

17. ResponseEntity & Response Codes | 1xx, 2xx, 3xx, 4xx and 5xx Return Codes

Response Generally contains 3 parts:

Status Code: HTTP return code like 200 OK, 500 INTERNAL_SERVER_ERROR etc.

Header: Additional information (Optional)

Body: Data need to be sent in response

We can use "ResponseEntity<T>" to create and in this 'T' represents the type of the 'body'.

```
// @GetMapping("/getUserProfile")
@RequestMapping(method = RequestMethod.GET, path = "/getUserProfile")
public ResponseEntity<String> getUserProfile() {
    HttpHeaders headers = new HttpHeaders();
    headers.add("header1", "value1");
    headers.add("header2", "value2");
    return ResponseEntity.status(HttpStatus.OK)
        .headers(headers)
        .body("My response body Object can go here");
}
```

body should be last, actually it's kind of using Builder design pattern, so, status, 'headers' all are returning Builder object and 'body' method call return the ResponseEntity object.

```
ResponseEntity.status(HttpStatus.OK)
    .headers(headers)
    .build();
```

//If you don't want to return body then you can simply use .build() method it's internally set body to null

```
public <T> ResponseEntity<T> build() {
```

```

        return this.body((Object)null);
    }

    public <T> ResponseEntity<T> body(@Nullable T body) {
        return new ResponseEntity(body, this.headers, this.statusCode);
    }

    B headers(@Nullable HttpHeaders headers);

```

by-default 200 ok is the status code set:

- Here what happens if we are not providing ResponseEntity then Spring Boot Automatically wrap it into ResponseEntity.

```

@RestController
@RequestMapping(value="/api/")
public class UserController{
    @GetMapping("/save")
    public User newUser() {
        User resObj = new User("XYZ", 30);
        return resObj;
    }
}

```

@ResponseBody

- When we return Plan string or POJO directly from the class, then @ResponseBody annotation is required

Why?

It tells to considered value as Response Body and not as the view

But in previous example you are not putting @ResponseBody then how it's work

```

@RestController
@RequestMapping(value="/api/")
public class UserController{
    @GetMapping("/save")
    public User newUser() {
        User resObj = new User("XYZ", 30);
        return resObj;
    }
}

```

it's because, @RestController, automatically puts @ResponseBody to all the methods

```
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Controller
@ResponseBody
public @interface RestController {
    @AliasFor(
        annotation = Controller.class
    )
    String value() default "";
}
```

```
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Component
public @interface Controller {
    @AliasFor(
        annotation = Component.class
    )
    String value() default "";
}
```

```
@Controller
@RequestMapping(value="/api/")
public class UserController{
    @GetMapping("/save")
    // @ResponseBody
    public String newUser() {

        return "XYZ";
    }
}

//response
{
    "timestamp": "",
    "status": 404,
    "error": "Not Found",
    "path": "/api/save"
}
```

Response codes:

- 1XX [INFORMATIONAL]
- 2XX[SUCCESS]
- 3XX[REDIRECTION]
- 4XX[VALIDATION ERROR]
- 5XX[SERVER ERROR]

2XX [SUCCESS]

- Request received from client is received and processed successfully.


Status Code	Reason	Mostly used in	More Details
200	OK	GET, POST [Idempotent Calls] DUPLICATE REQUEST FOR NEWUSER CREATION NO DUPLICATE ROWS INSERTED	Request is successful and we are running the response body.
201	Created	POST	Request is successful and new resource is created.
202	Accepted	POST	Request is successfully accepted but processing is not yet completed. Batch processing like Export, Import etc.
204	NO Content	POST	Request is partial successful, say during Bulk Addition of 100 Users, 95 passed and 5 request failed, so, this response code can be used.

3XX [REDIRECTION]

- Client must take additional action to complete the request.

```
@RestController
@RequestMapping(value="/api/")
public class UserController{
    @GetMapping(path = "/old-get-user")
    public ResponseEntity<Void> getUser()
    {
        return ResponseEntity.status(HttpStatus.MOVED_PERMANENTLY)
            .header("Location", "/api/new-get-user").build();
    }

    @GetMapping(path = "/new-get-user")
    public ResponseEntity<String> getNewUser()
    {
        return ResponseEntity.status(HttpStatus.OK)
            .body("Success");
    }
}
```

 **http://localhost:8080/api/old-get-user** Save Share

GET

http://localhost:8080/api/old-get-user

Send

Params

Authorization

Headers (6)

Body

Scripts

Tests

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (5)

Test Results

200 OK

27 ms

170 B

🌐

🔍

...

Pretty

Raw

Preview

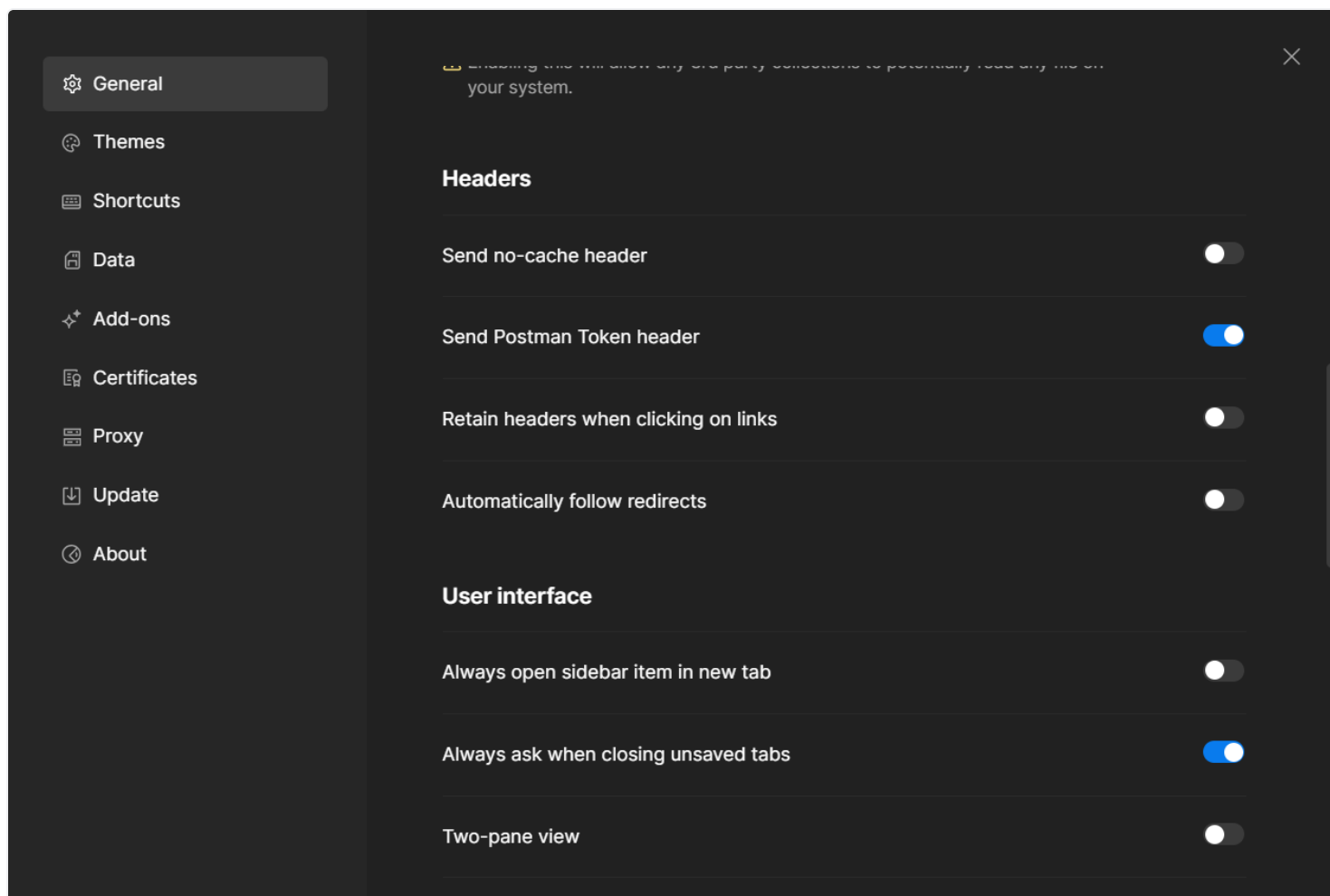
Visualize

Text

🔍

1 Success

But in postman you go to the setting
then turn it off the Automatic follow Redirection



After that you will get reponse somethong like this

The screenshot shows a web browser interface for testing HTTP requests. The URL bar displays `http://localhost:8080/api/old-get-user`. The method is set to `GET`. The response status is `301 Moved Permanently`. The 'Headers' tab is active, showing the following headers:

Key	Value
Location	/api/new-get-user
Content-Length	0
Date	Sun, 01 Dec 2024 06:37:22 GMT
Keep-Alive	timeout=60

Check for Location:=/api/new-get-user

Status Code	REASON	Mostly Used In	More details
301	Moved Permanently	When we migrate from Legacy API new API [OLD Status Code, new One is 308]	All request should directed to the new URL.
308	Perament Rediret	When we migrate from Legacy	Same as 301, but it dot not allow HTTP Method to change while redirect [for ex: if OLD API Call IS POST, THEN NEW API should also be POST , Which is relaxed in 301]
304	NOT MODIFIED	GET PATCH { TRY to avoid using it with PATCH for	<ol style="list-style-type: none"> Client makes a GET CALL, Server returned it with LAST-MODIFIED time in header. Client cache the response.

		<p>example: you trying to update a name of the user, but let's say name is already same in the DB, So no update required .</p> <p>In that case, we should not throw 304 [NOT_MODIFIED] error code, instead 204 [NO-CONTENT] OR 200 [OK] is more Appropriate.</p>	<ol style="list-style-type: none"> 3. Client make another GET CALL, pass this LAST MODIFIED time in "if-Modified since" header. 4. Server check the particular resource last update time with what client provided, if resources is not updated, server simply returns 304 [NOT_MODIFIED]. 5. let's last modified time of header is matched with last updated time of a row in DB 6. IF Modified, server process the request as usual and return the new value.
--	--	--	---

4XX [VALIDATION ERRORS]

- Client need to pass correct request to server.

Staus Code	Reason	Mostly used IN	More Details
400	BAD REQUEST	GET,POST,PATCH,DELETE	Client is not passing the required details to process the request.
401	UnAUTHORIZED	GET,POST,PATCH,DELETE	ANY API, Which require Authentication [LIKE Bearer token, Basic authentication etc..] and client try to access it without providing authentication details
403	Forbidden	GET,POST,PATCH,DELETE	Let say, only ADMIN Can perform certain operation. BUT IF API get invoked apart from ADMIN.

			We should throw 403 status code as clients apart from ADMIN do not have permission to access the resources.
404	NOT FOUND	GET,PATCH,DELETE	The requested resources which clients passes, is not found in DB by the server. for ex. GET the user details with ID: 123, but in DB THERE IS not such ID present.
405	Method NOT Allowed	GET,PATCH,DELETE,POST	EX: Hitting GET API, but with POST HTTP METHOD. In Spring BOOT, dispatcher servlet might throw this error, as control not even reach the controller
422	Un-processable Entity	GET,POST,PATCH,DELETE	Your application Business Validation: Like France User should not be allowed to open an account [as country is not supported yet]
429	Too Many Request	GET,POST,PATCH,DELETE	Let say: our rule is 1 user can max make 10 call in a minute. if user :12345 makes the 11th call in a minute then this 11th call should get failed and we can throw 429 error code.
409	Conflict	,PATCH,DELETE,POST	If one request is under in-progress

5XX [SERVER ERROR]

Request got failed at server, even though client passed the valid request. Means Something wrong at server.

Status Code	Reason	Mostly Used in	More Details
500	Internal SERVER Error	GET,POST,PATCH,DELETE	Generic error code when no more specific error code is suitable LIKE NULL POINTER EXCEPTION
501	Not Implemented	GET,POST,PATCH,DELETE	API Lacks the ability to fulfill the requests or say, API is the in development and in future it will be available.
502	BAD GATEWAY	GET,POST,PATCH,DELETE	<p>Server acting as a proxy and while calling upstream got invalid response.</p> <p>Example: My Application is deployed behind Reverse Proxy [NGINX]. If NGINX is not able to communicate with my application {because of misconfiguration of port number or something}, then it is eligible to thro 502 Bad GATEWAY</p>

1xx (Informational)

Interim response to communicate request progress or its status before processing the final request.

Status Code	Reason	Mostly used in	More details
100	Continue	POST	<p>Before sending the request, client check with server, if it can handle the request and ready:</p> <ol style="list-style-type: none">1. Client add few things in the header first, like:<ul style="list-style-type: none">- content length : 1048576- content type : multipart/form-data- Expect: 100-continue2. Server, checks that in header, 'Expect:100-continue' is present, means, client is just checking. So server validate everything (authentication, authorization, content type, length etc.)3. If Server is okay, it return 100 CONTINUE status code4. Client receives it and then invokes the API again without Expect and server process the request.