. Difference between session and session factory ?

469. Replace and replace all methods ?

470. How to read application properties file ?

471. Trust full API ?

472. How to copy one hash map to other ?

473. Address of last data in linked list ?

474. Which control used in project ?

475. What is DDL , DML

476. Difference between Web server and Application server

477. How to test Web service in postman ?

478. Can multiple finally blocks execute ?

479. How to connect two tables in spring ? 155)

480. ADVANCE JAVA QUESTIONS

481. Explain the life cycle of Servlet , JSP ?

482. What is servlet , jsp , html pages ?

483. What is Inter Servlet connectivity ? ( ISC )

484. What are session variables ?

485. Explain about JSP tags ?

486. Explain about MVC architecture ?

487. Write a logic for JDBS connection ?

488. Explain about CSS and Java Script ?

489. Explain about JSTL tags ?

490. Different types of creating Sessions ?

491. Explain about request dispatcher ?

492. Write SQL queries ?

493. What are data types of SQL ?

494. What are constrains ?

495. Explain about joins ?

496. Difference between application server and Web server ?

497. Explain about J unit ?

498. How to create Spring boot project ? ( maven )

499. Explain about application. Properties , repository layer , Controller , view ,

service layers ?

500. Explain about Entity classes ? ( JPA , POJO )

501. What are the annotations used in springBoot . Explain each ?

502. Explain about Session factory and logic to implement it in project ?

503. Explain about Hibernate and advantages over JDBC ?

504. Explain about hibernate configuration and components of its object ?

505. Explain the steps to develop a web application using hibernate?

506. What are core interfaces of hibernate ?

507. What are the advantages of ORM over JDBC ?

508. Explain about spring ?

509. Difference between spring and spring boot ?

510. Explain about spring bean ?

511. Explain about web services ?

512. Difference between REST and SOAP ?

513. Why we use xml files over JSON ?

514. Explain about micro services ?

515. Explain about POSTMAN ?

516. What are the errors in postman . Explain about each ?

517. Explain the methods of hibernate ?

518. Explain HTTP methods ? ( get , put , patch , delete , post )

519. What are hibernate annotations . Explain each ?

520. Difference between hibernate with annotations and hbm.xml files ?

521. Explain about **Agile** completely ?

522. What is dispatcher servlet , how it works ?

523. How will generate pdf in all the view ?

524. View resolver , Jack Son

525. How will you redirect to view ?

526. What are object states in hibernate ?

527. Explain dilate hibernate properties ?

528. Explain about level 1 , level 2 cache ?

529. Explain about Session tracking system ?

530. Difference between save and persist method in hibernate ?

531. What are @primary and @qualifier annotations ?

532. How to monitor more than one micro services ?

533. What are roll back and commit ?

534. How to handle exceptions in spring boot ?

535. How to authenticate application ?

536. @springBoot @cross origin @bean

537. Features of html and CSS ?

538. Difference between do get , do post ?

539. Types of dependency injection ?

540. @Component , @controller

541. @mapping annotations , @qualifier , @primary , @response body

542. Life cycle of spring bean ?

543. What is API ?

544. How expose and consume are done in applications ?

545. Entity class

546. ORM

547. Collection

548. How to store a student object into collection

549. How to retrieve a record from the student objects based on your name

550. Instance variables