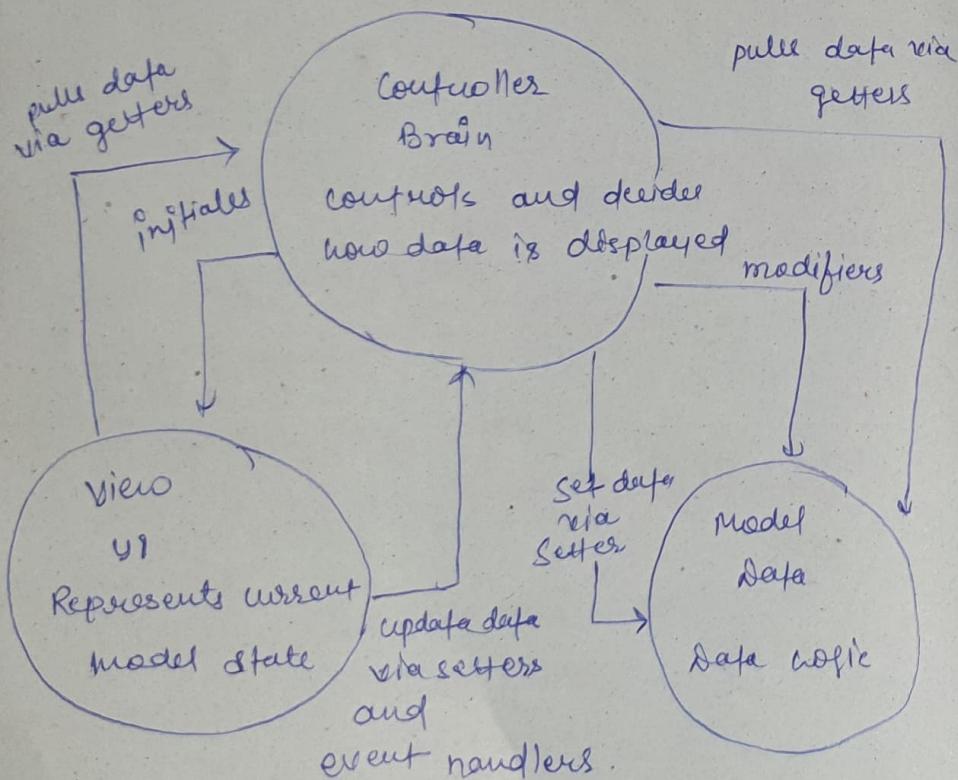
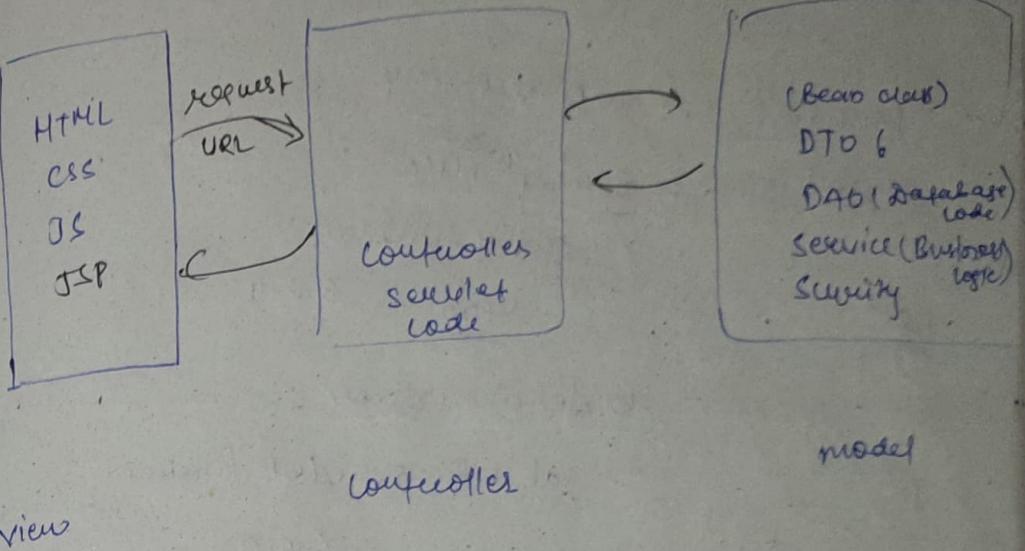


Spring MVC

- Is a Java framework which is used to build web-applications.
- clear separation of roles.
- Customisable binding and validation.
- Customisable handler mapping and view resolution.
- Reusable business code, no need for duplication.
- flexible model transfers.
- Customisable locale and theme resolution
Support for JSP with or without Spring tag library, support for JSTL.
- easy to test and build.

MVC architecture





DS ser
↓
dispatch

02

Maven Project → same as servlet project

pom.xml → mvc repository

dependencies : Spring web-mvc

: (Spring-webmvc)

6.0.9

5440

1.8

→ servlet dependencies

→ jakarta-servlet (latest version)

6.0.0

A

src/main/java

com.ty

→ @Controller →

→ @GetMapping ("hi");

→ configure DispatcherServlet.

web.xml

====

* libraries

↳ maven dependencies

↓
spring-webmvc-6.0.9.jar

↓
org.springframework.web.servlet

↓
DispatcherServlet.class

↓
copy fully qualified class name

create .xml file for bean configuration

↓ Note XML file should have same name as servlet name from web.xml

↓
eng:

Web-INF → create new xml file

↓ student - servlet.xml

bean configuration

xml file name

→ go vikas banavara
git & repository
Suppose we file
↓

[Spring - xsd - xml]

copy
=

& paste in

[student - servlet.xml file]

- * configure paths of controller package
- * create jsp file (of web app)
- * student.jsp → [ns]

↳ controller

return "Student.jsp";

exception 500 , 404,

Class Not Found Exception

b) Server → New → Server → Apache → 10.0 server

→ fish

3. Creating spring mvc maven project and annotations based configuration.

maven project

pom.xml →

- Spring web mvc
- Jakarta-servlet-api.

~~web.xml~~ → class based configuration,

* Initializes class

- ① StudentWebInitializer.java
- ② configuration class → servlet config class
- ③ controller class

website gfe,
Spring boot
REST API intro.

u 2
y

Spring boot

1. Introduction to web services

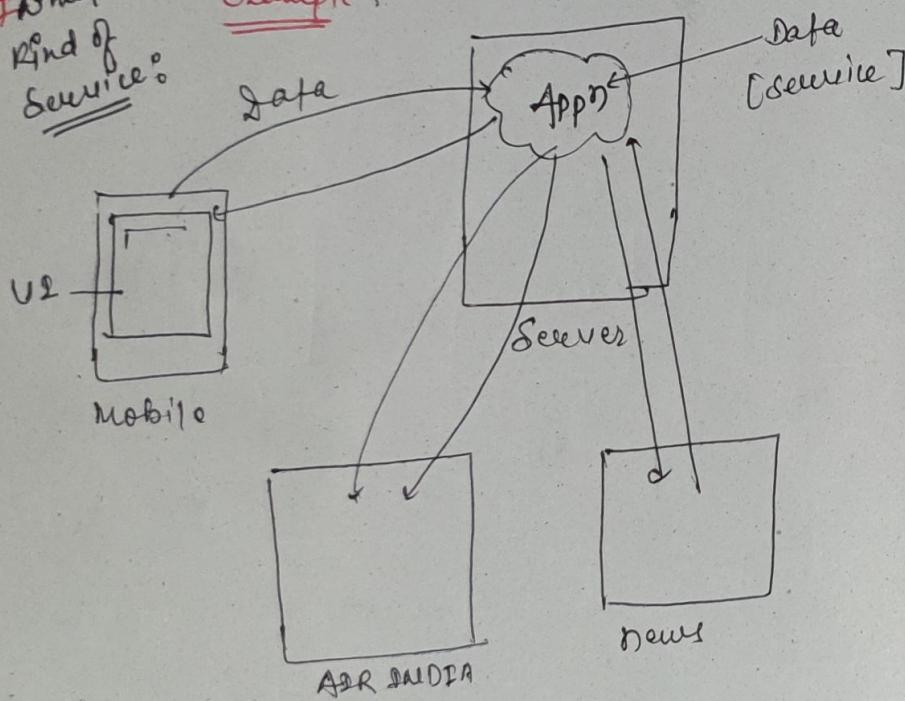
web service

- A service which can be accessed over an internet called as web service.

web application :-

Any application which can be accessed using internet called as web application

what kind of service : example : weather Report



- An app which provide the data or service called as web service.
- web service is also a web application
→ all web service are web application but all web applications are not web services.

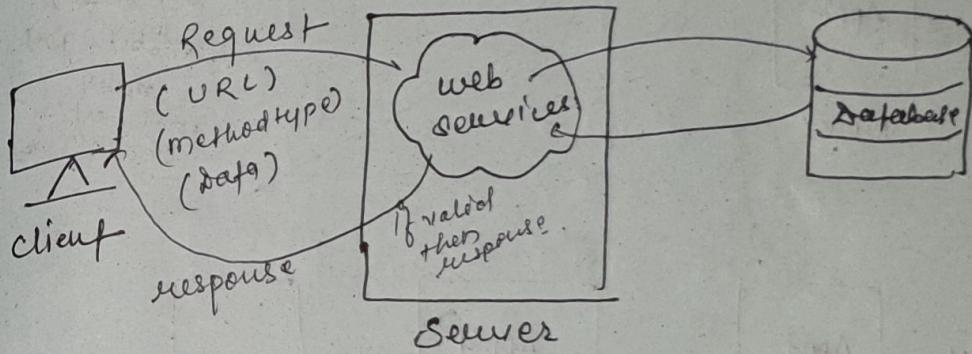
→ to build web service we follow some architecture,

• That is make use of "REST"
(Representational State Transfer)

REST is an architecture to build or help to build web service applications.

→ whenever we creating web service using REST architecture the service is known as RESTful services.

Q2. How does web service work.



Request contain

- URL has information what I want. [mandatory]
- Request has method type, like, post, get, delete etc. in request to invoke.
- Data [optional]

Note:

Request

- * URL *
- * Method type *
- Data [optional]

1. Informational responses

1XX (100-199)

2. Successful responses

2XX (200-299)

3. Redirection message

3XX (300-399)

4. Client error responses (400-499)

4XX (400-499)

Response

- Status code 404, 500 etc
- Content type (image, json, video)
- Actual content

= How does web services share the data...?

take one example whether report ...
to display

developed

C#

share)

APP1

AirDnida

+ {text}
{JSON}
{XML}

share the
data only

data

JSON
XML

Databas

web
Service

web server

Java
App2

Netel

"text has problem"
(Parse ambig
of date)
readability goes
messy
not well
organis
mann

Ruby

App3

website

APP4

Android

App5

App6

App7

JSON: Javascript Object Notation

- I should share data in universal language so any language can understand data.

i.e. formatted structure.

parser at JSON i.e. JSON → JSON will understand by any language.
over Java or both

- processing will convert the JSON object to java object or other language.

responding & consuming data by the web service in form of XML, JSON or both.

What is JSON

- JSON stand for java script Object Notation.
- It is easy to read or, parse, describe, understand the data in JSON format.
- JSON Object are light weight.
- JSON Object are well Standard.
- JSON Object can store, String, Number, Boolean, null.

How to create JSON Object

Class Student

```
{  
    int id;  
  
    String name;  
    double weight;
```

representation of JSON Object

JSON Object { "key": "value" }

→ open close flower braces.

all key should be in double quote.

```
{  
    "id": 10,  
    "name": "priya",  
    "height": 5.16
```

Student
JSON
Object.

Q5. JSON with Java Collection

How to → Multiple values?

Store: or array =

example

```
class Car {
```

String name;

double cost;

String[] color; → array

Represent array
in
JSON Object

JSON Array: []

Object used to represent
{ }

```
{
    "name": "Swift",
    "cost": 25000,
    "color": ["red", "blue", "green"]
}
```

↑ Key ↑ array start ↑ value ↑ array end.

representation
collection in JSON =

JSON Object

class Car

{ string name;

double cost;

List<String> color;

"name": "Swift",

"cost": 25000,

"color": [

"pink",
"red",
"green",
"yellow"

}

represent Object in JSON

example

class Tv

{ string name;
double cost;

}

class House

{ int dNo;
String city;
Tv t;

example

Array of Object

represent multiple object in JSON

class Account

{ } = "acc : [{ }] :

class Bank

{ } = "list <Account> Acc;"

]

↑
array of
Object

#2 download postman

Assignment

Class Product

```
{  
    String productName;  
    double productCost;  
    String productBrand;
```

Review review;

Class Review

```
{  
    int reviewed;  
    String Username;  
    String description;
```

JSON Object:

```
{  
    "productName": "ABC",  
    "productCost": 700,  
    "productBrand": "X42",  
    "review": {  
        "reviewed": 101,  
        "Username": "BCD",  
        "description": "AIB"  
    }  
}
```

06. Build Spring boot project: Theories

→ to build web services we can make use of any programming lang.
C#, java, ruby, python

using Java

→ JAX-RS

→ Spring REST

→ Spring Boot

] to create restful web service

→ to create Spring boot project use search (Spring Initializer) go to browser.

① <https://start.spring.io>

- ① Project type → Maven

② Language → java

③ Spring boot → 3.0.1

④ Project Metadata. ↗ Group → Project Name
 ↗ Artifact → (Simpleboot-api)

⑤ Dependencies ↗ Name (11 version)

↳ spring web

⑥ Generate project → zip folder to

Import → maven
↓
existing maven project
↓
next
↓
(Browser)
Select
↓
Finish
↓
↳ wait until 100%.

Q7. Practical

→ Driver class

@SpringBootApplication

public class SimpleBootApplication

?

public static void main (String [] args)

?

SpringApplication.run (SimpleBootApplication.class, args);

?

g

TestController.java

② RestControllers

public class TestController

? @GetMapping ("message")

public String message ()

? returns "first message";

4

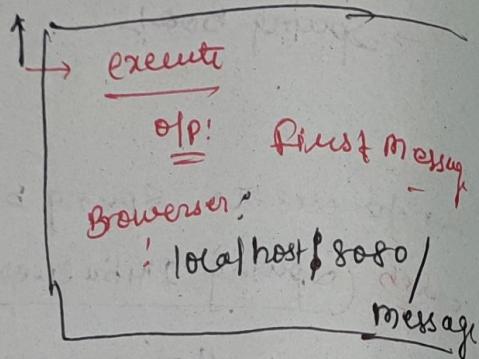
@GetMapping ("age")

public int age ()

? return 12;

5

6



whenever we add new mapping you need to restart the server.

08

Controller class and Mapping annotations

@SpringBootApplication

Driver Class

↳ It is a default class Spring executes first from here.

→ It contains main method.

→ need to create controller class ^{to} create to create end point

→ /student }
→ /age } endpoint ⇒ controller
→ /info } class

How to make controller class

using

@RestController

↳ annotations

@Controller
↑ extends
@RestController.

↳ sub class of @Controller

@RestController

class Demo

- mapping
- method

request → URL →

Spring controllers
will always
look for the rest
@Controller class.

@Rest Controllers

Class Demo}

@GetMapping (" /sweety")

String m ()

return "hello";

@DeleteMapping (" /delete")

int n ()

return 1;

}

}

HTTP

Mapping annotations

@GetMapping

@PostMapping

@PutMapping

@DeleteMapping

@PathMapping

} use any annotation to trigger the method

end point ↓

request:

URL : http://localhost:8080/sweety

Method type: Get

How to send request using Postman

open post man

↓

give the url

we can't make post request using browser, so use postman.

09. Types of mapping annotations

@GetMapping
@PutMapping
@PostMapping

String on()

} returns "Hi";

{

→ each mapping method design will perform specific task.

functionality.

→ In industry we have following rule.
(convention)

Save any record → use "POST" mapping
@PostMapping

Retrive the record → "GET" mapping
@GetMapping

Update the record → "PUT" mapping
@PutMapping

Delete the record → "DELETE" mapping
@DeleteMapping

Small change | Partial update → "PATCH" mapping
@PatchMapping

→ Before use, see what kind of task method will perform, based on that we make respective mapping.

i.e. Industry convention =

<u>Method</u>	<u>method-type</u>	<u>URL</u>
Save	POST @PostMapping("/students")	/students
Delete	DELETE	/students
Get	GET	/students
Update	PUT change in method type	/students common

- * @GetMapping("/Students")


```
int m() } --- { GET /students
```
- * @PostMapping("/Students")


```
double y() } --- { POST /students
```
- * @PutMapping("/Students")


```
char t() } --- { } error
```

so, either change the method type of URL
so unique URLs

→ URL can't specify what type of operation we are going to perform so make URL as common & change the method type only.

10. Trigger the end point

src/main/java

(Driver class)

project name : school / college

Driver class name } @SpringBootApplication
School Application.java
College Application.java

having.

& annotated with → main() method.

@SpringBootApplication

* Execution of Spring Boot Application always starts with Driver class.

<projectName> Application.java

Spring need to identify which file need to trigger for execution & execution starts with main() method.
& class annotated with @SpringBootApplication.

How to design end point

* @RestController
class TestController

}

@GetMapping("/info")

public String message()

? returns " ---";

4

URL : http://localhost:8080/info

Method type: GET

11. Pass Value using QueryString : Theory

- QueryString, piece of information which is going to travel along with URL
 - Query string travel with data in form of key & value pairs.

eg
URL : 1student?id=10

↑ ↑
key value

↓
Separate
URL with
QueryString

1employee?empid=101&companyid=unidgog

↑
each value

key value

separated
with ampersand
(&)

How to read Query string in method?

1student?id=10
read
@GetMapping("/student")
public String info(@RequestParam int id){
}

@RequestParam

help me read the query string & store inside variable.

Note

① Query string variable & method formal argument / & Should be same

1 Student ? id=10

@RequestParam int id

12. Practical

Create URL: in postman

http://localhost:8080/student?myId=10

& id:5

@RestController

public class TestController

{ @GetMapping("student")

public String myInfo(@RequestParam int myId)

{ return "my Id is :" + myId;

}

4

don't want change URL & name.

Search for automatic Server start

Spring boot Dev Tool - in Spring Boot

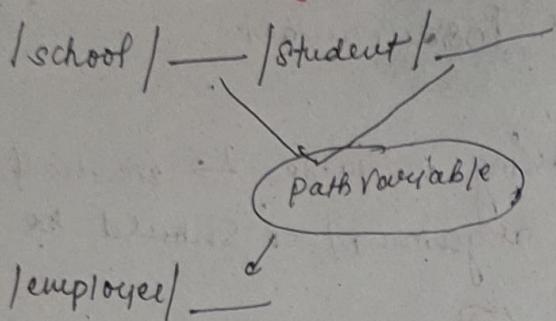
@GetMapping("emp")

public String empInfo(@RequestParam(name="myId") int id)

{ return " " + id;

{

13. Pass Value using Path variable (theory.)



→ using path variable we are sending information in URL

e.g. /company/{cid} ~~dynamic~~ ^{value need}

```
public code = @GetMapping("/company/{cid}")  
public String info(@PathVariable int cid)  
{  
    return " " + cid;  
}
```

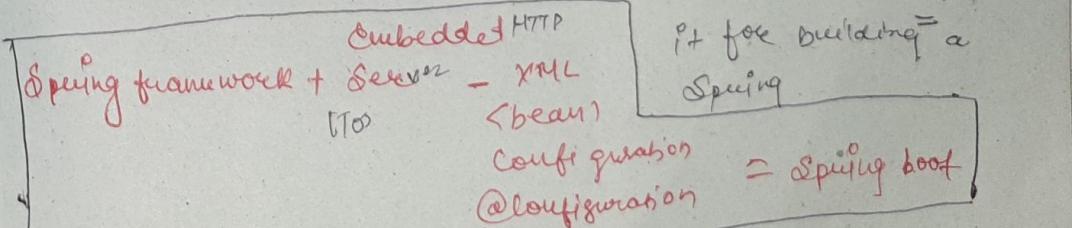
@GetMapping("/company/{cid}/{name}")

```
public String info(@PathVariable int cid,  
                   @PathVariable String name;  
?  
    return cid + " " + name;
```

UR
http://localhost/company/101/name/Shelobham

Difference b/w Spring MVC and Spring boot

- Spring MVC is a Model view and controller based framework, widely used to develop web applications.
- Spring boot is built on top of the conventional Spring framework. widely used to develop REST APIs.
- if we are using Spring MVC we need to build the configuration manually
- In the Spring MVC, a development descriptor is required. (web.xml)
- there is no need for a deployment descriptor.
- Spring MVC specifies each dependency separately
- It wraps the dependencies together in a single unit.
- Spring MVC framework consists of four component model, view, controller, front controller
- presentation layer
Data Access layer
Service layer
Integration layer



Spring framework is combination of Spring framework and Embedded Server.

14. Pass Value using header parameter

We can share data in form of ^{information} data

Query string → key = value

Path parameter
variable → | ~ |

Request
URL

[Request Header]
{key = value}

Share content type

credentials info

Tokens

in form of

key - value

to be share with
expert object

request header part of Request
Object

Program

@RestController

public class TestController

help to read data
from request header

@GetMapping("info")

public String test(@RequestHeader String token)

return "My token is :" + token;

}

go to postman

set

Header
↓
token

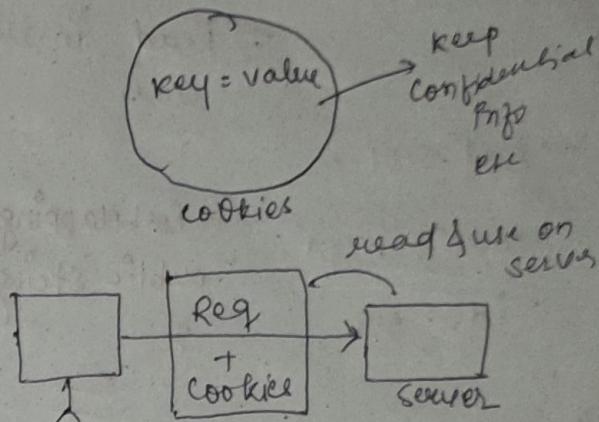
.vikas@123

URL : http://localhost:8080/info

7b. Pass value using cookies

How share data with cookies

- cookies stored in client place.



why cookie

- need to share information from

How to read value from cookie

@CookieValue

Program

Create cookie in Postman

post man:-

cookies → type (localhost) → add cookie

bearer = JWTOKEN67 ; path / ; expires: Fr

key matched
with the
format org

name = Shubham ; path / ; ex -

@RestController

public class TestController

? @GetMapping (" /data ")

public String testcookie (@CookieValue String bearer,
@CookieValue String name)

? return "My cookie is " + bearer + " " + name;

G

How to pass json object using request body: Theory

→ Read in the controller class

"roll": 101,

"name": "Rajiv",

"id": 5.2

@PostMapping("student")

public String save(@RequestBody

Student

stu)

Student JSON

Object

need to convert in java object

from request

using @~~St~~ **@RequestBody**

@RequestBody :-

Read the object from requestBody
& mapping the respective variable
of student class / entity should be
in the project.

18.

Program

create a class

Student.java

class Student

{ private int roll;

private String name;

private double height;

// public getter & setter.

controller class

@RestController

public class TestController

? @PostMapping (" /mystudent")

public string receive(@RequestBody Student student)

} return "student name is: " + student.getName();

}

pass the student object using UR

postman →

http://localhost:8080/mystudent + post

select → Body → row → ~~Text~~ → JSON (application/json)

}
 "id": 101
 "name": "prerna",
 "height": 5.8

→ send → off

student name is

4

prerna

19. Spring boot project with DB

- ① Spring Initializer → simple-boot-dB-ap
↳ create one project
add dependencies
→ Spring web
→ dev tools
→ Spring data JPA
(interact with)
Java version: 11
→ drivers S/W
(PostgreSQL Driver)

- ② interact with database
need: URL / user / pass password

→ src/main/resources

↓
application.properties

- Paste here
→ change db Name, username, password

Note go to browser
github.com/vikasbanavar

repositories

configuration files

application.properties for postgres
copy here

if error
Note

* if error :- failed to determine

How to fix it.

go → pom.xml

within ing

Artifact ID Spring-boot-starter-data-JPA1)

→ scope provider < 1.8 (OP)

if no error → successful
db configuration

* if you make any mistake
then you will get exception

Note

* if entity is not
converting
Create new
word code
& project
=

20. Example to insert Record

- ① create an entity class

student.java

@Entity → package jakarta.persistence

public class student

{ @Id

private int id;

private string name;

private double height;

11 getter & setter

}

- ② create Interface & extends JpaRepository & make it generic.

public interface StudentRepository extends JpaRepository<Student, Integer>

2

6

- ③ Create controller

@RestController

public class TeststudentController

{
 }
 1 Create an object for Student
 repository

@Autowired

private studentrepository studentrepository;

@PostMapping (" / save")

public string saveStudent
 (@RequestBody student s)

{
 }
 studentrepository.save(s);
 return "student is saved";

}

Create JSON object

{
 }
 " id": 101
 " name": "perva
 " height": 7.9

curl

post | http://localhost:8080 / save

11 Find all the record

@GetMapping (" /getall")

public List < Student > getAllStudent()

{
List < Student > list = studentRepository.findAll();
return list;

4

postman

Get | <http://localhost:8080/getall> ↗

[opt

JSON Object format

"Get an record by id"

@GetMapping (" /{id}")

public Student getStudentById (@PathVariable int id)

{
Optional < Student > opt = studentRepository.findById(id);

if (opt.isPresent())

return opt.get();

else

return null;

5

update the record

@PutMapping ("update")

```
public String updateStudent(@RequestBody  
                           Student s)
```

}

```
StudentRepository.save(s);  
return "Student updated";
```

g

Note:

* if primary key existing
 save method act as update method
 or update in existing object

2 if primary key is not present
 two new object will create and
 store.

JSON }

```
{"id": 101,  
 "name": "Shikha",  
 "height": 7.6}
```

f

URL
= put | <http://localhost/update>

11 Delete the record by Id

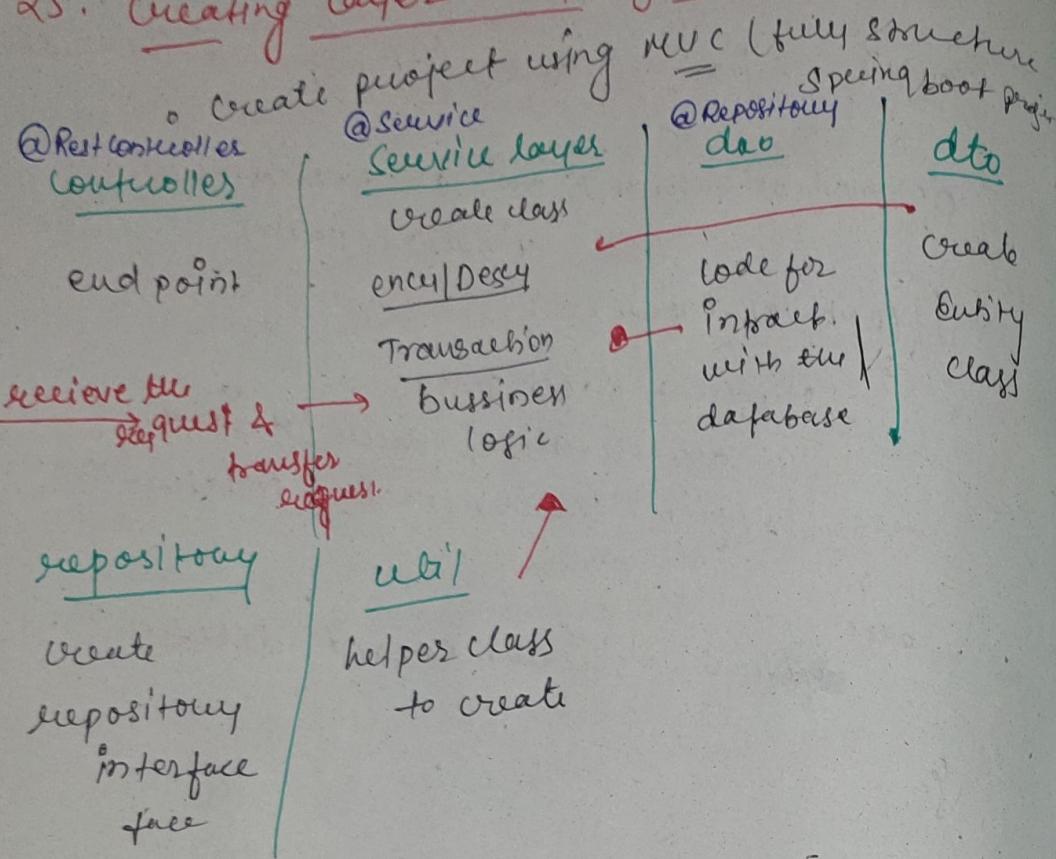
```
@DeleteMapping ("remove/{studId}")  
public String deleteRecord (@PathVariable int studId)  
{  
    Optional<Student> opt = studentRepository.  
        findByid (studId);  
    if (opt.isPresent())  
    {  
        Student s = opt.get();  
        studentRepository.delete (s);  
        return "Student Record Deleted";  
    }  
    else  
        return "Student Record deleted";  
}
```

postman:

```
[Delete | help://localhost:8080/remove/101] el
```

Status code

Q5. Creating layers in spring boot



- controller receive the request & transfer to server layer
- Service layer talk to → dto layer
dao layer
util layer
- only
- dao take to help to repository it's
~~nothing~~ no other layer can talk.

controller classes for all controller class use

@RestController

Service classes

@Service

dao classes

@Repository

helper class / util class / configuration class

@Component

@Configuration

26. How to use @RequestMapping : Theory

↓ use top of the controller class

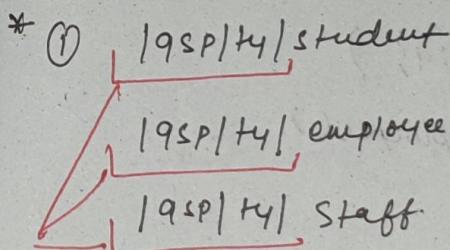
@RestController

@RequestMapping (" /qsp/t4") Starting for all class Demo

{ @PostMapping (" /qsp/t4 / student") → " /qsp/t4 / student" } m() = { }

@GetMapping (" /qsp/t4 / employee") m() = { }

@PutMapping (" /qsp/t4 / staff") y() = { }



Common url

So no need to use each end point

So use @Requestmapping at the top of two controllers. & like

@RestController
@RequestMapping (" /qsp/t4 ")

class Demo

? = { }

① 1. /qsp/t4 / employee

Program

controller

Employee.java

@RestController

@RequestMapping(" /employees /Employee")

public class EmployeeController

URL;

/employees /Employee /save

@PostMapping (" /save")

public String saveEmployee

return " ";

}

@GetMapping (" /getall")

public List getAllEmployee()

return null;

Q

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

public Object getEmployee()

return null;

F

@DeleteMapping (" /delete/{id}")

public String deleteEmployee()

return null;

G

@PutMapping (" /update")

public String updateEmployee()

return null;

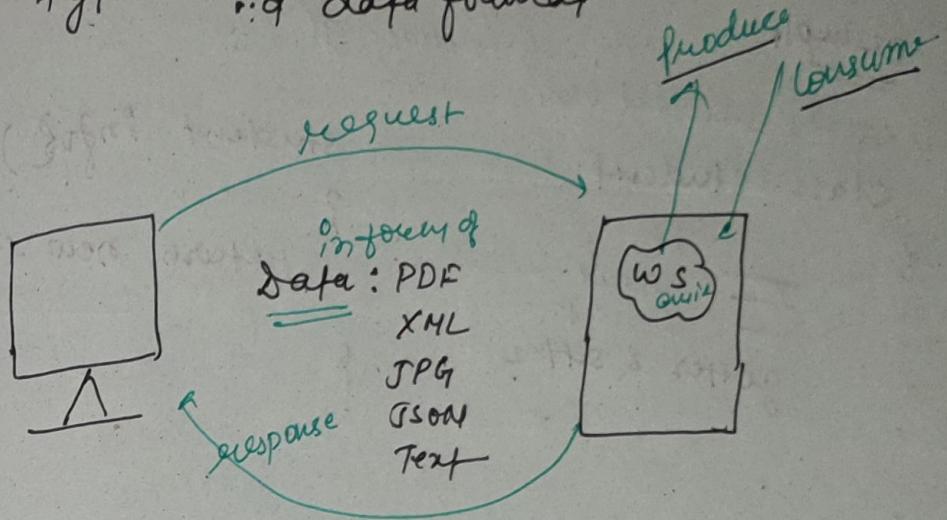
{ }.

& Test all method

Media type - ~~represents the content type consumed or produced.~~

28.

- * Media type represents the format of type i.e. data format



data format: PDF, XML, JPG, JSON, Text etc.

- * Whenever exchanging data we need to specify the data format

* by default media type is JSON
in request & response.

- * we need to specify to web service what type of data format to produce or consume.

: by default JSON format

* produce & consume are parameters of my mapping method

* produce & consume of my mapping method

→ if we want to ~~consum~~ web service

consume XML, Rfc data format

then we make use of consume parameter with in mapping.

produce

* if we want to ~~web service~~
produce the XML format data

then make use of produce parameter in mapping.

example

① create student class

class student

{
 id;
 name;
 height
 getter & setter}

@GetMapping("/test")

student info()

? returns new student()
 JSON format

4

JSON format:

program: @RestController

public class DemoController

{
 @GetMapping("/test")
 public Student getStudent()

? Student s = new Student()
 s.setId(101);
 s.setName("Shubu");
 s.setHeight(6.7);

return s;

5

URL: http://localhost:8080/test

XML format

Note make use of "produce" parameter
but not directly in mapping.

* ~~URL base~~

whenever there are multiple
parameters declared with in
getmapping , then it has every

parameters should have values
attribute.

i.e.
↓
@GetMapping (value: "/test", produces =
MediaType.APPLICATION_XML_VALUE)

public Student getStudent()

{ Student s = new Student();

s.setId(101);

s.setName("Shubhi");

s.setHeight(3.8);

return s;

↓
e.g. Spring framework
work. help.

}

500

problem error: Internal server error

because XML is not default
data format

solution

to add dependencies (XML
dependencies).

gitHub → repository → configuration file

Starter dependencies

<groupId> com.fasterxml.jackson-dataformat
</groupId>
<artifactId> jackson-dataformat-xml
</artifactId>

→ add the dependency in

POM.XML

O/P:

<Student>

<id> 101 </id>

<name> Shubham </name>

<height> 5.1 </height>

</Student>

=

consume the XML data

@PostMapping(value = "/receive", consumes =

MediaType.APPLICATION_XML_VALUE)

public String readObj(@RequestBody Student student) {

}
return "My student name is " + student.getName();

}

4

Postman

(60)

post / http://localhost/receive

body - raw → XML

<Student>

<id> 101 </id>

<name> Shubh </name>

<height> 5.1 </height>

</Student>

O/P:

My student name is Shubh ↴

produce & consume data

@GetMapping(value = "/mystudent",
 produces = MediaType.APPLICATION_XML_VALUE,
 consumes = MediaType.APPLICATION_XML_VALUE)

public Student getMyStudent (@RequestBody
 Student student)

{ return student;

}

postman

GET | http://localhost:8080/mystudent

Body - raw - XML

==

↙

consume is XML/JSON & produce XML/JSON

@PostMapping(value = "/myinfo", consumes =

} { MediaType.APPLICATION_XML_VALUE,
 MediaType.APPLICATION_JSON_VALUE},
 produces = {MediaType.APPLICATION_XML_VALUE,
 MediaType.APPLICATION_JSON_VALUE})

public Student getStudent (@RequestBody
 Student student)

{ return student;

}

29. Response Structure

How to create Response structure.

Class Response structure $\langle T \rangle$ → Generic

→ Industry $\xrightarrow{\text{map to}}$ Standard name

{

private int status code;

private String message;

private $\langle T \rangle$ data;

|| getter & setter ()

or not successful message

Parameter to send to the customer

e.g. Employee
student
collection...etc.

}

what my response should contain
or show to customer

Every company has different different
Response structure

• Resp. common for all end point

student.java

class Student

{

private int roll;

private String message;

private double height;

|| getter & setter ()

}

StudentController.java

@RestController

public class StudentController

} @PostMapping("/save")

public ResponseStructure<Student> saveStudent

(@RequestBody Student student)

} ResponseStructure<Student> rs = new

ResponseStructure();

rs.setStatuscode(200);

rs.setMessage("success");

rs.setData(Student);

return rs;

}

Postman

POST http://localhost:8080/save

Body → raw → JSON

{ "roll": 101,
"name": "Shubhi",
"height": 7.3

}

Output
{ "statuscode": 200,
"message": "success",
"data": {

"roll": 101,
"name": "Shubhi",
"height": 7.3

}

{

31. what & why Resourcefulty

GetMapping

@RestController (" /student/{roll}")

public ResponseStructure < Student > getByRoll
(@PathVariable int roll)

{ if (roll == 1)

Student student = new Student();
student.setRoll(1);
student.setName("Dimple");
student.setAge(22);

ResponseStructure < Student > rs = new

ResponseStructure < Student >;

rs.setStatuscode(200);
rs.setMessage("success");
rs.setData(student);
return student; };

}

else {

ResponseStructure < Student > rs =

new ResponseStructure < Student >();

rs.setStatuscode(404);
rs.setMessage("Id not found");
rs.setData(null);

return rs; }

status code 200 or
200 OK

{

URL
old
Get / http://localhost:8080
/student/1

or

200 OK

problem



? -- - 404

data = null

But status
code
200 OK

to alter the

- I'm changing the body of Response,
 - I'm not altering the ^{http} response!
 - to alter, the ^{entire} http response
- I need help of Response Entity ..

} so far

* Response Entity is a structure of a HTTP response

inbuilt class

going contain statuscode
contain body / data
contain headers [optional]
↳ Content-type
↳ Actual content

statuscode

body / data

headers

• instead of returning http response,
return Response Entity.

How to return the Response Entity?

ResponseEntity contain .

body : ResponseStructure

statuscode : 200, 201, 400, 404, 403 ..

new ResponseEntity<ResponseStructure>(data, statuscode)

new ResponseEntity<ResponseStructure<Student>>
(data, statuscode) | Employee
| User

Program

if
new ResponseEntity<ResponseStructure<Student>>

(res, HttpStatus.OK); → all 200 ok

else?

new ResponseEntity<ResponseStructure<Student>>

(res, HttpStatus.NOT_FOUND); → 404 not found

~~It is recommended to return ResponseStructure Response Entity~~

~~→ it changes response.~~

~~• ResponseStructure is a bad practice ..~~

~~to use because it not
change the response.~~

note
• It has

• The Response Entity has capability & capacity to alter the entire http Response.

• ResponseStructure

can alter the body part of Http response.

82. Creating and handling custom Exception - in Spring boot

student.java

```
↳ id  
↳ name  
↳ height
```

ResponseStructure.java < T >

```
int successCode  
String message  
T data
```

StudentController.java

ResponseEntity ↗ previous lab

Problem

```
{ "id": 1, null  
  "name": "Shubham",    ↗ 500 error  
  "height": 7.6  
  ↗ exception }
```

33

def

Exception Handler = creating Controller advice class.

MyAppExceptionController.java

@ControllerAdvice

+ class MyAppExceptionController extends

```
ResponseEntityExceptionHandler { }
```

@ExceptionHandler (NullPointerException.class)

public ResponseEntity<ResponseStructure<String>>

```
handleNullPointerException(NullPointerException e) { }
```

```
} ResponseStructure<String> rs = new ResponseStructure();  
rs.setStatusCode(HttpStatus.BAD_REQUEST.value());
```

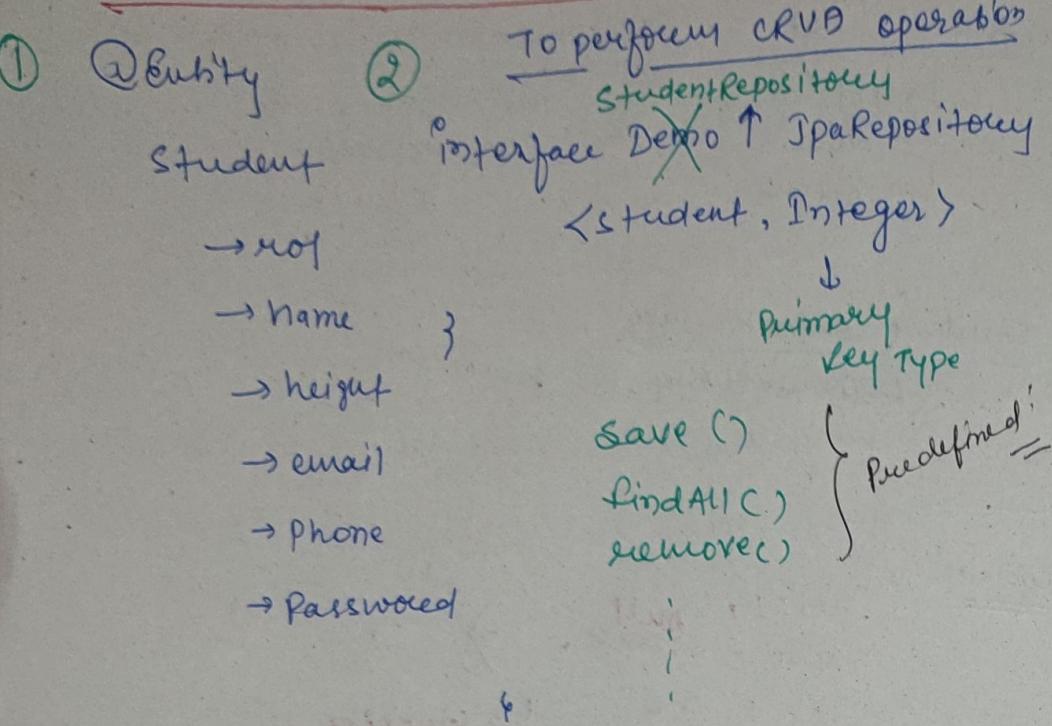
```
e.printStackTrace(); rs.setMessage("Message: " + e.getMessage());
```

```
rs.setData("Don't deal with null");
```

```
return new ResponseEntity<ResponseStructure<String>>(
```

```
(rs, HttpStatus.BAD_REQUEST));
```

34. Custom methods in Repository - Theory



How can we write custom Method in repository interface.

`List<student> findByName (String name)`

↳ method name should start by "findByName"
then variable

`List<student> findByHeight(---);`

`List<student> findByEmail(--);`

`List<student> findByPhone(--);`

`List<student> findByHeightGreaterThanOrEqual(double d);`

Spring Data JPA

* in back end query will create
like select * from student
where height >= d;

* Spring Container will create a query -

list Student findByEmailAndPassword (String e, String p);

①

Employee

↳ id

name

email

password

phone

age

② Interface EmployeeRepository

extends JpaRepository

<Employee, Integer>

{
List<Employee> findByName (String e);

Employee findByEmailAndPassword
(String email, String password)

{

③ Controllers:

③ Program

@Mapper ("employee")
public List<Employee> getAllAge (@PathVariable String age,
String name)

{ refers repository . findByName (name);

4

URL Get http://localhost:8080/employee/searchName

④

{

—

4

(36) Custom Query in Repository - Theory

Interface StudentRepository extends

JPARepository < Student, Integer >

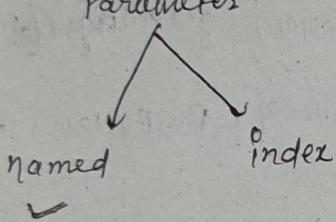
}

pass
JPQL query as String parameter

@Query ("SELECT s FROM Student s WHERE s.status = ?")
List < Student > getInfo()

↓
alias
=

* Parameters to set a value



* eg:
Index Parameter

dynamic query

① @Query("SELECT e FROM Employee e WHERE e.role = ?1")

1st parameter
(index)

List < Employee > getEmp (String role)

② @Query("SELECT e FROM Employee e WHERE e.age = ?1,
e.email = ?2, e.phone = ?3")

List < Employee > getEmp (int age, String email, long phone)

(1)

(2)

(3) parameter

Named Parameter

: Variable

@Query ("select e from Employee e
e.name = : emp-name")

Variable

Employee find (string emp-name)

~~At *~~

Variable name in Query & in
method param parameter should
be same.

② @Query ("select e from Employee e
e.role = : emp-role")

List<Employee> getRole (@Param("emp-role") string r)

both variable
name change

So make use of
@Param("emp-role")

Project

- add all dependencies
 - add application properties
 - Everything should be in Base package
 - ~~add dependency~~
- com.qsp.Employeeapp
- packages
- {
- controller
 - repository
 - service
 - dto/entity
 - util
 - configuration
 - dao

<scop> import <scop>

(1) ~~entity~~ entity

Employee.java

id
name
email
Salary
getter & setters

(2) repository

interface EmployeeRepository
extends JpaRepository

(3) dao

@Repository
EmployeeDao
save
getBy Id

(3) • service

@Service

Employee Dao

(4) • dto

Response Structure (T)

(5) • controller

@RestController

EmployeeController.java

→ EmployeeService employeeService

(6) run the project

If error:

- data-spa

<scope> import <scope>
provided ✓

(7) • exception custom exception
exception Handler

⑧ Service controller

Student/save

name
email
phonenumber
password
mark

Coworkspace

Wst <Building> building

id

Name

city

address

Building

buildingId

buildingName

area

address

Wst <floor> Floor

Method

{ create
update
cowork
space }
find → id, name
Delete →

Building

create
update
findAll
Delete

floor
create
find
update

workspace
CRUD

company
CRUD

ROOM
CRUD

Shubham Kumar