

Contents

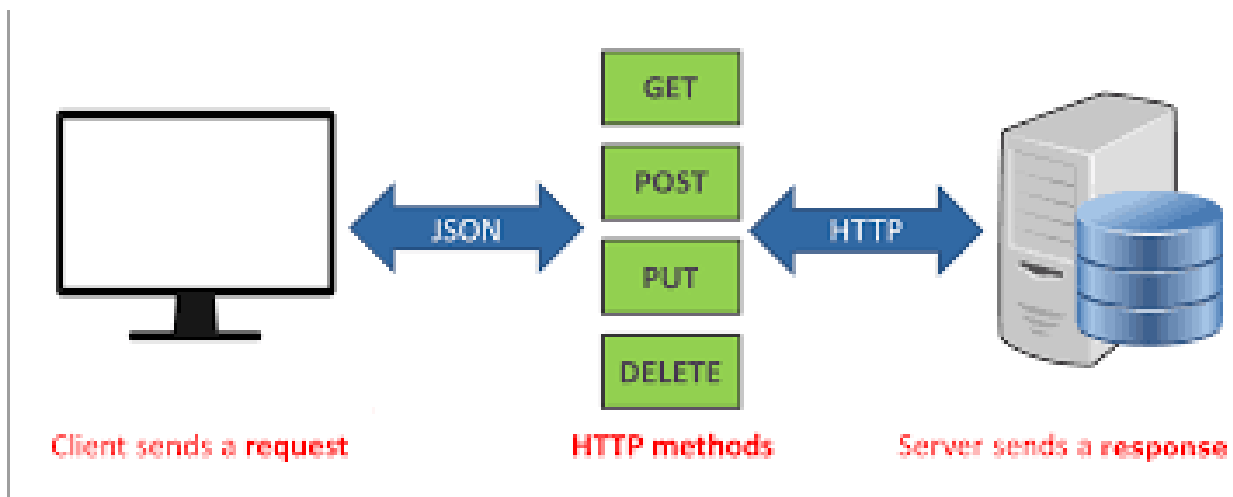
- REST API
- HTTP REQUEST (HTTP REQUEST METHODS , HTTP RESPONSE)
- HTTP RESPONSE
- POSTMAN : AGENT FOR API TESTING , CURL
- ANNOTATIONS
- LOMBOK
- SPRING MVC

REST API : Application Programming Interface .



A Short Introduction of REST : REST stands for representational state transfer. Basically, it's an architectural style for designing networked applications. REST relies on a stateless, client-server protocol and in almost all cases it's going to be HTTP. Initially, programmers were relying on SOAP to implement the API in web services but in recent years REST has become the choice of programmers due to its simplicity and scalability.

REST was made to treat objects on the server-side as resources that can be created, updated, and deleted. REST can be used by virtually any programming language. Let's discuss the benefits of using REST.



HTTP Request

You make HTTP calls sending the HTTP Request. HTTP request methods include **Request Method, Request URL, Request Headers, Request Body, Pre-request Script, and Tests.**

Let's talk about these Request methods one by one...

Request Methods: You will find several types of Request methods in POSTMAN. Depending on your requirements or test you can choose one of them. Mainly you will be using four request methods in your application. These methods are given below...

- **GET Request:** To retrieve or fetch data
- **POST Request:** To create and update data
- **PUT Request;** To update data
- **DELETE Request:** For deleting data

Request URL: You will find a long-width bar in Postman where you will have to enter the URL to make the HTTP request.

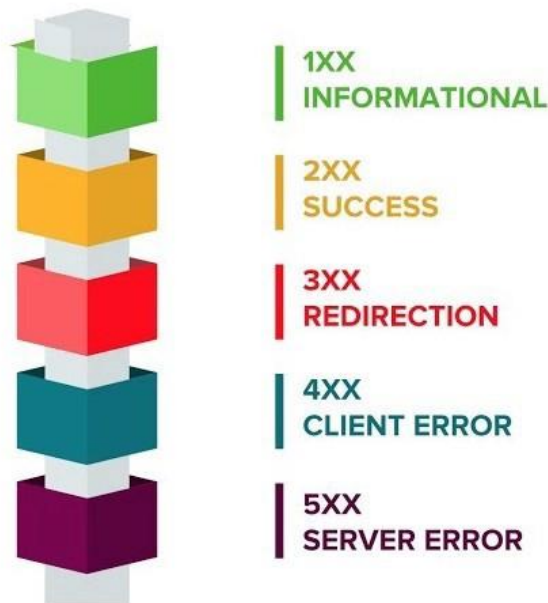
Request Headers: In the request header, you enter the key value of the application. The two main key values are given below.

- **Content-Type:** The format of data is specified by Content-Type. Mainly developers use JSON format in the content type.
- **Authorization:** This information is included to identify the requester.

Request Body: In Postman, you will get the tab of Body where you can mention some specific information that needs to be sent with the request. You will get the option to send the data either in raw, binary, or any other form. Most of the time you will select raw form. You will also get the option of Pre-request scripts. This gets executed before a request is sent. In Postman, you are also allowed to write and run the test for each request. You can use JavaScript language for this.

HTTP Response

HTTP Status Codes



Once you send the request to Postman, you get the response back from the API that contains Body, Cookies, Headers, Tests, Status Code, and API Response time. Body and Header get organized in different tabs. Status code gets displayed in another tab with the time taken to complete the API call. Some important status codes are given below to verify the response.

- **200**– For successful request.
- **201**– For successful request and data was created
- **204**– For Empty Response
- **400**– For Bad Request. Server is expecting something and Frontend is sending something else .
- **401**– For Unauthorized access. Authentication failed or the user does not have permission for the requested operation.

- **403**– For Forbidden, Access Denied (User is authenticated but unauthorized)
- **404**– For data not found.
- **405**– For method not allowed or requested method is not supported. Like you have created an api for post but you are request in the browser as get . You are giving wrong HTTP Method to access this api.
- **500**– Internal server error.
- **503**– For Service unavailable

Challenge 1 : Suppose you have one post mapping with api endpoint as /create_user and one get mapping with api endpoint as /get_user/id/{id} . What will be the response status code if in the browser you type the following in the url bar and press enter .

- (i) <http://localhost:8080/getusers> (HINT : 404)
- (ii) http://localhost:8080/create_user (HINT : 405)
- (iii) Where is the response status for API in chrome dev tools ?

Challenge 2 : Suppose the above API is defined in my controller class What will be the response status code if in the browser you type the following in the url bar and press enter ?

```
@GetMapping("/test2")
public String testFunc2(@RequestParam("name") String name){
    logger.info("Inside test2 func");
    return "Hi " + name + "!";
}
```

<http://localhost:8080/test2> (HINT : 400 as request param “name” is required by server but frontend is not giving it . It would return 200 for <http://localhost:8080/test2?name=mitsuha>)

How Postman Works?

Postman sends the request to the webserver and then the server sends the response back to it. A user has to set all the headers and cookies API expects to check the response.

You can install the postman from the link [Postman](#). This tool provides a collection of API calls, and you need to follow these API calls for testing APIs of the application. You will find a dropdown list with multiple methods.

CURL : Client Url for API Testing . (HomeWork)

<https://www.geeksforgeeks.org/sending-http-request-using-curl-set-1/>

Challenge 1 : What is an executable build in java ?

Challenge 2 : What is REST API ? What is Request Header , Response Header ?
(HINT : RequestHeaders are sent from frontend to backend .Response headers are sent from backend to frontend . Request headers contain more information about the resource to be fetched, or about the client requesting the resource. Response headers hold additional information about the response, like its location or about the server providing it. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>)

Challenge 3 : Can we use the POST method to implement fetch functionality ?

Challenge 4 : What is @RequestParam ?

HINT : @RequestParam annotation enables spring to extract input data that may be passed as a query, form data, or any arbitrary custom data. Here, we will see how we can use @RequestParam when building RESTful APIs for a web-based application.

2. A Simple Mapping

Let's say that we have an endpoint `/api/foos` that takes a query parameter called `id`.

```
@GetMapping("/api/foos")
@ResponseBody
public String getFoos(@RequestParam String id) {
    return "ID: " + id;
}
```

In this example, we used `@RequestParam` to extract the `id` query parameter.

A simple GET request would invoke `getFoos`:

```
http://localhost:8080/spring-mvc-basics/api/foos?id=abc
----
ID: abc
```

<https://www.geeksforgeeks.org/spring-mvc-requestparam-annotation/>

Simply put, the `@PathVariable` annotation can be used to handle template variables in the request URI mapping, and set them as method parameters.

Challenge 5 : What is @PathVariable ?

HINT :

2. A Simple Mapping

A simple use case of the `@PathVariable` annotation would be an endpoint that identifies an entity with a primary key:

```
@GetMapping("/api/employees/{id}")
@ResponseBody
public String getEmployeesById(@PathVariable String id) {
    return "ID: " + id;
}
```

In this example, we use the `@PathVariable` annotation to extract the templated part of the URI, represented by the variable `id`.

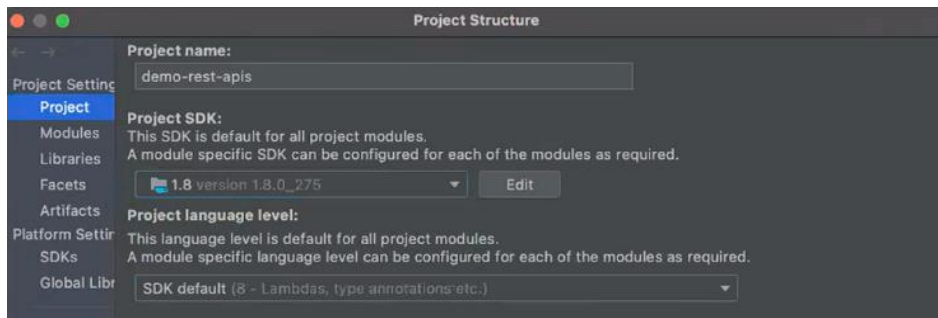
A simple *GET* request to `/api/employees/{id}` will invoke `getEmployeesById` with the extracted `id` value:

```
http://localhost:8080/api/employees/111
```

<https://www.baeldung.com/spring-pathvariable> IMP

Challenge 6 : How to change the Java version to be used in your project ?

HINT : File > Project Structure >



Challenge 7 : You have to implement a User Service .This means you can admit a new user into the database (use hashmap as a database), delete an existing user , update an existing user and fetch the details of an existing user from the database .

Challenge 8 : Can we have 2 api with the same endpoints? Consider the below two cases :

(i) Will the server start ?

```
@PostMapping("/create_user")
public void createUser(@RequestBody User user){
    userMap.put(user.getId(), user);
}

@PostMapping("/create_user")
public void createUser2(@RequestParam("id") int id,
    @RequestParam("name") String name,
    @RequestParam("age") int age){
    userMap.put(id, new User(id, name, age));
}
```

(ii)

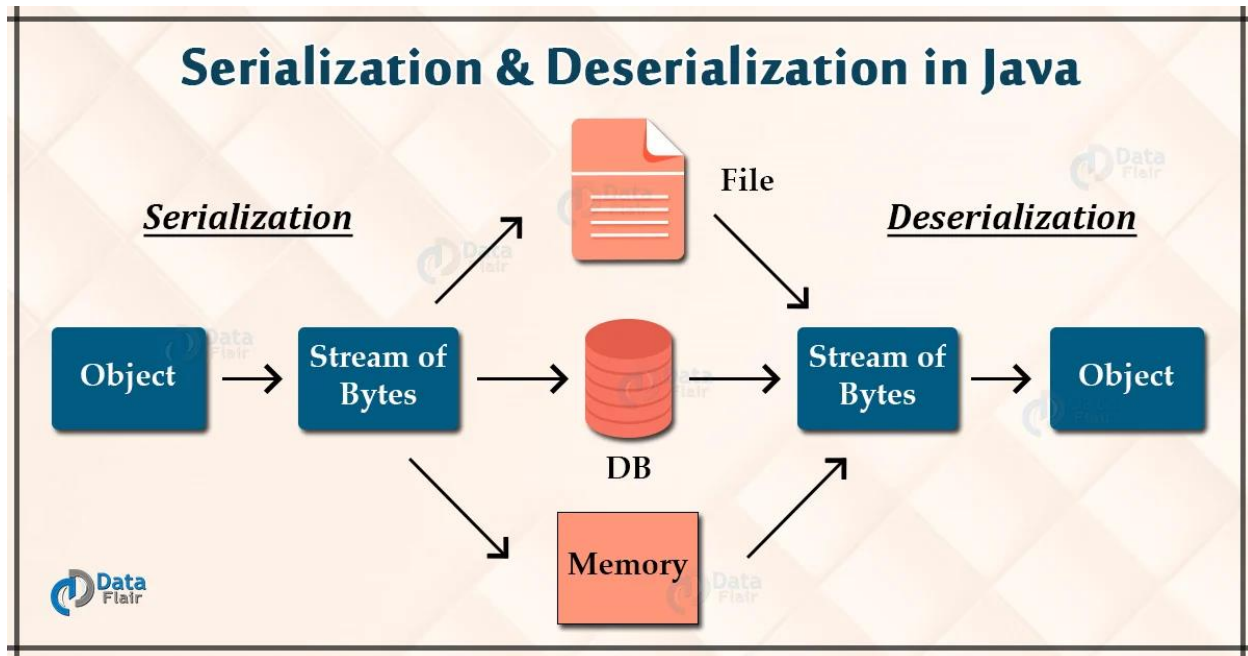
```
// http://localhost:8080/create_user
@PostMapping("/create_user")
public void createUser(@RequestBody User user){
    userMap.put(user.getId(), user);
}

@GetMapping("/create_user")
public void testFunc(){
}
```

HINT : To get exception , the api endpoint and the mapping should be the same . This is checked by @RestController .

Challenge 9 : When we return an object from @GetMapping method in Spring boot, how does it turn into json ? HINT : Serialization

Also, when we give json as @RequestBody but in spring boot we receive it as a java object. How ? HINT : Deserialization of JSON to Java objects done by embedded server .



\References :

<https://www.geeksforgeeks.org/why-rest-api-is-important-to-learn/>

<https://www.geeksforgeeks.org/best-coding-practices-for-rest-api-design/> IMP

<https://www.geeksforgeeks.org/postman-working-http-request-responses/> IMP

<https://www.baeldung.com/spring-request-param> IMP

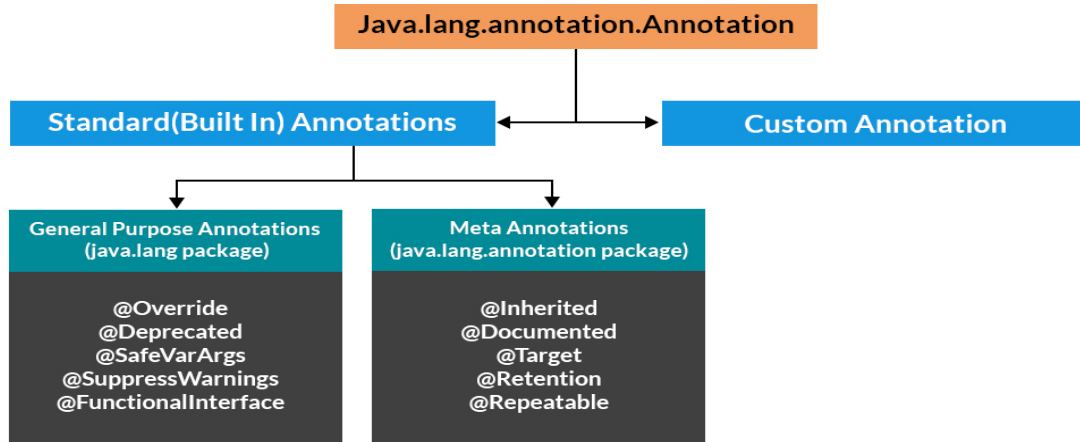
<https://www.baeldung.com/spring-requestparam-vs-pathvariable>

<https://apipheny.io/api-headers/> IMP

<https://www.geeksforgeeks.org/serialization-in-java/> IMP

<https://data-flair.training/blogs/serialization-and-deserialization-in-java/> IMP

@PostMapping is an annotation . What is the meaning of an annotation in spring boot : Annotations are used to provide supplemental information about a program.



- Annotations start with '@'.
- Annotations do not change the action of a compiled program.
- Annotations help to associate *metadata* (information) to the program elements i.e. instance variables, constructors, methods, classes, etc.
- Annotations are not pure comments as they can change the way a program is treated by the compiler. See below code for example.
- Annotations basically are used to provide additional information, so could be an alternative to XML and Java marker interfaces.

Challenge 1 : What is the significance of @RestController Annotation ?

HINT : 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. The @RestController annotation in Spring MVC/Spring BOOT is nothing but a combination of @Controller and @ResponseBody annotation.

<https://medium.com/@akshaypawar911/java-spring-framework-controller-vs-restcontroller-3ef2eb360917>

Challenge 2 : The functions inside the Controller class are non static .This means we must have created an object of the UserController class .When we invoke the api , functions mapped to it are called . Who creates this object ? Is there some rule that objects will be created for only certain classes ? Where can we see those objects ?

[IMP] HINT : Whenever a class is annotated with @Component directly or indirectly , Spring will create an object for you .

```

'.OrderedFormContentFilter      : Filter 'formContentFilter' configured for use
defaultListableBeanFactory      : Creating shared instance of singleton bean 'demoRestApisApplication'
defaultListableBeanFactory      : Creating shared instance of singleton bean 'UserController'
defaultListableBeanFactory      : Creating shared instance of singleton bean 'org.springframework.boot.autoconfigure.AutoConfiguratio
defaultListableBeanFactory      : Creating shared instance of singleton bean 'org.springframework.boot.autoconfigure.context.Property
defaultListableBeanFactory      : Creating shared instance of singleton bean 'org.springframework.boot.autoconfigure.websocket.servle
defaultListableBeanFactory      : Creating shared instance of singleton bean 'org.springframework.boot.context.properties.EnableConfi

```

We can verify if an object of our class is created by running the application in debug logging level and checking for “creating a shared instance of singleton bean <your_class>__”.

HW :

1. What is the meaning of shared instance ?
2. What is the meaning of singleton ?
3. What is the meaning of bean ?

Challenge 3 : @RequestBody vs @ResponseBody ?

HINT : @RequestBody : Converts JSON to java Object (Deserialization)

@ResponseBody : Converts Java Object to JSON and returns to client (Serialization)

Challenge 4 : Can we replace the @RestController annotation in the Controller class with @Component annotation ? HINT : No

Challenge 5 : Can we replace the @RestController annotation in the Controller class with @Controller and @ResponseBody annotation ? HINT : Yes

Challenge 6 : Can we replace the @RestController annotation in the Controller class with just @Controller annotation ? HINT : 500 Internal Server error

Challenge 7 : Can we annotate methods of Controller class with `@ResponseBody` rather than annotating it at the class level ? Where are the rules specified for each annotation whether its a class level or method level ?

HINT : Yes . Checkout the `@ResponseBody` annotation's interface . We can see `@Target({ElementType.TYPE, ElementType.METHOD})` .

References :

<https://www.geeksforgeeks.org/annotations-in-java/> IMP

<https://www.baeldung.com/spring-boot-annotations>

<https://lightrun.com/java/the-complete-list-of-spring-boot-annotations-you-must-know/>

<https://www.geeksforgeeks.org/spring-framework-annotations/>

<https://www.geeksforgeeks.org/spring-boot-annotations>

LOMBOK : Project Lombok is a java library that automatically plugs into your editor and build tools, spicing up your java. Never write another getter or equals method again, with one annotation your class has a fully featured builder, Automate your logging variables, and much more.

Please follow the given steps to install Lombok dependency which includes additional step of installing Lombok plugin to IDE other than adding to POM.xml :

For IntelliJ , please refer :

<https://www.baeldung.com/lombok-ide>

For STS , please refer :

<http://codeomitted.com/setup-lombok-with-stseclipse-based-ide/>

<https://www.javacodestuffs.com/2020/06/configure-lombok-in-eclipse.html>

In case , lombok is still not working after restarting IDE , refer the below :

<https://stackoverflow.com/questions/35842751/lombok-not-working-with-sts>

Challenge 1 : Where can I see my Getter and Setter functions added using Lombok ?

HINT : In the compiled classes inside target folder

Challenge 2 : After adding lombok as a dependency to POM.xml , why do we still need to install the Lombok plugin ?

HINT : Getters, Setters and all lombok Code is created fine only . We need to install the plugin in IDE so that IDE recognises the code created by Lombok .

Challenge 3 : If I publish my code written using Lombok , will the person who clones it be able to run it without the Lombok plugin installed in IDE ? HINT : No

Challenge 4 : How to use @Getter , @Setter , @NoArgsConstructor , @AllArgsConstructor and @Builder ?

Challenge 5 : How is Builder working internally ? HINT :
<http://www.javabyexamples.com/delombok-builder> (v imp)

Challenge 6 : Why do we need @Builder when we have a parameterised constructor ?

References :

<https://stackoverflow.com/questions/11881663/what-is-the-difference-in-maven-between-dependency-and-plugin-tags-in-pom-xml>

<https://medium.com/@alisnagina/how-to-install-lombok-in-sts-eclipse-on-window-10-2e2169b41636>

<http://www.javabyexamples.com/delombok-builder> V IMP

SPRING MVC : Spring MVC Framework follows the Model-View-Controller design pattern. It is used to develop web applications. It works around DispatcherServlet. DispatcherServlet handles all the HTTP requests and responses. It dispatches the requests to handlers. It uses @Controller and @RequestMapping as default request handlers. The @Controller annotation defines that a particular class is a controller. @RequestMapping annotation maps web requests to Spring Controller methods. The terms model, view, and controller are:

Model: The Model encapsulates the application data.

View: View renders the model data and also generates HTML output that the client's browser can interpret.

Controller: The Controller processes the user requests and passes them to the view for rendering.

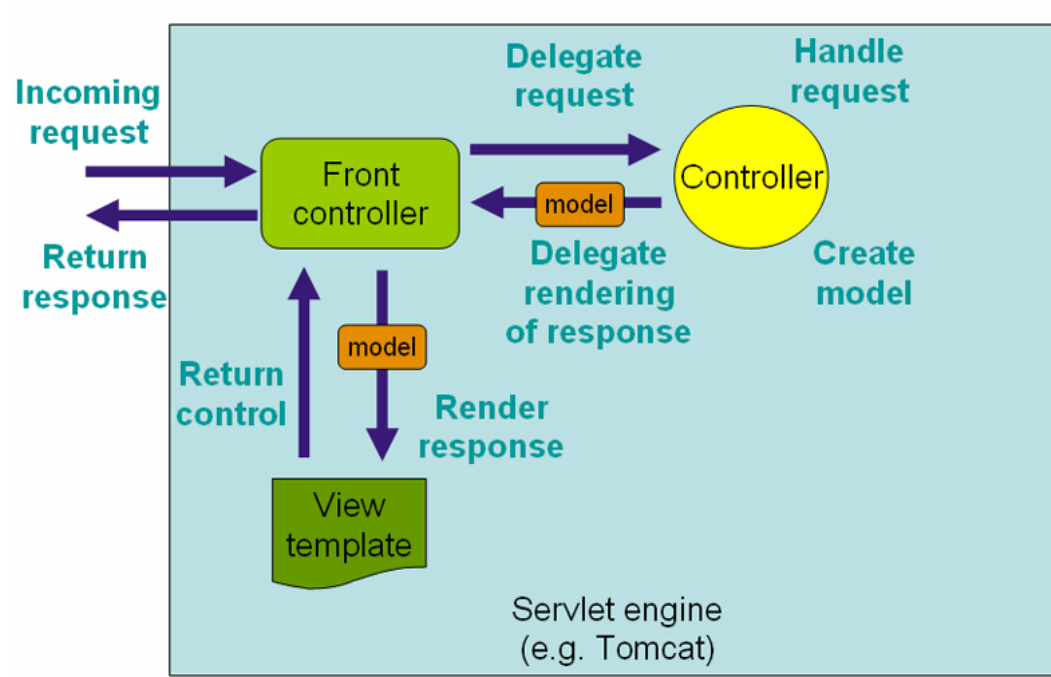
Spring MVC Framework works as follows:

1. All the incoming requests are intercepted by the DispatcherServlet that works as the front controller.
2. The DispatcherServlet then gets an entry of handler mapping from the XML file and forwards the request to the controller.
3. The object of ModelAndView is returned by the controller.
4. The DispatcherServlet checks the entry of the view resolver in the XML file and invokes the appropriate view component.

References :

<https://www.geeksforgeeks.org/spring-mvc-framework/>

<https://www.baeldung.com/spring-mvc-tutorial>



HomeWork :

Challenge 1 : What is Dispatcher Servlet ? Is it coming from spring boot or embedded server ?

Challenge 2 : What is the concept of the following in Java :

- (i) Static class (HINT : <https://www.geeksforgeeks.org/static-class-in-java/>)
- (ii) Inner Class (HINT : <https://www.geeksforgeeks.org/inner-class-java/>)
- (iii) Static inner class (HINT : <https://www.geeksforgeeks.org/java-program-to-illustrates-use-of-static-inner-class/>)
- (iv) Nested Class (HINT : <https://www.geeksforgeeks.org/nested-classes-java/>)
- (v) Difference Between Static and Non Static Nested Class in Java (HINT : <https://www.geeksforgeeks.org/difference-between-static-and-non-static-nested-class-in-java/>)

Challenge 3 : HashMap is stored in which memory ? Once the server stops , does hashmap persist our data ?

Challenge 4 : What do we mean when we say Spring Boot Servlet ?

HINT : A Servlet is a Java software component that extends the capabilities of a server.
<https://www.baeldung.com/spring-boot-dispatcherservlet-web-xml> IMP

