

i2i Academy

Training Document

Topic	SWAGGER
Document Name	SWAGGER

Document Difficulty Level			
Beginner	Junior	Senior	Expert
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Copyright of  i2i Systems Turkey 2025

Document History

Date	Author	Ver	Comments
05.03.2025	Murat Kağan Temel	1.0	Initial Draft

SWAGGER

Exercise SWAGGER:

Definiton: Create a layered Spring Boot RESTful application that provides basic CRUD operations for managing Customer data. The application should include OpenAPI-compliant API documentation using the `springdoc-openapi-starter-webmvc-ui` dependency. This dependency also enables Swagger UI, so no additional configuration is required.

You may choose to simulate database operations using in-memory data structures such as Map or List. Using a real database is not mandatory.

The application should expose endpoints for the following operations:

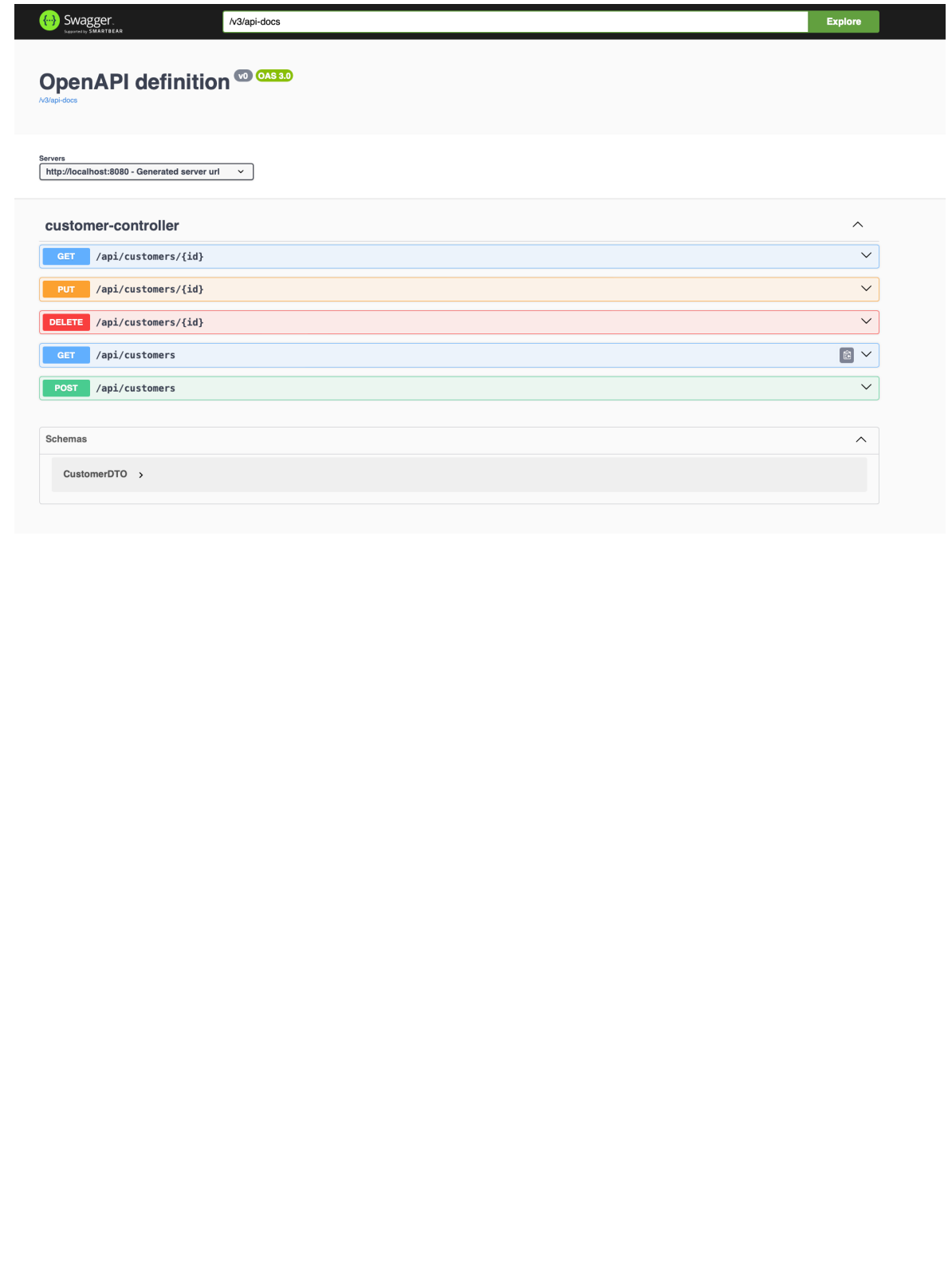
- **Create a new customer**
- **Retrieve a customer by ID**
- **Retrieve all customers**
- **Update an existing customer**
- **Delete a customer**

After completing the implementation, open the Swagger UI in your browser and share screenshots of the generated API documentation and the available endpoints. A sample CustomerDTO (DTO refers to Data Transfer Object) is shared in the next page.

```
public class CustomerDTO {  
  
    @Schema(description = "Unique ID of the customer", example =  
"1")  
  
    private Long id;  
  
  
    @NotBlank  
  
    @Schema(description = "Full name of the customer", example =  
"John Doe")  
  
    private String name;  
  
  
    @Email  
  
    @Schema(description = "Email address", example =  
"john.doe@example.com")  
  
    private String email;  
  
}
```

SWAGGER:

Your Answer:



The screenshot displays the Swagger UI interface for an OpenAPI definition. At the top, the Swagger logo is visible next to the text "Powered by SMARTPLAN". A search bar contains the text "v3/api-docs", and an "Explore" button is located to its right. Below the search bar, the title "OpenAPI definition" is shown with a "v0" badge and a "OAS 3.0" badge. Underneath, the "Servers" section contains a dropdown menu with the value "http://localhost:8080 - Generated server url". The main content area is titled "customer-controller" and features a list of API endpoints. Each endpoint is represented by a colored bar with a verb (GET, PUT, DELETE, GET, POST) and a path. The endpoints are: GET /api/customers/{id}, PUT /api/customers/{id}, DELETE /api/customers/{id}, GET /api/customers, and POST /api/customers. Below the endpoints, the "Schemas" section is visible, showing a single schema named "CustomerDTO".