SPRING BOOT BASIC CONCEPT AND UNDERSTANDING:

PROCESS FLOW:

Browser(client) -->request -->Springboot (REST API)-->database(server)-->fetch data(response)-->Browser(client)

---

front-end app -->fires a request --> and waits for response

HTTP- req,resp (understandable Language)

REST api architecture (springboot/java backend)--API (interface)

HTTP Methods:

-->GET -> to fetch data

-->POST -> to insert data

-->PUT -> to update data (replace)

-->DELETE -> to delete data

Why Springboot:

-->Java -->Spring framework -->Spring mvc (Model,view,controller)

-->Spring boot (Easy, wide & Advance features like.,:Auto Configuration)

How browser will make request calls?

--> browser --> request --> Users/userId/ email?? : Through API endpoint url (Ex: http://localhost:8080/userEmail?userId=100)

How springboot/backend catches the request:

-->In RESTController class: Main class in springboot implementation.

How to Create Springboot project:

--> go to SpringInitializr.io --> select Maven --> add starter web dependency

---

Annotations:

what are annotations? --> giving commands to springboot.

public class user{

} --> controller ni cheyyi ani --> springboot ki ela cheppali?? @RestController annotation vaadi cheppali

@RestController //request vachina dhaggara nunchi, backend work em cheyyali, ela cheyyali ane dhanni CONTROL chestundi.

public class userController{

} --> ee annotation valla spring ki artham ayyi , okay -> ee class ni CONTROLLER class ga chestundi

---

Annotations examples: @RestController - to make normal java class as a controller class.

@Entity - Oka data object/Table (neeku oka table loki data insertion,deletion,creation,updation (CRUD operations))

@Entity

@Table(name = "user_details")// appudu java/springboot artha ayyi, ee particular table dhaggariki velli mana operations execute chestundi.

public class User{

  @Id //ee particual parameter user_details aney table lo PRIMARY KEY laga undi ani JAVA ki cheppey Annotation

  private int id;

  @Column(name = "f_name") //column name update chesi search cheyyataniki

  private String name;

  private String email;

  //generate getters, setters, constructors

}

Database: (for example)

create table user_details (id int PRIMARY KEY, f_name varchar(255), email varchar(255));

insert into user_details (id, f_name, email) values (1,"Pavan","123@gmail.com");

MAIN CLASSES in SPRINGBOOT:

1.)Entity Class --> Uses annotation (@Entity)

2.)Controller class --> Uses Annotation (@RestController) Usage: To fetch/controll the REQUESTS (Handling requests)

3.)Service Class --> Uses annotation (@Service) Usage: Controller class teeskunna requests ni --> observe chesi --> asalu ah request enti--> IMPLEMENTATIONS chestundi

--> REQUESTS IMPLEMENTATIONS

4.)Repository Class --> Uses annotation (@Repository) Usag: DATABASE CONNECTIVITY or TO perform DATABASE/DATA RELATED OPERATIONS.

--> Main ga query no need to write, In-built methods of @Repository will handle everything

--> but, if there are no such methods with our requirements, In that case, we can write query.

DEPENDENCIES: (pom.xml)

--> Java/springboot ki konni konni panulu cheyyataniki konni konni vaati meedha depend avutundi, avey dependencies.

--> springboot project configuration lo "POM.xml" file lo java ki/springboot ki related anni dependecies untay.

for ex:- SpringIntializr.io --> Add Dependencies lo basic dependencies like

- Starter web dependency, starter data-jpa dependency (for db configuration)
