Controllers-6 tables-8

explain flow of your project?

when we run both front end and backend, then we see home page.In home page there are sectors. This home page is display because in react we have index.js in that we have render app component.

which annotation used in project?

@springBootApplication(@EnableAutoConfiguaration @ComponentScan @configurration)

Many Spring Boot developers like their apps to use auto-configuration, component scan and be able to define extra configuration on their "application class". A single @SpringBootApplication annotation can be used to enable those three features, that is:

@EnableAutoConfiguration: enable Spring Boot’s auto-configuration mechanism

Spring Boot auto-configuration attempts to automatically configure your Spring application based on the jar dependencies that you have added. For example, if HSQLDB is on your classpath, and you have not manually configured any database connection beans, then Spring Boot auto-configures an in-memory database.

@ComponentScan: enable @Component scan on the package where the application is located (see the best practices)

@Configuration: allow to register extra beans in the context or import additional configuration classes

@Repository Annotation is a specialization of @Component annotation which is used to indicate that the class provides the mechanism for storage, retrieval, update, delete and search operation on objects

@Service One most important thing about the @Service Annotation is it can be applied only to classes .It is used to mark the class as a service provider. So overall @Service annotation is used with classes that provide some business functionalities

@RestController is used for making restful web services with the help of the @RestController annotation. This annotation is used at the class level and allows the class to handle the requests made by the client.

**@RestController= @Controller + @ResponseBody**

@RequestBody : Annotation indicating a method parameter should be bound to the body of the HTTP request.

@ResponseBody annotation can be put on a method and indicates that the return type should be written straight to the HTTP response body

@CrossOrigin

enables cross-origin resource

allows a server to make cross-domain calls from the specified domains,

@Autowired is use for automatic dependency injection.

Dependency injection is basically providing the objects that an object needs (its dependencies) instead of having it construct them itself.

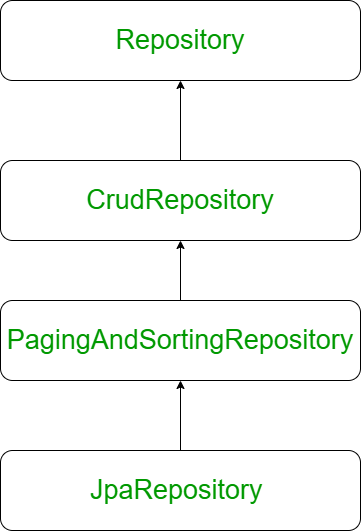
@Component is an annotation that allows Spring to automatically detect our custom beans.

*@Query* it provides us with the opportunity to write a specific JPQL or SQL query in the *@Query* annotation

**The**[*@Modifying* annotation](https://docs.spring.io/spring-data/jpa/docs/current/api/org/springframework/data/jpa/repository/Modifying.html)is used to enhance the *@Query* annotation so that we can execute not only *SELECT* queries, but also *INSERT*, *UPDATE*, *DELETE*, and even *DDL* queries**.**

**Default available methods of JPA**

CrudRepository and PagingAndSortingRepository offer default methods such as: findAll, findAllById, findById, deleteAll, deleteById, save, saveAll.



DEPENDENCIES

<artifactId>spring-boot-starter-web</artifactId>

Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container

<artifactId>spring-boot-starter-data-jpa</artifactId>

Starter for using Spring Data JPA with Hibernate

<artifactId>spring-boot-maven-plugin</artifactId>

The Spring Boot Maven Plugin provides Spring Boot support in Apache Maven. It allows you to package executable jar or war archives, run Spring Boot applications

<artifactId>mysql-connector-java</artifactId>

It provides JDBC Type 4 drive. This driver supports auto-registration with the Driver Manager.

JDBC

java database Connectivity

using jdbc, java application can connect to any database such as MySQL,Oracle,SQLServer etc.

jdbc provides two things:

a) jdbc API [classes and interfaces] which remains same no matter which database you use.

b) jdbc drivers

driver is a software to connect to database

There are 4 types of drivers in jdbc:

Type 1 - which requires ODBC to be installed on a machine. It has almost become obsolete.

Type 2 - which requires database client library to be installed on client machine. cannot be used for internet purpose.

Type 3 - which requires middleware (Application server) to get connected to database.

Type 4 - which is the fastest driver as it directly connects to the database.

Steps required for jdbc application:

a) load and register the driver with DriverManager.

b) get connection with database

c) communicate with database with the help of JDBC API.