**Documentation for JAVA Spring Boot Application.**

**WorkFlow Diagram:-**

Repository Interface

DB

Model

Client

Controller

Service

***Annotations.***

* @SpringBootApplication is the Combination of @Configuration, @EnableAutoConfiguration, @ComponentScan.
* @RestController is the combination of @Controller and @RequestBody.
* @Controller will define the Controller Class.
* @Service will define the Service Class.
* @Repository will define the Repository Interface with extends of Crud Repository.
* @Entity will refer the class are table, POJO, model.
* @Id will notify primary key of the table.
* @Table will say the name of the table for corresponding POJO class.
* @Autowired will provide the bean.
* @RequestMapping is the endpoint.
* @ResponseBody will facilitate the response in JSON.
* @GetMapping is similar to @RequestMapping. The only difference is @GetMapping used for Method Level and @RequestMapping for Class Level.
* @ComponentScan is used to specify the Service Package for Corresponding Controller Class.
* @EntityScan is used to specify the Model Package for Corresponding Controller Class.
* @EnableJpaRepositories is used to specify the Repository for Corresponding Controller Class.
* **Controller Package.**
* **Service Package.**
* **Repository Package.**
* **Model Package.**

***Controller Package.***

The Controller is a class-level. It is a specialization of **@Component**. It marks a class as a web request handler. It is often used to serve web pages. By default, it returns a string that indicates which route to redirect. It is mostly used with **@RequestMapping** annotation.

***Service Package.***

 It is also used at class level. It tells the Spring that class contains the **business logic**.

***Repository Package.***

It is a class-level annotation. The repository is a **DAOs** (Data Access Object) that access the database directly. The repository does all the operations related to the database.

***Model Package.***

It is a class-level. In this class we can create a Pojo class with getter, setter & constructor.

***Example:***

Step 1: when user hit the Url Localhost:8085/addStudent it will reach the Controller Class has @RestController , @Autowired of Service class and the Method is

@Autowired

StudentService studentService;

@RequestMapping(Value=”/addStudent”,Method=RequestMethod.POST)

Public void addStudent(@ResquestBody Student student){

studentService.addStu(student);

}

Step 2: The Method will go to StudentService Class and it has @Autowired of StudentRepository interface.

@Autowired

StudentRepository studentRepo;

Public void addStu(Student student){

studentRepo.save(student);

}

Note: Save is Default querytemplate in StudentRepo

Step 3: The Method will go to StudentRepository Interface with extends of CRUD Repository and it configured @Repository which means Spring Boot provide 19 basic QueryTemplate.

@Repository

Public interface StudentRepository extends CrudRepository<Student, String>{

}

Note: Student is the POJO class & Model Class has @Entity and @Table Annotation. Behind the Screen using hibernate the action is preformed.

POJO CLASS:

@Entity

@Table(name=”student”) # name of the table in database

Public class Student{

@Id #Primary Key in the student database table

int rollno;

String name;

int age;

// with Getter and Setter

// constructor with fields

//empty Constructor

}

Step 4: Using Model Class required querytemplate will execute and add student value in DataBase table(student).