**WEB API -1:**

**RESTful Web Service**

* **REST (Representational State Transfer)** is an architectural style for designing networked applications.
* REST uses standard HTTP methods like GET, POST, PUT, DELETE.
* RESTful services are **stateless** and communicate via **resources** (usually represented in JSON/XML).

**Features of REST Architecture**

* **Stateless**: Each request from client to server must contain all the information.
* **Representation**: Resources are represented in formats like JSON, XML.
* **Messages**: Client sends HTTP requests and gets responses.
* **Cacheable**: Responses can be cached to improve performance.
* **Uniform Interface**: Same rules and structure for all resources.

**Web API vs Web Service**

| **Feature** | **Web API** | **Web Service** |
| --- | --- | --- |
| Protocol | HTTP | SOAP (Mostly) |
| Data Format | JSON, XML, etc. | XML only |
| Lightweight | Yes | No |
| Speed | Faster | Slower |
| Supports REST | Yes | No (Only SOAP-based) |

**Web API**

* Web API is a framework for building RESTful services on the .NET platform.
* In .NET Core, Web API is part of ASP.NET Core itself.

**Microservice**

* An **architecture style** where the app is broken into small independent services.
* Each microservice:
  + Focuses on a single business capability
  + Is independently deployable
  + Communicates over lightweight protocols like HTTP/REST

**2. What is HttpRequest & HttpResponse**

**HttpRequest**

* Sent from client to server.
* Contains:
  + HTTP Method (GET, POST, etc.)
  + Headers (Authorization, Content-Type)
  + Body (JSON data for POST/PUT)
  + URL

**HttpResponse**

* Sent by the server to client.
* Contains:
  + Status Code (200, 400, 500, etc.)
  + Body (usually JSON)
  + Headers

**3. Types of Action Verbs (HTTP Methods)**

| **Verb** | **Meaning** | **Usage in Web API** |
| --- | --- | --- |
| HttpGet | Fetch data | [HttpGet] |
| HttpPost | Create new data | [HttpPost] |
| HttpPut | Update existing data | [HttpPut("{id}")] |
| HttpDelete | Delete a resource | [HttpDelete("{id}")] |

Each is declared in controller like:

csharp

CopyEdit

[HttpGet]

public IActionResult GetData() { ... }

[HttpPost]

public IActionResult AddData([FromBody] DataModel data) { ... }

**4. HTTP Status Codes in Web API**

| **Code** | **Name** | **Meaning** |
| --- | --- | --- |
| 200 | OK | Success |
| 400 | BadRequest | Validation or format issue |
| 401 | Unauthorized | No/invalid token or authentication |
| 500 | InternalServerError | Server crashed during processing |

**Common Action Result Methods:**

csharp

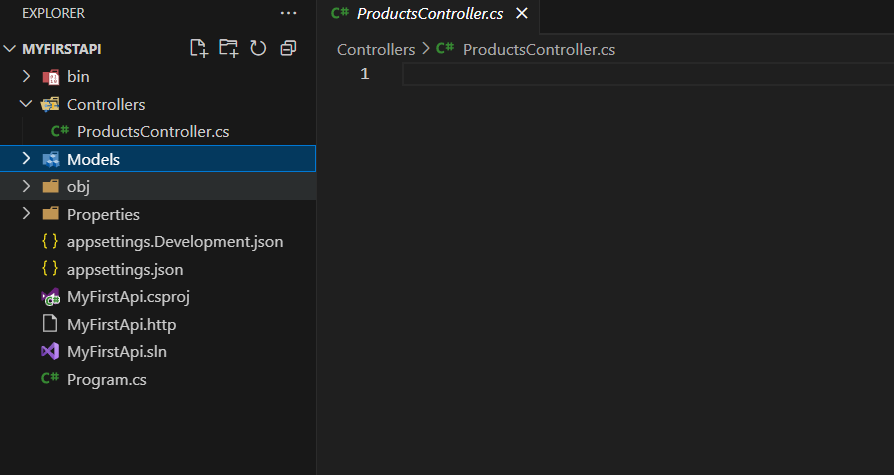
CopyEdit

return Ok(data); // 200

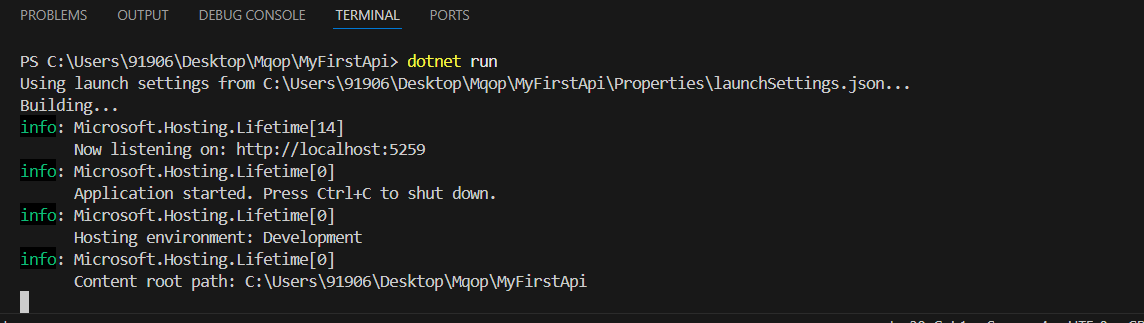
return BadRequest("Invalid"); // 400

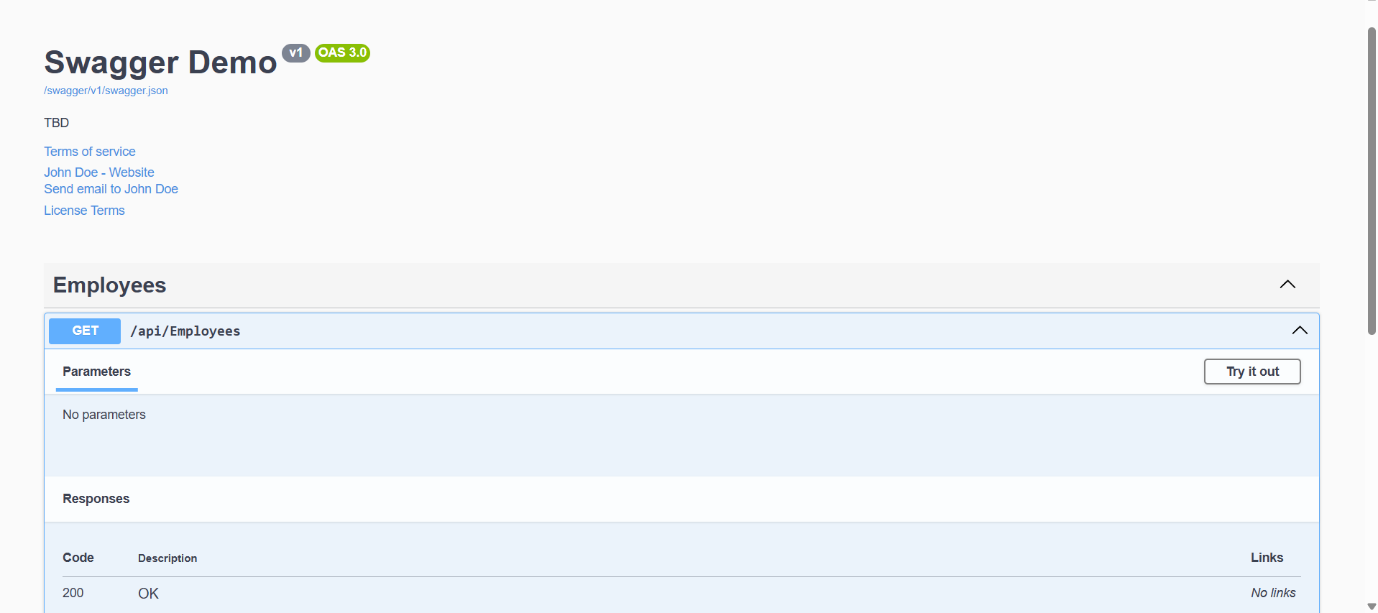
return Unauthorized(); // 401

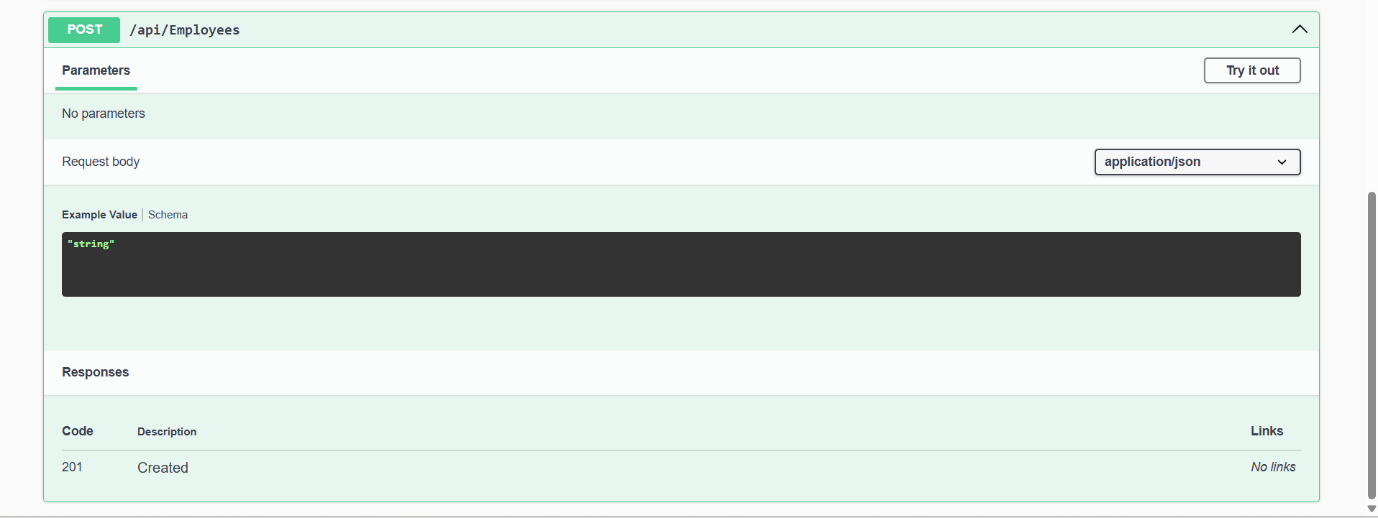
return StatusCode(500, "Error");// 500



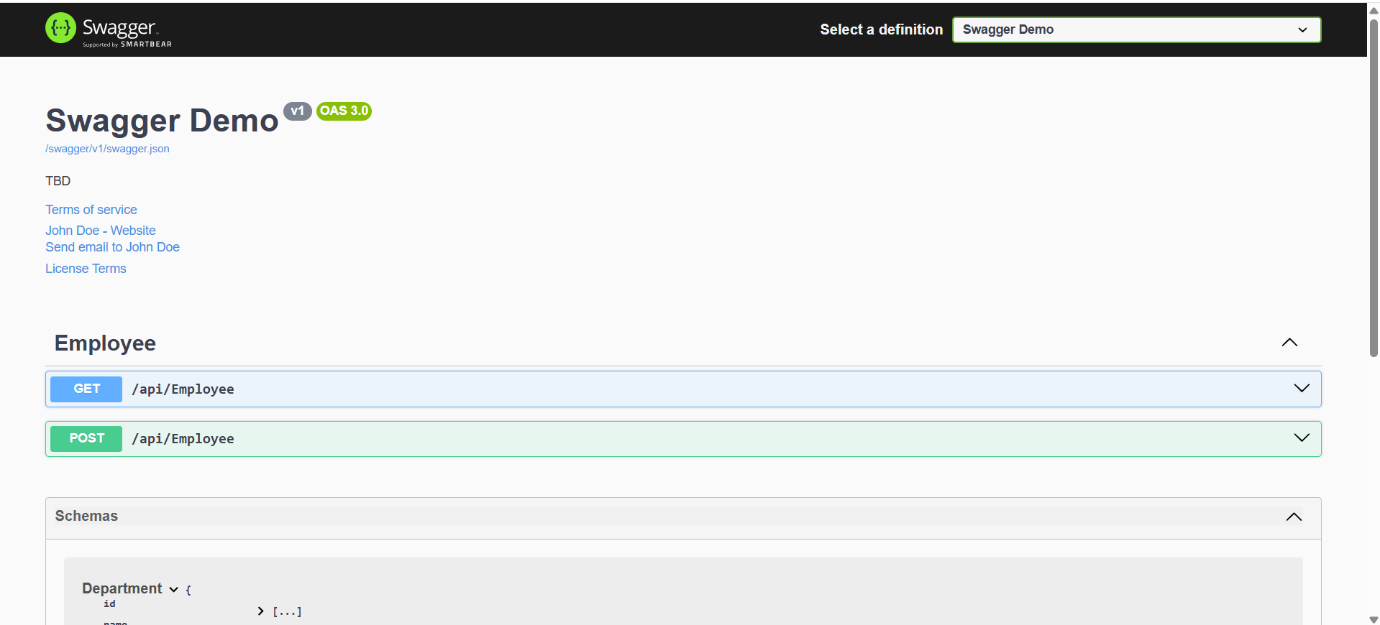
**WEB API -2:**

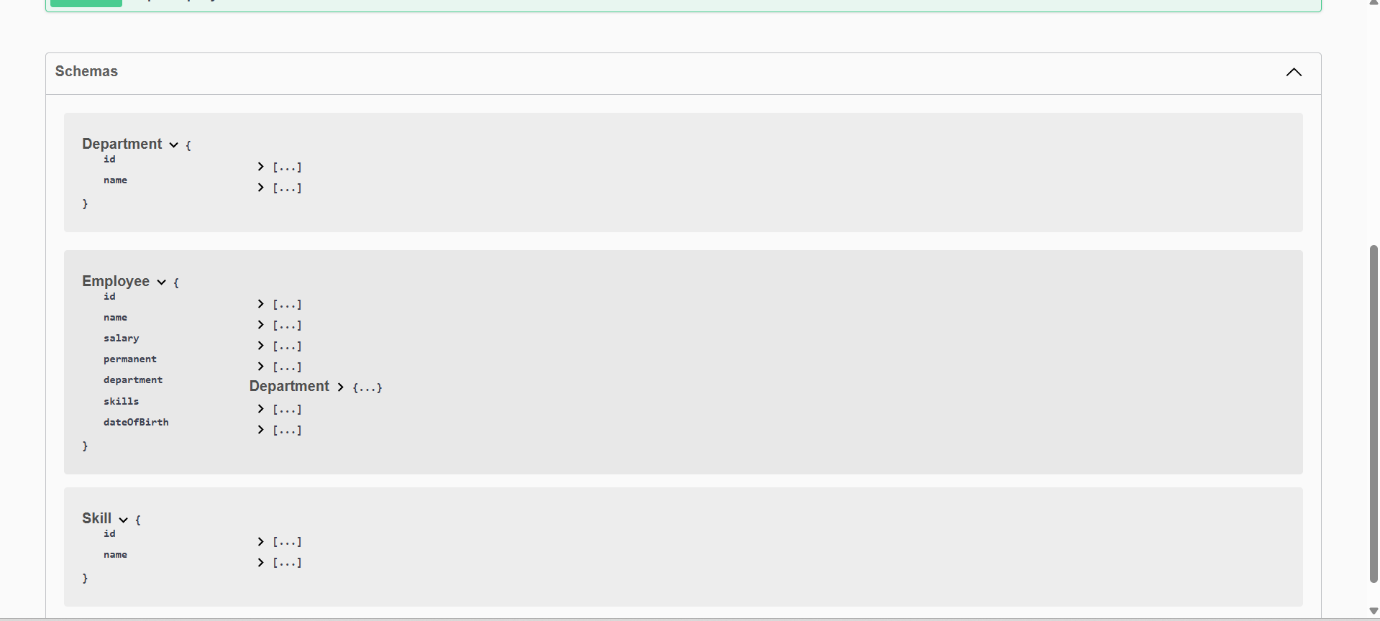




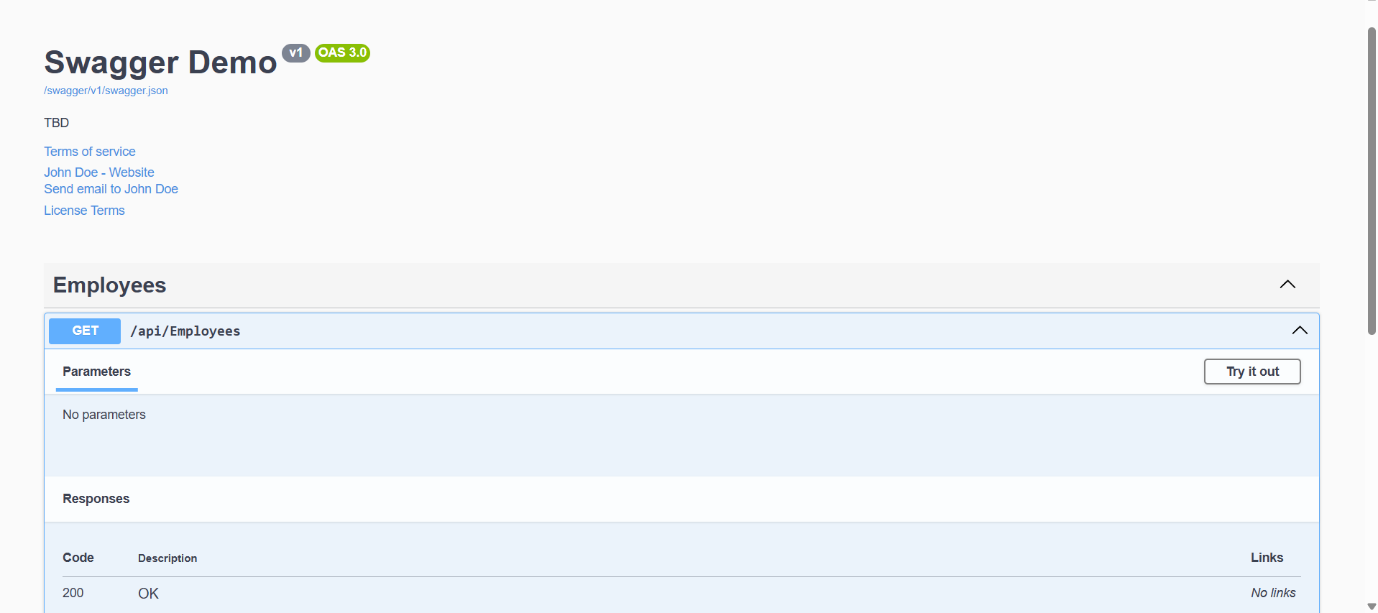


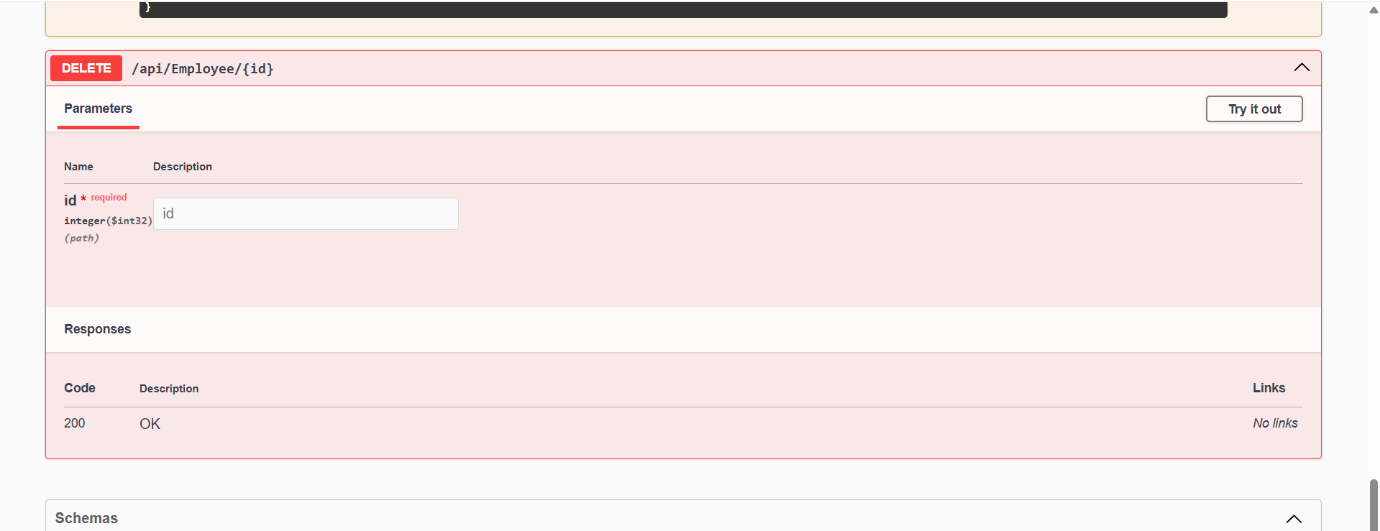
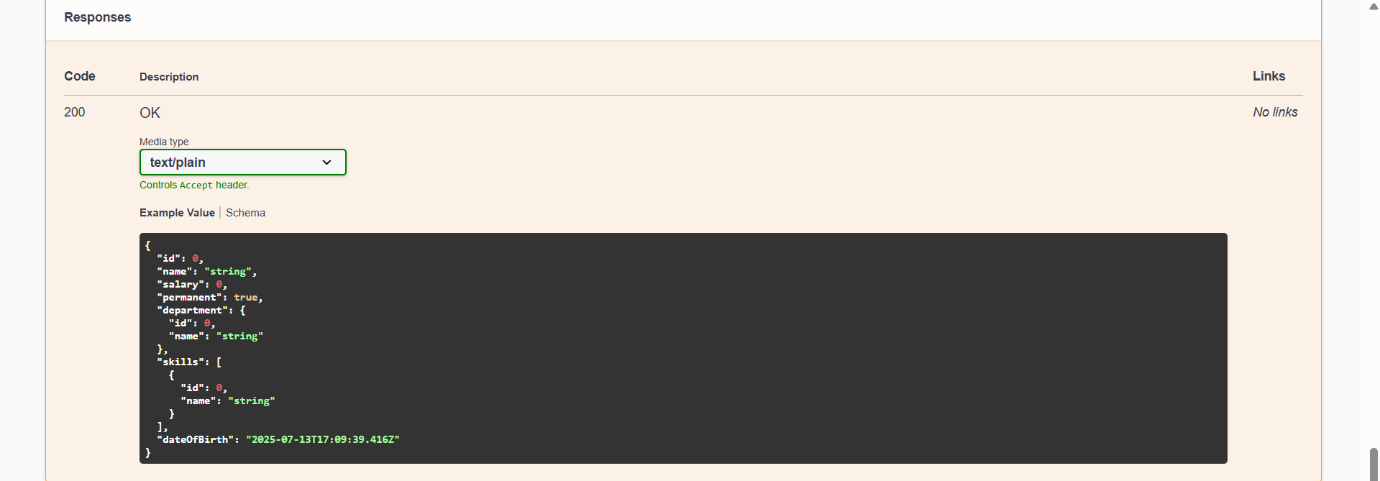
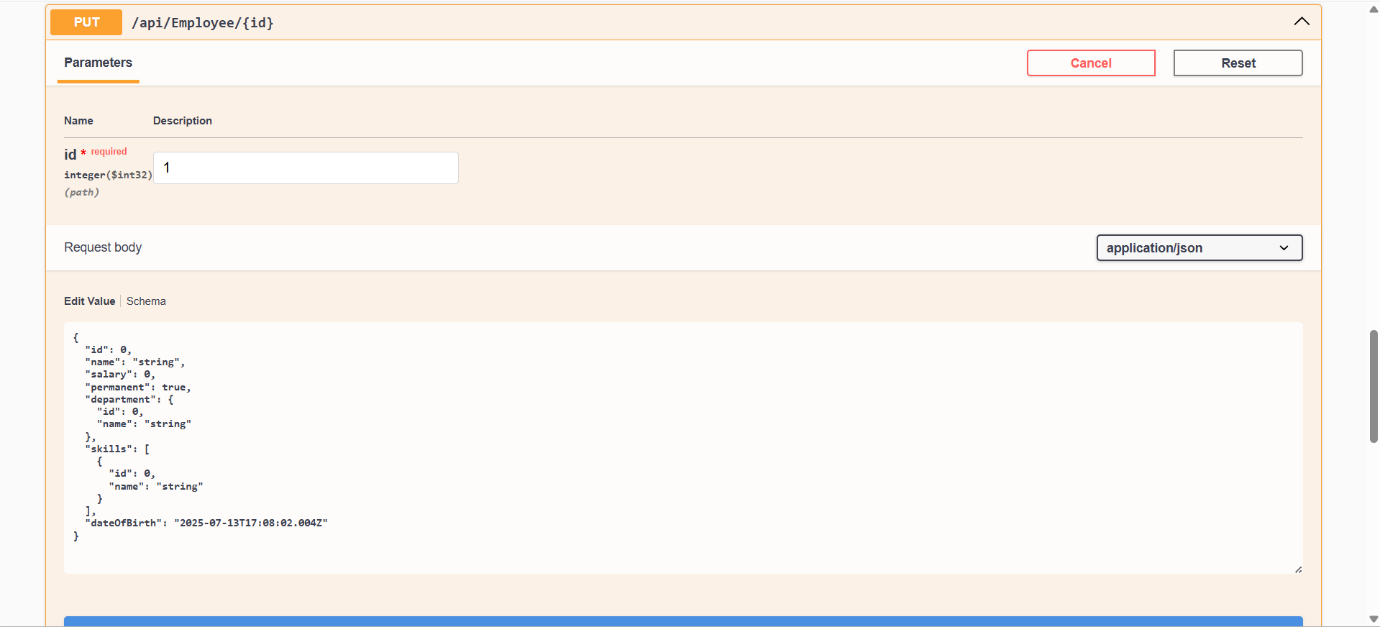
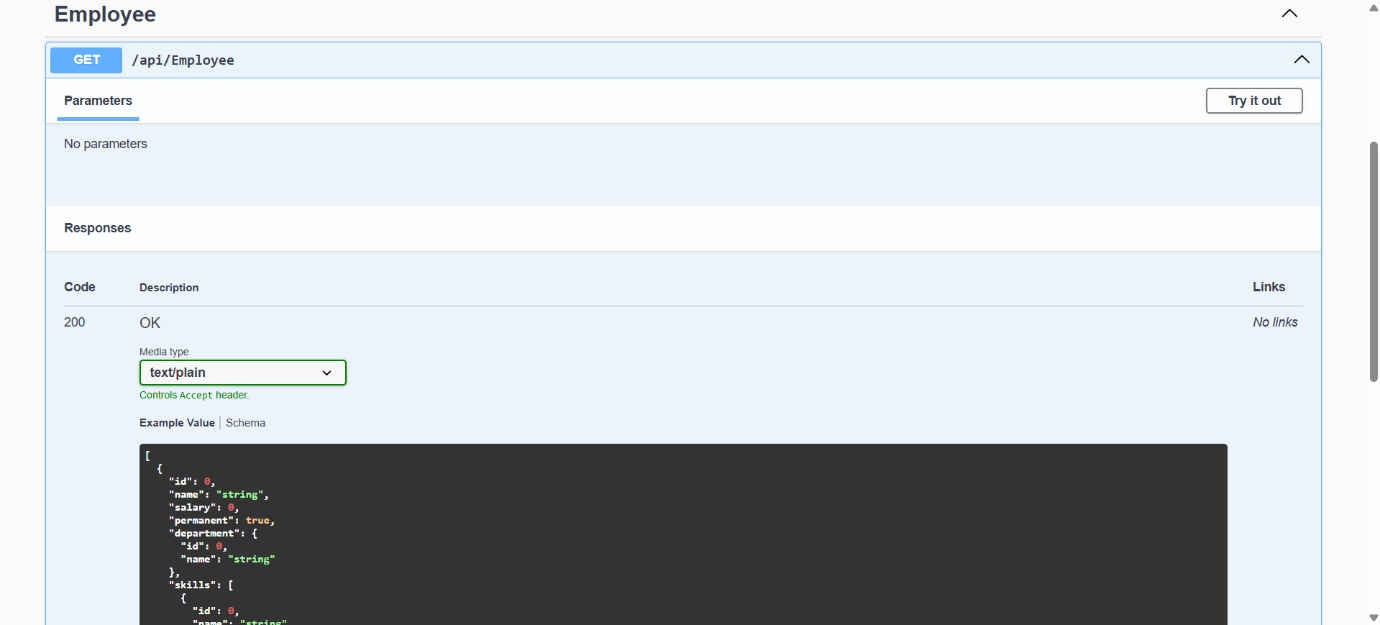
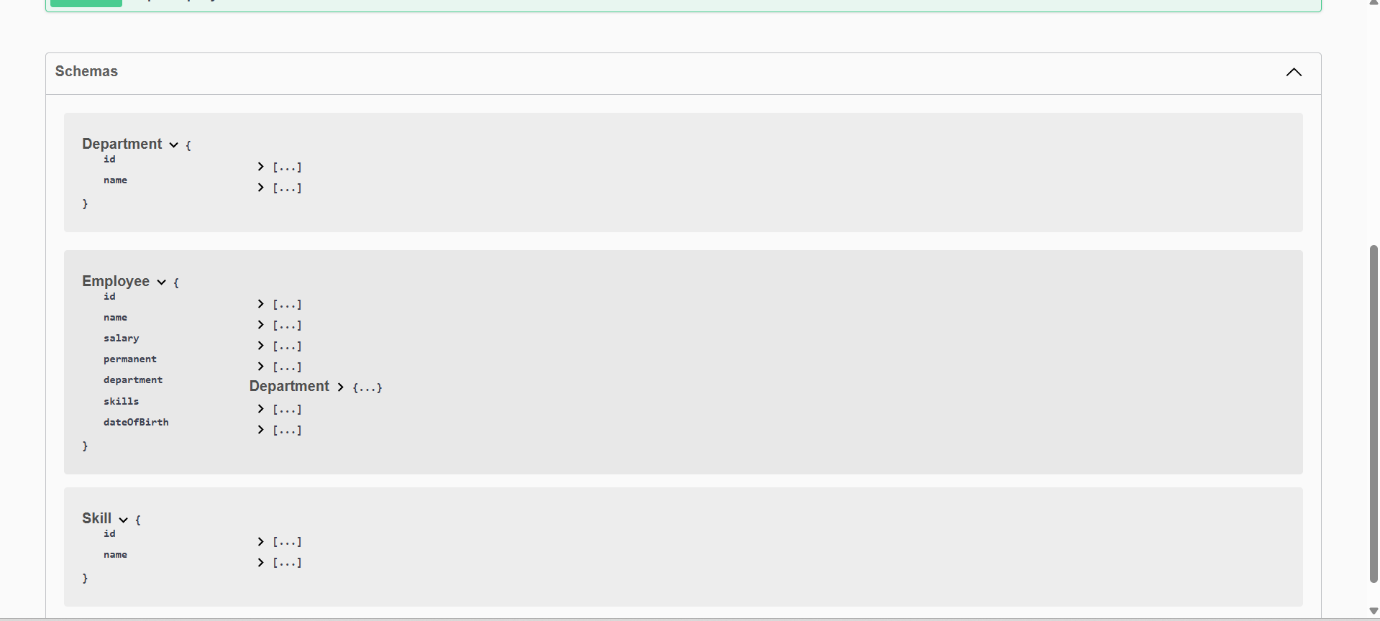
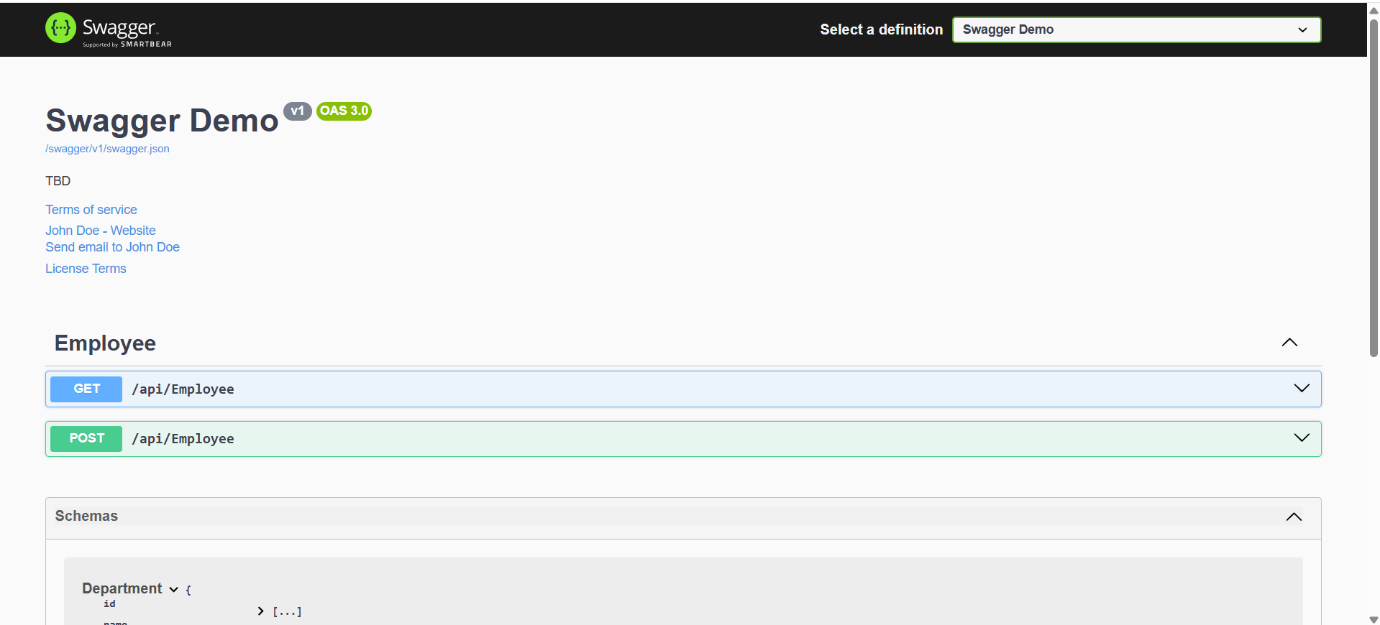
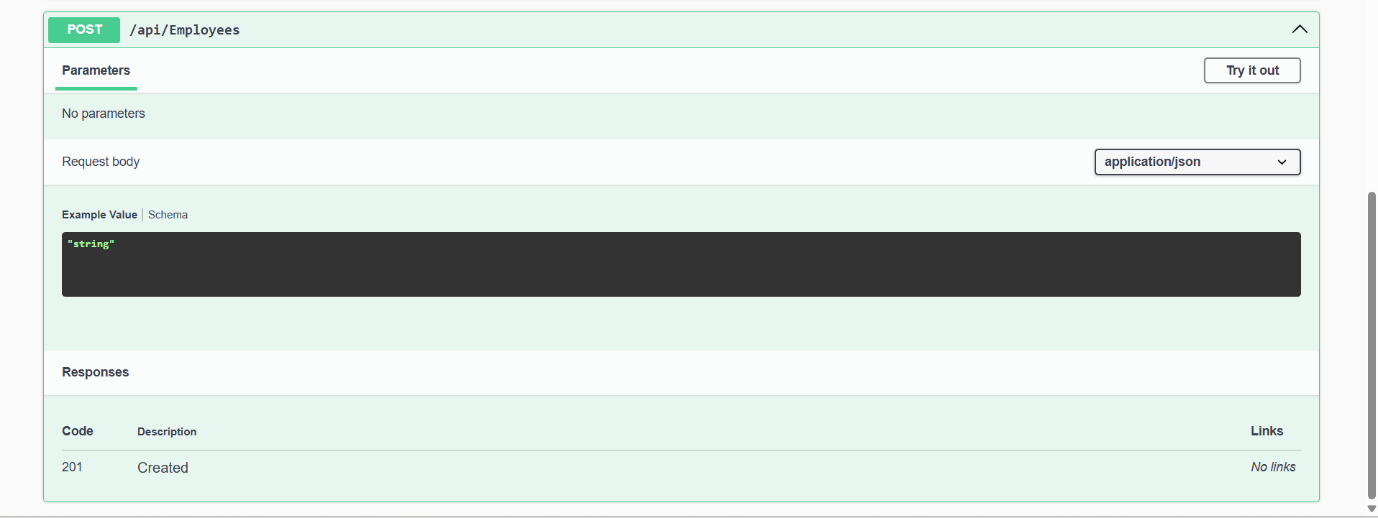
**WEB API -3:**





**WEB API -4:**

****



**WEB API -5:**

