**OPEN POINTS**

**Name : S.R.Lakshmi**

**Employee id:761154**

**Batch: End to End**

**1).PURPOSE OF JENKINS**

* Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose.
* It is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.
* It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.
* With Jenkins, organizations can accelerate the software development process through automation.
* Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis and much more.

**2).Purpose of JWT**

Jwt are used to prove that the sent data actually created by an authentic source.

**3).Difference between TypeScript and JavaScript**

|  |  |
| --- | --- |
| **TypeScript** | **JavaScript** |
| TypesScript is known as Object oriented programming language | JavaScript is a scripting language |
| TypeScript has a feature known as Static typing | JavaScript does not have this feature |
| TypeScript gives support for modules | JavaScript does not support modules |
| TypeScript has Interface | JavaScript does not have Interface |
| TypeScript support optional parameter function | JavaScript does not support optional parameter function |

**4).Frequently used starters in Spring Boot**

* spring-boot-starter
* spring-boot-starter-web
* spring-boot-starter-security
* spring-boot-starter-jdbc
* spring-boot-starter-data-jpa

**5).Difference between observable and promise**

|  |  |
| --- | --- |
| **Observable** | **Promise** |
| An observable is like a st and allows to pass zero or more events where the callback is called for each event. | A Promise handles a single event when an async operation completes or fails. |
| allows to cancel the subscription | doesnot to cancel the subscription |

**6).Advantages of Hibernate**

Hibernate ORM easily solves the data mismatch found between the object oriented classes of an application and relational database.

Hibernate is database independent

Hibernate supports a powerful query language called HQL (Hibernate Query Language).

The lazy-loading concept fetches only the necessary object that is required for the execution of an application.

Hibernate is highly scalable.

**7).Difference between NoSQL and SQL Database**

|  |  |
| --- | --- |
| **SQL** | **NoSQL** |
| SQL Database is a Relational Database and a structured one | NoSQL is a Non-relational database |
| SQL Databases have a well-designed pre-defined schema | NoSQL databases have a dynamic schema for document type or unstructured data |
| SQL databases are vertically scalable | NoSQL databases are horizontally scalable |

**8). Methods in Jpa Repository that are not in CRUD Repository**

findAll(Sort sort);

findAll(Pageable pageable);

saveAndFlush(S entity);

flush()

**9).Spring bean life cycle methods**

**Initialization callbacks** - afterPropertiesSet()

**XML-based configuration**

use init-method attribute

<bean id = "exampleBean" class = "examples.ExampleBean" init-method = "init"/>

Following is the class definition

public class ExampleBean {

public void init() {

// do some initialization work

}}

**Destory callbacks** - destroy()

**XML-based configuration**

use destroy-method attribute

<bean id = "exampleBean" class = "examples.ExampleBean" destroy-method = "destroy"/>

Following is the class definition −

public class ExampleBean {

public void destroy() {

// do some destruction work

}}

**Spring bean life cycle annotations**

* @PostConstruct
* @PreDestroy

**10). Different Http Request methods in JMeter**

1. GET
2. POST
3. HEAD
4. PUT
5. OPTIONS
6. TRACE
7. DELETE
8. PATCH
9. PROPFIND
10. PROPPATCH
11. MKCOL
12. COPY
13. MOVE
14. LOCK
15. UNLOCK
16. REPORT
17. MKCALENDER
18. SEARCH

**11). Attributes used in @Query**

@Query(

value = "SELECT \* FROM Users ORDER BY id \n",

countQuery = "SELECT count(\*) FROM Users",

nativeQuery = true)

**Attributes**

1. value
2. countQuery
3. nativeQuery

**12.Spring boot annotations for validation**

* @NotBlank
* @Min
* @Max
* @AssertTrue
* @size
* @Email