Week 2

Introduction To Spring Boot Web Muc, Muc Architecture

REST APIS:

-> REST (Representational State Transfer) APIs (Application Programming Interfaces) are a Set of rules and Conventions for building and interacting with Web! Services who is igo

* GIET / Users ?- Retrieve to list of allousers.) spaid and for allow

* GET losers/fidy:- Retvieve a specific user by ID? i alcabo 18

* POST (Users: - Create donew Userotz (12) stops | airabo | 15 | ise

* PUT (Users/ Lidy: - Update an existing user by ID * PATCH/users/lidy:-Partially update an existing user by ID.

* DELETE [Users [Ridy !- Delete a User by ID: 10 1912) Abros 2 short with

Spring-boot-starter-web:

-> Spring - book - starter-web Contains the following dependencies

* Spring-boot - Starter

* Spring-boot-Starter | blinsory souson | blistod his alord in ign

* Spring - Core

* spring - mvc

* spring - book - starter - tomicat

Spring Boot Mvc Architecture:

Class Providing CRUD Operations

priso (89) spaidoui (re) ige

HHtp request

Dependency injection

elated tabe Wigo I'm

client → Controller > Service layer < object XML JSON

Can Boot Mile
1. Spring Boot Muc is a part of Spring Boot used to build web Applications using the Model-View-Controller (Muc) Pattern, using the Model-View-Controller (Muc), ul (View), and User input handling
using the hospiness logic (Model), UI (View), and User input handling
using the Model-View-Convoller (Model), UI (View), and User input handling 2. Helps Separate business logic (Model), UI (View), and User input handling
(controller).
(controller). 3. Spring Boot Mvc uses Dependency Injection to loosely Connect Components. 3. Spring Boot Mvc uses Dependency Injection to loosely Connect Components.
and close independent
I lolds data and some
Tala Is displayed (like
11-Alas veauests and connects to
Las Cada Clancer and Carrier
7 Since Components are independent, they can do be reused in other Projects.
7 Since Components are independent, mey can do so without loading 8. You can test individual parts (like Controllers (or) Services) without loading
. 1 10 DDN
The whole app. 9. As your Project grows, you can easily add new features without breaking
9. As your respect grows, applications in the profit of six is a six in the s
the old Ones the Aller projects because of the Clear Structure.
Layered Architecture
Lagered Aichitean)
1) Presentation Layer > * Ison Translation
The server and replace to the survey of the control
*Business Logic
(2) Rusinger Laurer + Volidation -> Dervice Classes
Business Layer > * Validation > Service Classes * Authorisation Hibeinet Col
*Business Logie *Business Logie Service Classes *Authorisation *Hiberinate Sel
(3) Pertaistance Laury - * Storage Lange (++)
(3) Pertaistance Laury - * Storage Lange (++)
(3) Perisistence Layer -> * Storage Logic (1) CRUD Operations are Pertormed. (4) Database -> * Actual Database -> Actual database (SQL)
(3) Perisistence Layer -> * Storage Logic
(3) Perisistence Layer -> * Storage Logic
(3) Perisistence Layer -> * Storage Logic

Spring Boot Web Project Structure

pallband togal (wind) 10 (what) pard serviced	Service By
11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	9/040092 2
client Controller Service respository	Actuality
Total	Database
everto partin tomition ottopondo plica Entity obragation es	· ,
TO TO THE FLISH EVALUATION STATE SSI	nes each clo

elipol resonizod ban odob eblotic 1. Client -> Controller (HTTP Request with ISON)

The client (like exweb browsen (ox) Postman) sends an HTTP request to the Controller in Ison format. this clear separation makes code cleaner

2. Controller -> Service (DTO) Transfer) barabal se star according to the Controller receives the requesty Converts of into a DTO (Data Transfer Object) and Passes it to <u>Service layer</u> for <u>Processing</u>.

3. Service -> Repository (Entity & Business Logic) 200010 7 19/089 4000 24.

The service layer contains business logic and works with Entity Objects. the Repository to interact with the database

4. Repository -> Database (CRUD Operations)

The repository layer Performs CRUD operations directly on the actual database 921V192 & using Entity clames.

5- Response Flow (Bakeck to client) Tolking to

The Processed data flows back in reverse - from repository - service - Controller

* Action Laborace - . . . Action dutabase (SGL)

-> client, Usually as a JSON response, and it is required to the second of the company of the second of the company of the second of the company of the comp

6/5/25

The Presentation Layer, DTO and Controllers:

Annotated Controllers:

- -> Spring Muc Provides an annotation-based Programming model where
- @Controller and @RestController Components use annotations to express request mappings, request input, exception handling, and move.
- → The @ Rest Controller annotation is a shorthand for @ Controller and
- @ Response Body, Meaning all methods in the Controller will return ISON/XML directly to the response body. (togat) possible appeals
- @ Rest Controller Vs @ Controller:
- 1. @ Controller Annotation is used to represent Classes as Spring MVC Framework to implement server side logic [Business logic].
- 2. @ Rest Controller is a Specialized version of @ Controller, it Contains 2
 - you can use the stage attropping Annotation to map reached
 - 近のController
 - tes in @ Response Body H 1490 pd dotal of satural to sooner sod -> it will convert the response in plaintext / JSON /XML
- Dis Advantages: +noirer tootrode siting bout on 917H 30/JSTL) that Means 1. @ Rest Controller doesn't work with view Technology (JSP/JSTL) that Means methods can't return ModelAndView object.
- 2. This Annotation will be used at <u>class level</u>, whereas @Controller will return Model And View object but it you want to display the response directly on the browser as a Message, we have to use @ Response Body Annotation at either <u>Class</u> level (on <u>Method level</u>.

@ Request Bod. Vs @ Response Bods
@ Request Body Vs @ Response Body. 1. These Annotations are used to find the HttpRequest (or)
Hetp Response as a Method Parameter (or) Method return value
Post to lively as a regiment post of the p
Respectivelymorport board andologo on solivor sum gringe
Request Body will Find the Hotep Request body to the Method Bromelo
Eg: -@ Request Body Employee empas Paraméter por apoiggon 1230p
2. @ Request Body will Find the Hetep Request body to the Method Baromele Eq: -@ Request Body Employee emp - Pavameter 150N. 3. @ Response Body will return the Http Response body to the browser. I post a paid to the browser. I appear to the browser.
religion The Hetp Response body to the browsers in about on the prices of the browsers in
@ Request Body (Input) whood szarogusy out of ulfosi
Response Response Body (output) 21 noisotonna .[21pol 22200200] Side logic [bourse standard of Avourance to animal to the animal
. O whater Annoration server side logic [Business logic].
→ you can use the @RequestMapping Annotation to map requests to
Controllers methods
-> It has various aftributes to Match by URL, HTTP method, request
Parameters, headers, and media types 2000 ANT travno Illia to
Request Mapping. Reader III @ Gret Mapping Parangel wall to be see all wood to so the and a life or the analy the analy the analytic and the analytic analyt
@ Request Mapping.
Rother Model Model Model And Standard Model Model And Standard Model Mod
Man Annetolion will be the you want to display prigaply to the want for display prigaply the tot want for display prigaply the tot want for the want
Modelland Vice bounder of the level (as) Method level on the hold & Merchalion at either store level (as) Method level on Method (as)
* @ Delete Mapping bout of Method for notificational
* @ Patch Mapping

Dynamic URLs Paths

@Path.Variable

@ Request Pavam 1170

/employees/123
/employees? id = 123

>it gets the value directly from the URL after &

- Use query Parameters when the

-> Use path variables when the Parameter is optional and used for

Parameter is an essential Part filtering, sorling, (or) other Modifications

of the URL Path that identifies to the request.

a resource. -> where we Open the Amazon Website

- Value Comes from the Path - luser/lidy

Request Body: -> value Comes from the query -> ? key = value

1. @Request Body is used to bind the HTTP request body to a Java object.

2 When a Client sends data in the body of a request (e.g., IsoNI or XML),

spiral for y - x signification of middle will adopt the

@ Request Body maps this data to a Java Objects.

7/5/25

tersistence Layer & JPA:-

Persistence Layer = 1210 and bno pange of the parting, updating, deleting and fetching data from the database. I home of bose

2. it works directly with the database Using IPA con IDBC.

3 it works with Entity Classes that represent database tables.

Ex:- a User Java class maps to a User table in the DB. 4. Spring Data JPA Provides <u>Jpa Repository</u> (or) <u>CrudRepository</u> interfaces.

5 You just define a repository interface (e.g., UserRepository), and Spring gives

you built-in CRUD methods (like Savers, FindById(s).

6 You don't write san manually. 7. IPA (with Hibernake) Converts your Java Code into see beind the scenes.

8 You can switch from H2 -> MysQL -> PostgresQL just by changing

a few settings - The Persistence Layer code remains the same.

Java Persistence API-JPA object JPQL ORM SQL MysoL Driver JPA: JPA API Connectorly JDBC Postgre SQL Hibernate de filosof hout a Postgre Spring H2 Database: -> Spring Boot makes it very easy to set up an in-memory H2 database for development and testing Purposes. -> Hz is a lightweight, tast and Open - source relational database engine that Can run in both in-memory and Persistent modes. Dependency -> LgroupId > Com. hzdatabase 2/groupId> @Entity Annotation:-1. The OEntity annotation in spring and Java Persistence API (JPA) is used to mark a class as a Persistent entity, Meaning it represents a table in a relational database now stop poidoled boo 2 This is a fundamental part of the ORM (Object Relational Mapping) Pava digm, where Java Objects are mapped to database tables. hey Points of motority of Overland The Provider The Prints of Date The Prints Vou just define a reposition interface (e.o. shortation of spiritary sould & Class - Level Annotation of the class - Level Ann you built in CADD methods (like Sover) lindbyide). * Primary key. Appropriate solven from my Pripado na troj se That you need to Convert into a Table in a fow settings - The Parsistense Loyer code Database.

@ Table -> To mention the Table name.	
it is used to indicate one of the att	vibules in the class as
Primary key Column (Primary key = Uni	que + not null)
>it is a Field Level Annotation	
internal internal	JOHON TIME COLUMN THEM
(column -> it is used to provide additional interests, nullable (true false), unique (true false), Precision and
Seale.	pipod zermeud zelband
Note:	in the same will be some
Lit you don't mention the Table name then the	e table hame will be will
as class name.	Changeno Componento
2. it you don't mention the Column name then	the Colomy Hame on a
Same as attribute/ troperty name.	
3 decimal (P.S) where P means Precision and S	means scale
Sample: decimal (4,2) 2 MR (0) 1/4 (10)20.	90) planned in suntain it.
112,45 (Progans 123 and 5 means 713)	(payotai paible
is a loose attribute in the	table is nomber (30).
Columbias 1301 Doll Men	the shoot will a sold
make any Column as Unique then Unique =	- Apple
TOAR - Rollary Totertace	
The TPAPenacitory interface in Spring Darc	JPA provides a <u>ser of</u>
CRUD (Create, Read, Update, Delete) Operations	. (100 100 100 100 100 100 100 100 100 10
interacting with the database.	of solt a stole a manage
> Key points of Crud Repository: 2000 tox	2116 cold another makes
* Generic Interface.	d daile landes a
District Methods &	
* Predefined Methods ()	L. bunnasgo Bg
# CO3(4) 10 10 1 1 10 10 1 1 10 10 10 10 10 10 1	- Name to POTO COM:
Custom Quevies of more special solutions of solutions of solutions of the contract of the cont	etters this transport box
Ex! - @ Gretter, @ Setter, @ All Args Constructor, @1	No Avg & Consider

8/5/25

Service Layer:

- 1. Acts as a Middleman: 1. de manie : mondo per per men
- - -> it sits between the Controller (Presentation layer) and the Repository (data access layer) to Separate Concerns and manage

or a comment with the second

- 2. Handles Business Logic: --> it contains all the Core business rules and Calculations, keeping then
- Separate From the Controller and database logic.
- 3. Coordinates Components: → it Manages interactions blw different services, repositories and external APIs to folfill a request.
- y. Provides Clean interfaces: 1000 policional account of the Services to
 → it offers a Simple, Consistent API for the Controller (or) other Services to
 - Use, hiding internal Complexities, 2 bas all amount) 21/221 /2
- Jese, maintainability: I southly reported to select, scale and separating logic, it makes the Application easier to debog, test, scale and update.
 - -> The Service layer Acts as a bridge blw the <u>Persistence</u> layer (responsible for data access) and the Presentation layer (handling user interaction).
 - -> It encapsulates the business logic of the application, Orchestrates interactions blu different Components, and Provides a clean interface
- for external clients to interact with the System. -> By abstracting away the Complexities of data access and business Operations, the Service layer Promotes Modularity, Maintainability, Scalability. 24 of the control of th

- 1. Controller Layer (Entry Point):-→ The Controller handles incoming API requests (like GET/ POST). - it doesn't directly interact with the database. - it sends the request to the Service Layer. gricul di el sará 2 Service Layer (Business Logic):--> This layer Contains the main business logic. -> it ack as a bridge blu Controller and Repository. -> Ensures Clean <u>Separation</u> of <u>Concerns</u> and <u>Code reuse</u> (<u>DRY principle</u>). 3. Repository Layer (Database Acces):--> Handles all database Operations like CRUD. - Only the Service bayer should talk to it, not the Controller directly. → Direct Communication from Controller To Repository (or) Controller to DTO breaks layered Architecture. -> That's why we want Centralized logic in Service Layer. 5- Benefits: - DRY, Logging, Security -> keeping logic in one layer avoids duplication -> Easier to apply logging, Security, Validation Centrally in Service Layer.
- ModelMapper:
- 1. ModelMapper is a Java library used to <u>Automatically map objects</u> from One type to <u>another</u>.
- 2. Mostly used to Convert blu Entity Clauses and DTOs (Data Transfer Objects).
 3. Commonly used in REST APIs to Avoid exposing internal database entities
- 3. Commonly used in REST APIs to Avoid exposing internal database entities directly.