**1st Round**

**Java and Spring: Focus on Java 8 concepts**

-- Explain Oops concepts

-- Difference between functional programming and Object-oriented programming language.

-- Explain Java 8 features you know - <https://mkyong.com/tutorials/java-8-tutorials/>

-- What are lambda expressions

--Write a program to sort list of employees your Java 8 streams, How do you sort in reverse order.

--Difference between ConcurrentHashMap vs HashMap

--Design pattern you know? Follow up question, Explain Decorator Design patterns. Refer: <https://stackoverflow.com/questions/1673841/examples-of-gof-design-patterns-in-javas-core-libraries>

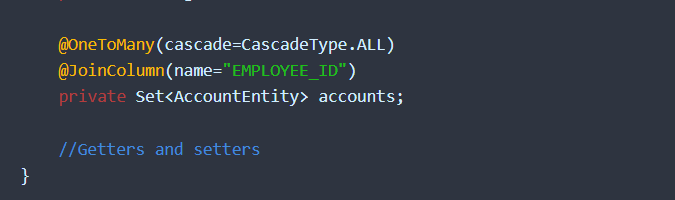
--In what order the files application.properties and yml loaded in spring boot.

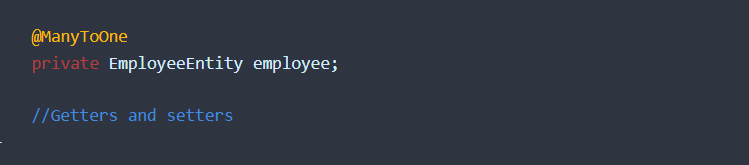
-write a method to get employee by name using JPA. Refer: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#jpa.query-methods.query-creation>

--Explain Transaction management - Isolation levels and Propagation levels

--Using @Transaction at class level vs method level

--Hibernate OneToMany and ManyToOne mappings





--JPA CascadeType and FetchType - <https://docs.oracle.com/cd/E19798-01/821-1841/bnbqm/index.html>

| **Cascade Operation** | **Description** |
| --- | --- |
| ALL | All cascade operations will be applied to the parent entity’s related entity. All is equivalent to specifying cascade={DETACH, MERGE, PERSIST, REFRESH, REMOVE} |
| DETACH | If the parent entity is detached from the persistence context, the related entity will also be detached. |
| MERGE | If the parent entity is merged into the persistence context, the related entity will also be merged. |
| PERSIST | If the parent entity is persisted into the persistence context, the related entity will also be persisted. |
| REFRESH | If the parent entity is refreshed in the current persistence context, the related entity will also be refreshed. |
| REMOVE | If the parent entity is removed from the current persistence context, the related entity will also be removed. |



--@RequestParam vs @Path Variable

@RequestParam and @PathVariable can both be used to extract values from the request URI, but they are a bit different.

While ***@RequestParam*s extract values from the query string, *@PathVariable*s extract values from the URI path**:

@GetMapping("/foos/{id}")

@ResponseBody

**public** String **getFooById**(@PathVariable String id) {

**return** "ID: " + id;

}

Then we can map based on the path:

http://localhost:8080/spring-mvc-basics/foos/abc

----

ID: abc

And for *@RequestParam*, it will be:

@GetMapping("/foos")

@ResponseBody

**public** String **getFooByIdUsingQueryParam**(@RequestParam String id) {

**return** "ID: " + id;

}

which would give us the same response, just a different URI:

http://localhost:8080/spring-mvc-basics/foos?id=abc

----

ID: abc

--PUT vs POST

--@RestConstroller vs @Controller

--@Component vs @Controller vs @Service @Repository

-- Refer: <https://martinfowler.com/articles/richardsonMaturityModel.html>

--How spring boot works and advantages of spring boot over spring.

--can you setup Jenkins and fix any error

--What is 12 factor app

--sonar - what is the max number of parameters to any method that raises violation in sonar

--Spring security - How to you manage Authorization with jwt token

**Angular**

- Types of communication in Angular, how sibling components communicate or share the data.

- Types of storage in Angular

<https://medium.com/@nixonaugustine5/localstorage-and-sessionstorage-in-angular-app-65cda19283a0>

- What is the version of Angular you worked on and what are the features added in that version.

-What is SPA

-Explain Angular Routing

Routing in AngularJS is used when the user wants to navigate to different pages in an application but still wants it to be a single page application. AngularJS routes enable the user to create different URLs for different content in an application. The **ngRoute** module helps in accessing different pages of an application without reloading the entire application.

**Important:**

* $routeProvider is used to configure the routes. It helps to define what page to display when a user clicks a link. It accepts either when() or otherwise() method.
* The ngRoute must be added as a dependency in the application module:

//const app = angular.module("myApp", ["ngRoute"]);

**2nd Round**

--Difference between JPA and Hibernate

--what is the difference between jdbc and Datasource

--How do you configure Hibernate with JPA

--Difference between parallel streams vs executor services and when to use what

--Find names of employees whose salaries between 10000 and 20000 using Java 8

--Java 8, what is the difference between intermediate operations vs terminal operations.