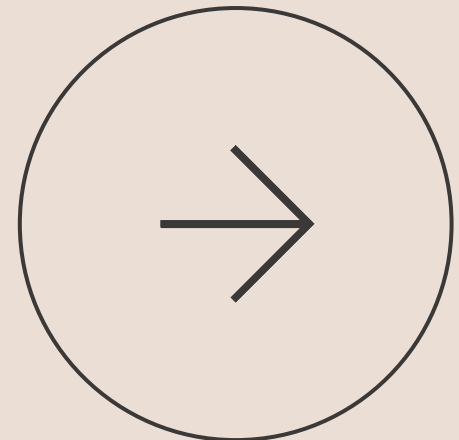


# Building RESTful Web Services

using Spring Boot

Part - 2

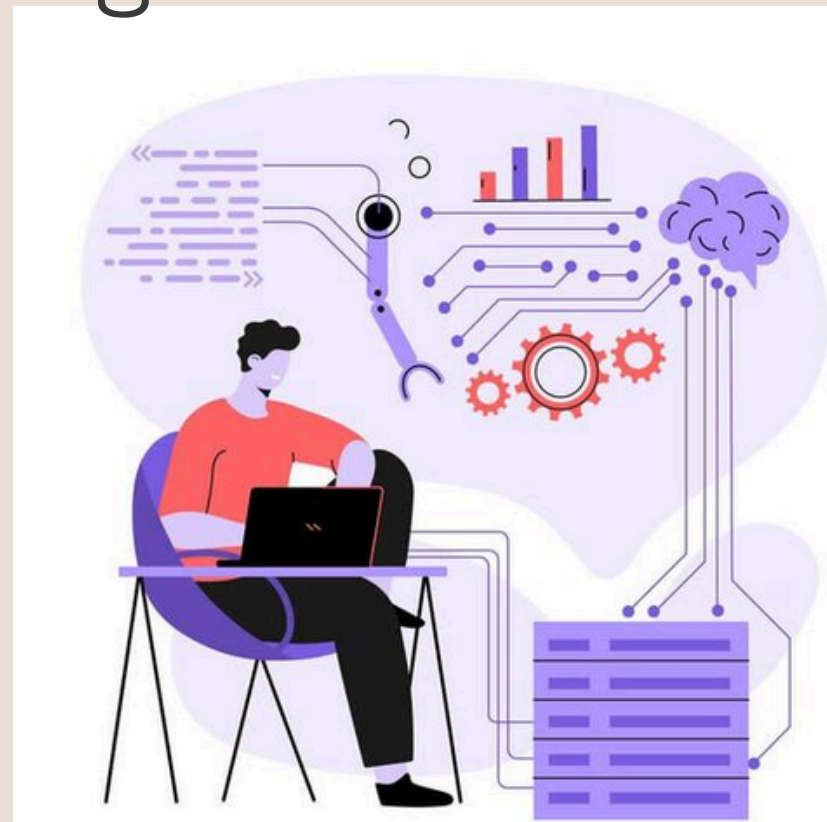


# HTTP Methods

1. **GET**: Used to retrieve information from the server.
2. **POST**: Used to create new resources on the server.
3. **PUT**: Used to update existing resources on the server.
4. **DELETE**: Used to delete resources from the server.

# Path Variables and Request Parameters

- **Path Variables:** Used to capture values from the URL itself.
- **Request Parameters:** Used to capture values from the query string.



# Created Api using Spring Boot:

This Spring Boot REST controller demonstrates how to handle different HTTP methods (GET, POST, PUT, DELETE) for managing items via API endpoints. The controller is mapped to the base URL /api/items and includes the following methods:

© ItemController.java x

```
1 package com.milap.myfirstProject;
2 import org.springframework.web.bind.annotation.*;
3 @RestController no usages
4 @RequestMapping("/api/items")
5 public class ItemController {
6     // GET request to retrieve an item by ID
7     @GetMapping("/{id}") no usages
8     public String getItem(@PathVariable("id") String id) {
9         return "Getting item with ID: " + id;
10    }
11    // POST request to create a new item
12    @PostMapping no usages
13    public String createItem(@RequestParam("name") String name) {
14        return "Creating item with name: " + name;
15    }
16    // PUT request to update an existing item by ID
17    @PutMapping("/{id}") no usages
18    public String updateItem(@PathVariable("id") String id, @RequestParam("name") String name) {
19        return "Updating item with ID: " + id + " to name: " + name;
20    }
21    // DELETE request to delete an item by ID
22    @DeleteMapping("/{id}") no usages
23    public String deleteItem(@PathVariable("id") String id) {
24        return "Deleting item with ID: " + id;
25    }
26 }
```

**Get:** Retrives Item by its ID

```
@GetMapping("/{id}")
public String getItem(@PathVariable("id") String id) {
    return "Getting item with ID: " + id;
}
```

**POST:** Creates a new item with a specified name.

```
@PostMapping
public String createItem(@RequestParam("name") String name) {
    return "Creating item with name: " + name;
}
```

**PUT:** Updates an existing item by its ID with a new name.

```
@PutMapping("/{id}") no usages
public String updateItem(@PathVariable("id") String id, @RequestParam("name") String name) {
    return "Updating item with ID: " + id + " to name: " + name;
}
```

**DELETE:** Deletes an item by its ID.

```
@DeleteMapping("/{id}")
public String deleteItem(@PathVariable("id") String id) {
    return "Deleting item with ID: " + id;
}
```

# Testing Endpoints:

- **GET:** <http://localhost:8080/api/items/1>
- **POST:** [http://localhost:8080/api/items?  
name=NewItem](http://localhost:8080/api/items?name=NewItem)
- **PUT:** [http://localhost:8080/api/items/1?  
name=UpdatedItem](http://localhost:8080/api/items/1?name=UpdatedItem)
- **DELETE:** <http://localhost:8080/api/items/1>

**TESTING  
MODE**



# Summary

- **GET:** Retrieve information using `@GetMapping` and `@PathVariable`.
- **POST:** Create a new resource using `@PostMapping` and `@RequestParam`.
- **PUT:** Update an existing resource using `@PutMapping`, `@PathVariable`, and `@RequestParam`.
- **DELETE:** Delete a resource using `@DeleteMapping` and `@PathVariable`.

# Follow for more updates



@milapeeeyyyy



@Milap-Magar