

|  |  |
| --- | --- |
| Format everething | cntrl + alt + l |
| Get methodes options | cntrl + space |

**What is ResponseEntity?**

* ResponseEntity is a Spring class used to send responses from the server to the client.
* It allows you to include **HTTP status codes**, **headers**, and **body** in the response.

**Step-by-Step Explanation of the Method:**

**1. Request Handling (@GetMapping("/list")):**

* This method is triggered when someone makes a GET request to /server/list.
* The /list endpoint is part of the /server route defined in @RequestMapping.

**2. Building the Response Body (Response.builder()):**

* **Response.builder()** is used to create a Response object.
* This object contains:
  + **timestamp(now())**: The current date and time when the response is created.
  + **data(Map.of("servers", serverService.getAllServers(30)))**: A Map where the key is "servers" and the value is the result of serverService.getAllServers(30). This is the actual data being sent to the client.
  + **message("Success")**: A message indicating the operation was successful.
  + **status(OK)**: HTTP status of the response (200 OK).
  + **statusCode(OK.value())**: The numeric value of the HTTP status (200).

**3. Wrapping in ResponseEntity.ok():**

* **ResponseEntity.ok(...)** creates a ResponseEntity object with:
  + A status code of 200 (OK).
  + The Response object built earlier as the body.

**What the Client Receives:**

When this method is called, the client gets a response like this (in JSON format):

json

Copy code

{

"timestamp": "2025-01-02T18:00:00",

"data": {

"servers": [

{

"id": 1,

"name": "Server A",

"status": "Active"

},

{

"id": 2,

"name": "Server B",

"status": "Inactive"

}

]

},

"message": "Success",

"status": "OK",

"statusCode": 200

}

**In Simple Terms:**

1. The method gets the list of servers from serverService.getAllServers(30).
2. It wraps the list in a Response object along with other information (timestamp, message, etc.).
3. It sends this Response object to the client as part of a ResponseEntity, which also includes an HTTP 200 status.

**servers" is Just One Key**

* In the Map.of("servers", serverService.getAllServers(30)), "servers" is a **single key** in the Map.
* Think of it like a label or a name that represents the **value** stored under it.

**The Value Can Contain Multiple Objects**

* The **value** associated with the key "servers" is the result of serverService.getAllServers(30).
* Typically, serverService.getAllServers(30) returns a **list of server objects** (e.g., List<Server>).

 **Response Generation:**

* The method creates a ResponseEntity<Response> object.
* ResponseEntity.ok(...) sets the HTTP status to 200 (OK).
* Inside ok(...), the Response.builder() constructs a Response object containing:
  + timestamp: The current time.
  + data: A Map with the key "servers" and a value from serverService.getAllServers(30) (likely a list of servers).
  + message: A string "Success".
  + status: The HTTP status OK (200).
  + statusCode: The numeric value of OK (200).

 **What the Client Receives:**

* The client receives a JSON response that looks something like this:

You need the **Angular CLI** to easily create and manage Angular applications. **CLI** stands for **Command-Line Interface**. It is a way to interact with your computer or a program by typing commands in a text-based window, like a terminal or command prompt.

|  |  |
| --- | --- |
| generate a ***new Angular project*** with a single command: | **ng new client-side**  **cd my-first-project**  **ng new quiz03**  **cd quiz03**  **ng serve** |

<https://www.youtube.com/watch?v=8ZPsZBcue50>

**1:17:22**