

//Basic ResponseEntity with Body

```
@GetMapping("/hello")

Public ResponseEntity<String> hello(){
    return ResponseEntity.ok("Hello world");
}
```

//Response Entity with Custom HTTP Status

```
@GetMapping("/not found")

Public ResponseEntity<String> notFound(){

Return new ResponseEntity<>("Resource not found", HttpStatus.NOT_FOUND);

}
```

//Response Entity with Custom Object DTO

```
@GetMapping("/user")

Public ResponseEntity<User> getUser(@PathVariable Long id){

User user = userService.getUserById(id);

    Return ResponseEntity.ok(user);

}
```

BASIC PRODUCT CRUD RESPONSE ENTITY RESPONSES EXAMPLE

Operation	Example(usage)
POST	<pre>@PostMapping("/create") public ResponseEntity<Product> saveProduct(@RequestBody Product product) { Product savedProduct = productService.saveProduct(product); return ResponseEntity.status(HttpStatus.CREATED).body(savedProduct); }</pre>

GET	<pre> @GetMapping("/all") public ResponseEntity<List<Product>> getAllProducts() { List<Product> allProducts = productService.getAllProduct(); return ResponseEntity.status(HttpStatus.OK).body(allProducts); } </pre>
GET	<pre> @GetMapping("/{id}") public ResponseEntity<Product> getProductById(@PathVariable Long id) { Product found = productService.getProductById(id); return ResponseEntity.status(HttpStatus.OK).body(found); } </pre>
PUT	<pre> @PutMapping("/{id}") public ResponseEntity<Product> updateProductById(@PathVariable Long id, Product updatedProduct) { return ResponseEntity.status(HttpStatus.OK).body(productService.updateProductById(id, updatedProduct)); } OR WITH CUSTOM STATUS CODE @PutMapping("/{id}") public ResponseEntity<Product> updateProductById(@PathVariable Long id, Product updatedProduct) { return ResponseEntity.status(HttpStatusCode.valueOf(200).body(productService.updateProductById (id,updatedProduct)); } </pre>
DELETE	<pre> @DeleteMapping("/{id}") public ResponseEntity<Product> deleteProductById(@PathVariable Long id) { productService.deleteProductById(id); return ResponseEntity.status(HttpStatus.OK).body(null); } OR WITH CUSTOM STATUS CODE @DeleteMapping("/{id}") public ResponseEntity<Void> deleteProductById(@PathVariable Long id) { productService.deleteProductById(id); return ResponseEntity.status(HttpStatusCode.valueOf(204).body(null); } </pre>

--	--

USAGE OF RESPONSE ENTITY WITH DTO

With DTO + Service Layer (Recommended)

UserDTO Class

```
Public class UserDTO{
    Private Long id;
    Private String name;
    Public UserDTO(User user){
        this.id = user.getId();
        this.name = user.getName();
        this.email = user.getEmail();
    }
}
```

UserService

@Service

```
Public class UserService{
```

@Autowired

```
    Private UserRepository userRepo;
```

```
    Public UserDTO getUserById(Long id){
        User user = userRepo.findById(id).orElseThrow(()-> new RuntimeException("User not found"));
        return new UserDTO(user);
    }
}
```

User Controller

@RestController

@RequestMapping("/users")

```
Public class UserController{
```

@Autowired

```
    Private UserService service;
```

@GetMapping("/{id}")

```
    Public ResponseEntity<UserDTO> getUser(@PathVariable Long id){
        UserDTO userDTO = service.getUserById(id);
        return ResponseEntity.ok(userDTO);
    }
}
```