Out of office

Tech Stack:
.net core 8.0 web api
Microsoft.AspNetCore.OpenApi
SwashbuckleMicrosoft.AspNetCore
Microsoft.AspNetCore.Authentication.JwtBearer
Microsoft.AspNetCore.IdentityFrameworkCore
Microsoft.EntityFrameworkCore.Design
Microsoft.EntityFrameworkCore.Sqlite
SQLite

To realize this task was created backend application using web api technology and based on endpoints

After launching the application, the browser automatically opens a tab with all endpoints at the address:

http://localhost:5283/swagger/index.html

Create an SQL database with the above tables and fields.

The created database is inside the project:

out_of_office/out_of_office.db

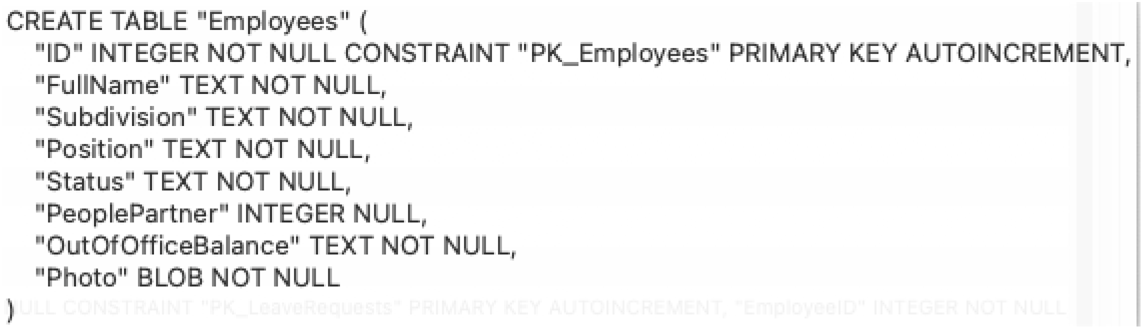
Also the automatically created code to create the database is inside the project:

out_of_office/Migrations/

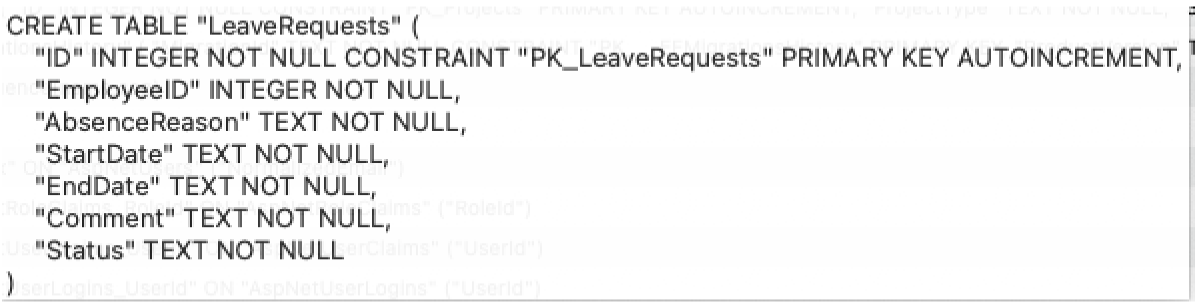
SQLite was chosen to create the database because of its ease of use, easy integration into the project.

Create migrations and database, after creating the necessary models console command:
dotnet ef migrations add InitialCreate
dotnet ef database update

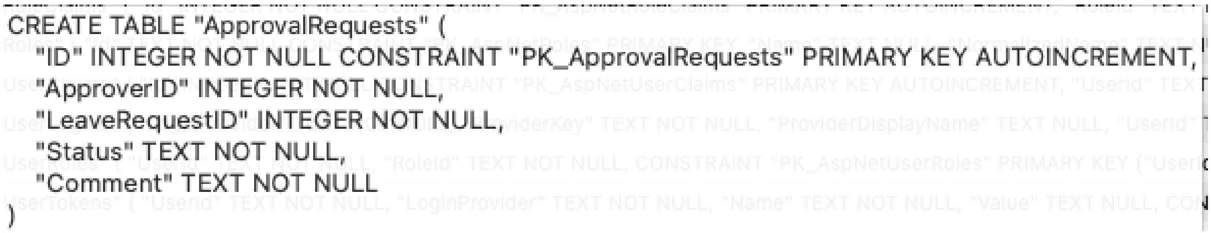
Sql tables were created automatically, however from dbms we can see the commands to create these schemas:



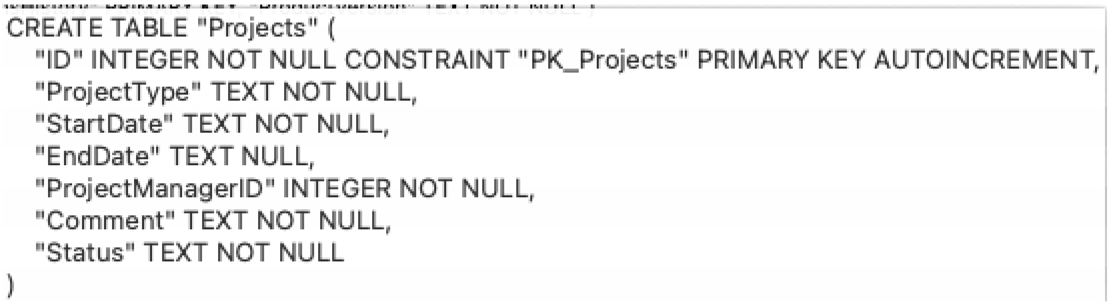
```
CREATE TABLE "Employees" (
    "ID" INTEGER NOT NULL CONSTRAINT "PK_Employees" PRIMARY KEY AUTOINCREMENT,
    "FullName" TEXT NOT NULL,
    "Subdivision" TEXT NOT NULL,
    "Position" TEXT NOT NULL,
    "Status" TEXT NOT NULL,
    "PeoplePartner" INTEGER NULL,
    "OutOfOfficeBalance" TEXT NOT NULL,
    "Photo" BLOB NOT NULL
)
```



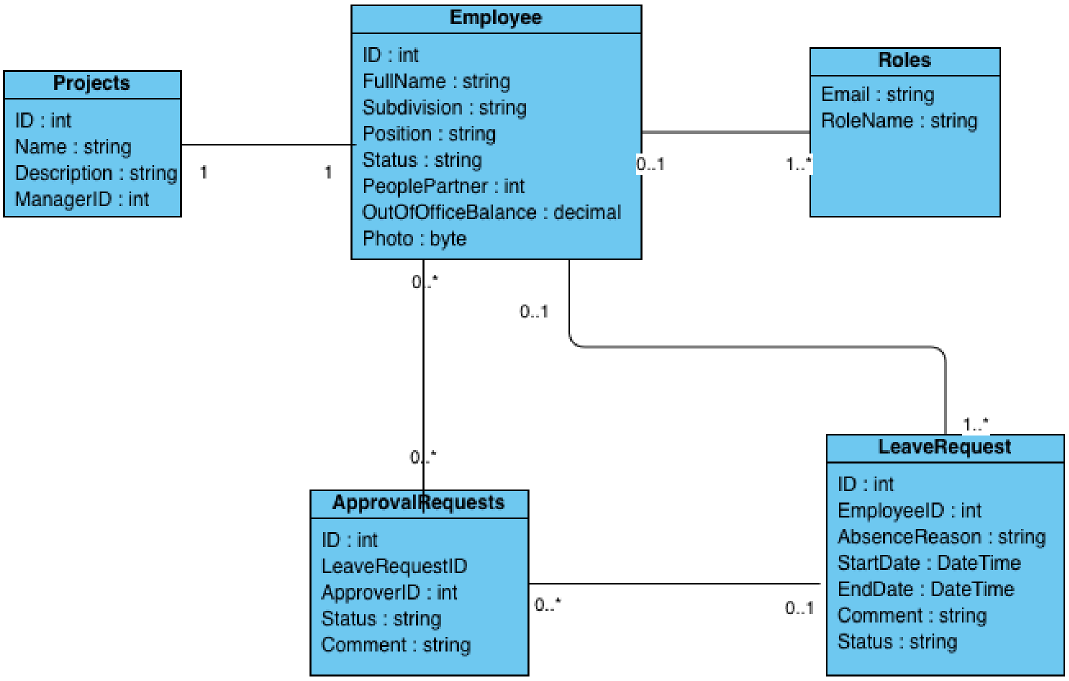
```
CREATE TABLE "LeaveRequests" (
    "ID" INTEGER NOT NULL CONSTRAINT "PK_LeaveRequests" PRIMARY KEY AUTOINCREMENT,
    "EmployeeID" INTEGER NOT NULL,
    "AbsenceReason" TEXT NOT NULL,
    "StartDate" TEXT NOT NULL,
    "EndDate" TEXT NOT NULL,
    "Comment" TEXT NOT NULL,
    "Status" TEXT NOT NULL
)
```



```
CREATE TABLE "ApprovalRequests" (
    "ID" INTEGER NOT NULL CONSTRAINT "PK_ApprovalRequests" PRIMARY KEY AUTOINCREMENT,
    "ApproverID" INTEGER NOT NULL,
    "LeaveRequestID" INTEGER NOT NULL,
    "Status" TEXT NOT NULL,
    "Comment" TEXT NOT NULL
)
```



```
CREATE TABLE "Projects" (
    "ID" INTEGER NOT NULL CONSTRAINT "PK_Projects" PRIMARY KEY AUTOINCREMENT,
    "ProjectType" TEXT NOT NULL,
    "StartDate" TEXT NOT NULL,
    "EndDate" TEXT NULL,
    "ProjectManagerID" INTEGER NOT NULL,
    "Comment" TEXT NOT NULL,
    "Status" TEXT NOT NULL
)
```

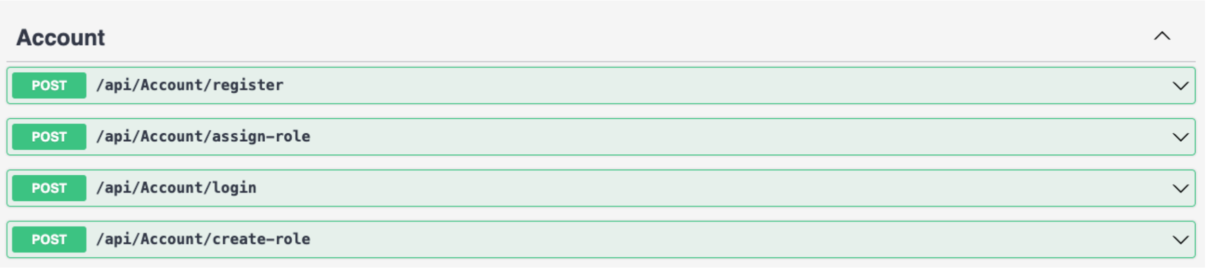


Create an application that will interact with the database.

Project link:
https://github.com/mavlevich/out_of_office

Due to the fact that in the project requirements we needed to use user identification to test the functionality of the application, Postman will be used

Endpoints for user registration and login, role creation and role attribution:



Account

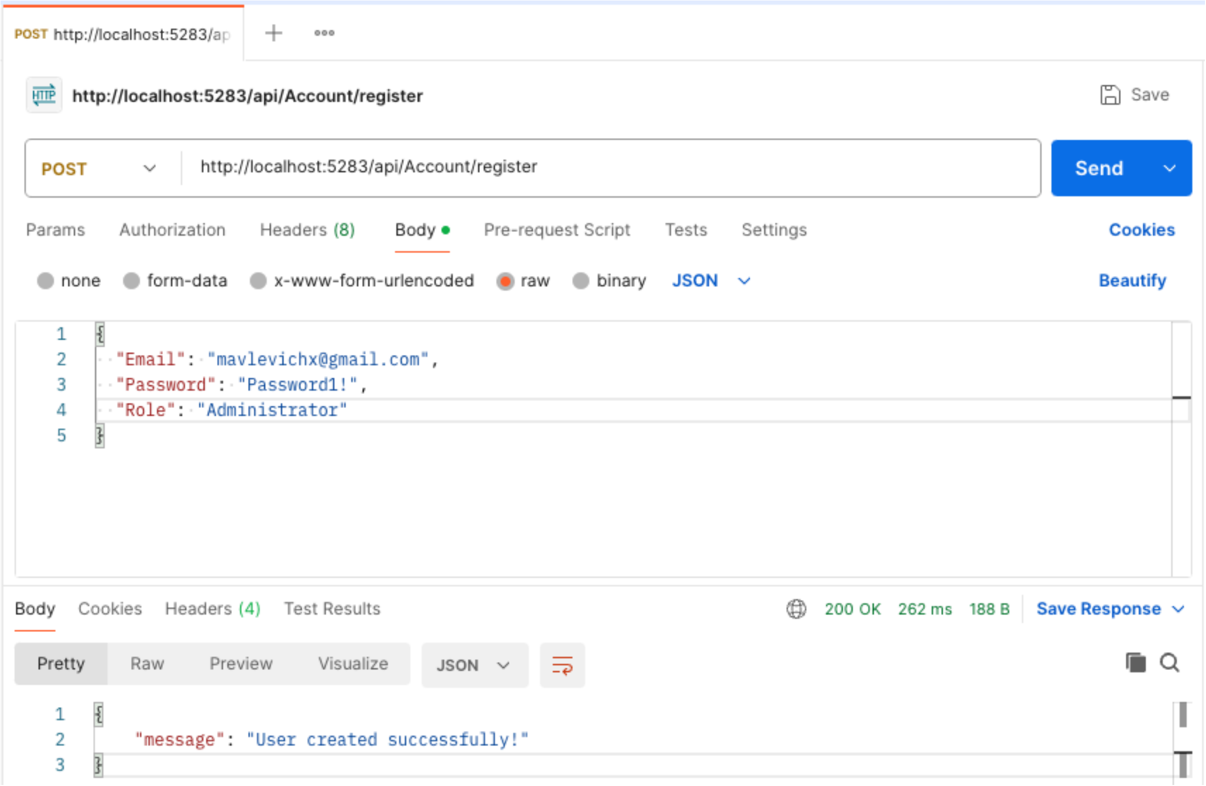
POST /api/Account/register

POST /api/Account/assign-role

POST /api/Account/login

POST /api/Account/create-role

User registration:



http://localhost:5283/api/Account/register

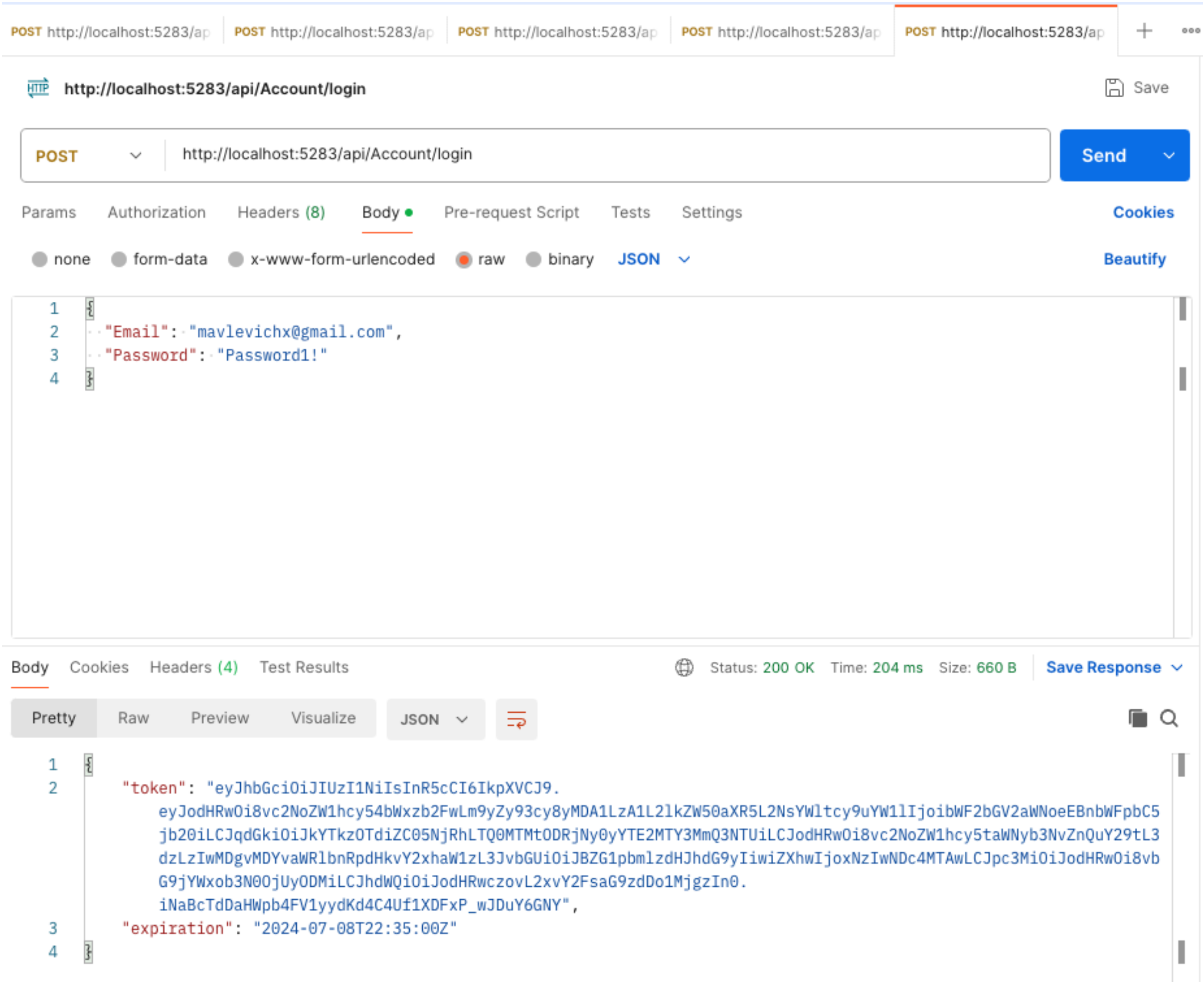
POST http://localhost:5283/api/Account/register

```
{
  "Email": "mavlevichx@gmail.com",
  "Password": "Password1!",
  "Role": "Administrator"
}
```

Body — 200 OK 262 ms 188 B — Pretty JSON

```
{
    "message": "User created successfully!"
}
```

User login:



http://localhost:5283/api/Account/login

POST http://localhost:5283/api/Account/login

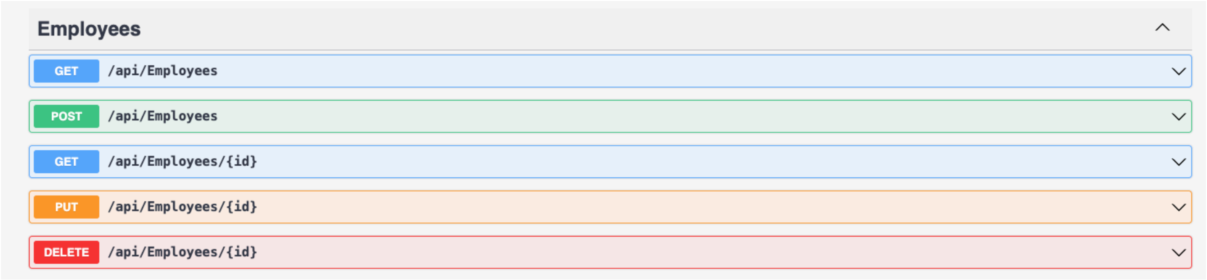
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON

```
{
  "Email": "mavlevichx@gmail.com",
  "Password": "Password1!"
}
```

Body — Status: 200 OK Time: 204 ms Size: 660 B — JSON

```
{
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoibWF2bGV2aWNoeEBnbWFpbC5
        jb20iLCJqdGkiOiJkYTkzOTdiZC05NjRhLTQ0MTMtODRjNy0yYTE2MTY3MmQ3NTUiLCJodHRwOi8vc2NoZW1hcy5taWNyb3NvZnQuY29tL3
        dzLzIwMDgvMDYvaWRlbnRpdHkvY2xhaW1zL3JvbGUiOiJBZG1pbmlzdHJhdG9yIiwiZXhwIjoxNzIwNDc4MTAwLCJpc3MiOiJodHRwOi8vb
        G9jYWxob3N0OjUyODMiLCJhdWQiOiJodHRwczovL2xvY2FsaG9zdDo1MjgzIn0.
        iNaBcTdDaHWpb4FV1yydKd4C4Uf1XDFxP_wJDuY6GNY",
    "expiration": "2024-07-08T22:35:00Z"
}
```

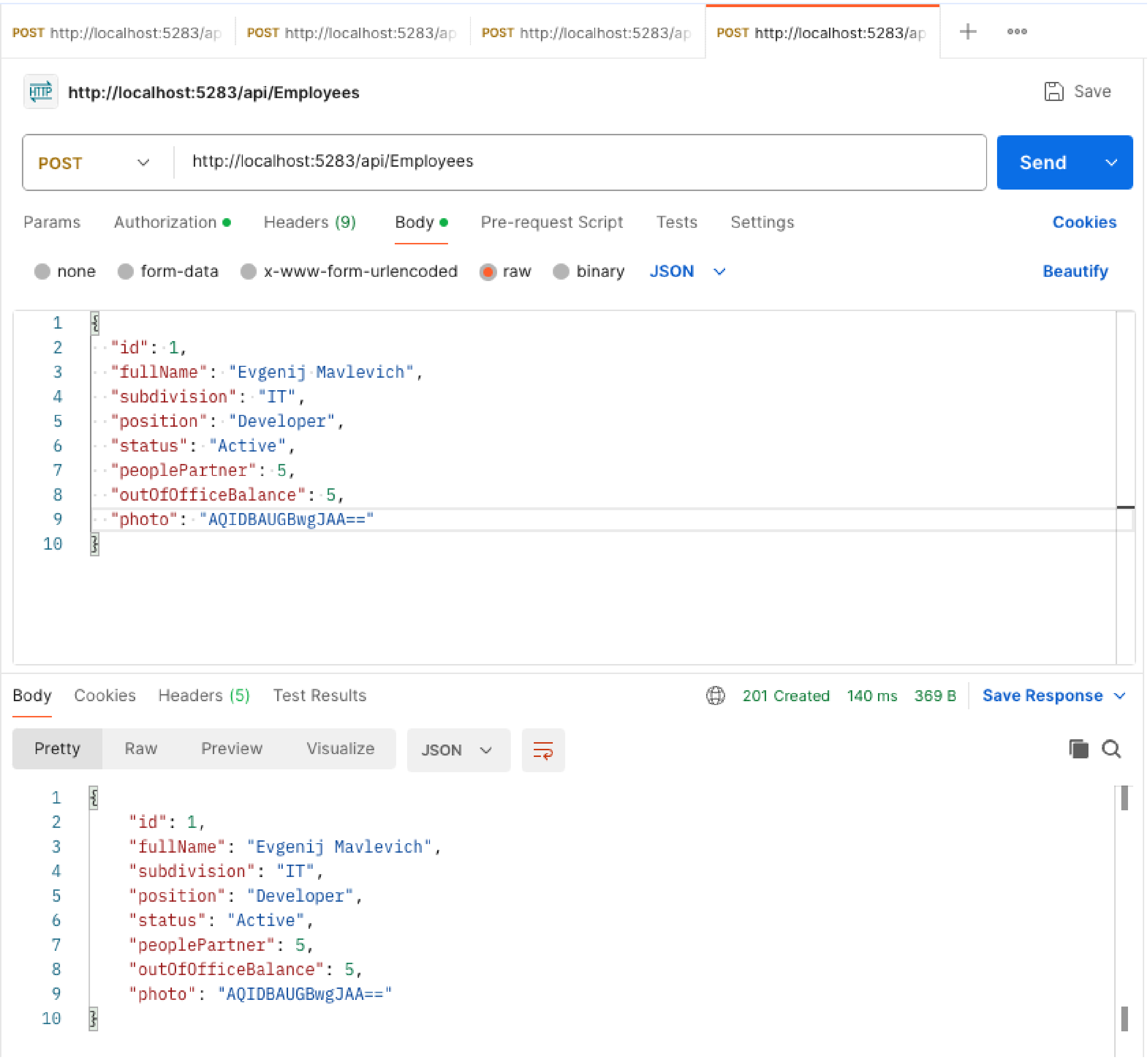
Endpoints for employee management:



Employees

GET	/api/Employees
POST	/api/Employees
GET	/api/Employees/{id}
PUT	/api/Employees/{id}
DELETE	/api/Employees/{id}

Creating an employee:



POST http://localhost:5283/api/Employees

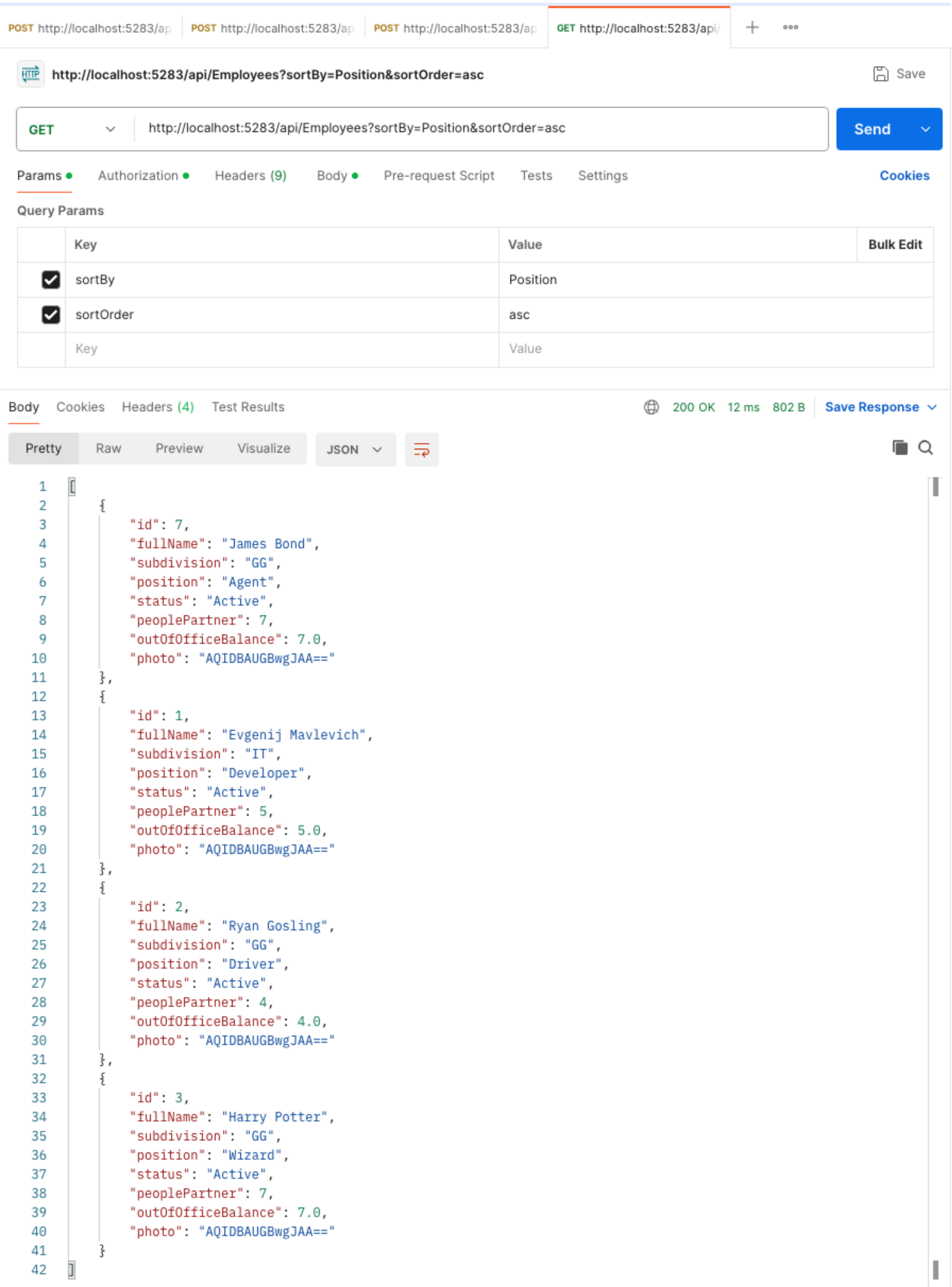
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON

```
{
  "id": 1,
  "fullName": "Evgenij Mavlevich",
  "subdivision": "IT",
  "position": "Developer",
  "status": "Active",
  "peoplePartner": 5,
  "outOfOfficeBalance": 5,
  "photo": "AQIDBAUGBwgJAA=="
}
```

201 Created 140 ms 369 B

```
{
    "id": 1,
    "fullName": "Evgenij Mavlevich",
    "subdivision": "IT",
    "position": "Developer",
    "status": "Active",
    "peoplePartner": 5,
    "outOfOfficeBalance": 5,
    "photo": "AQIDBAUGBwgJAA=="
}
```

Receiving employees sorted by position:



http://localhost:5283/api/Employees?sortBy=Position&sortOrder=asc

GET http://localhost:5283/api/Employees?sortBy=Position&sortOrder=asc

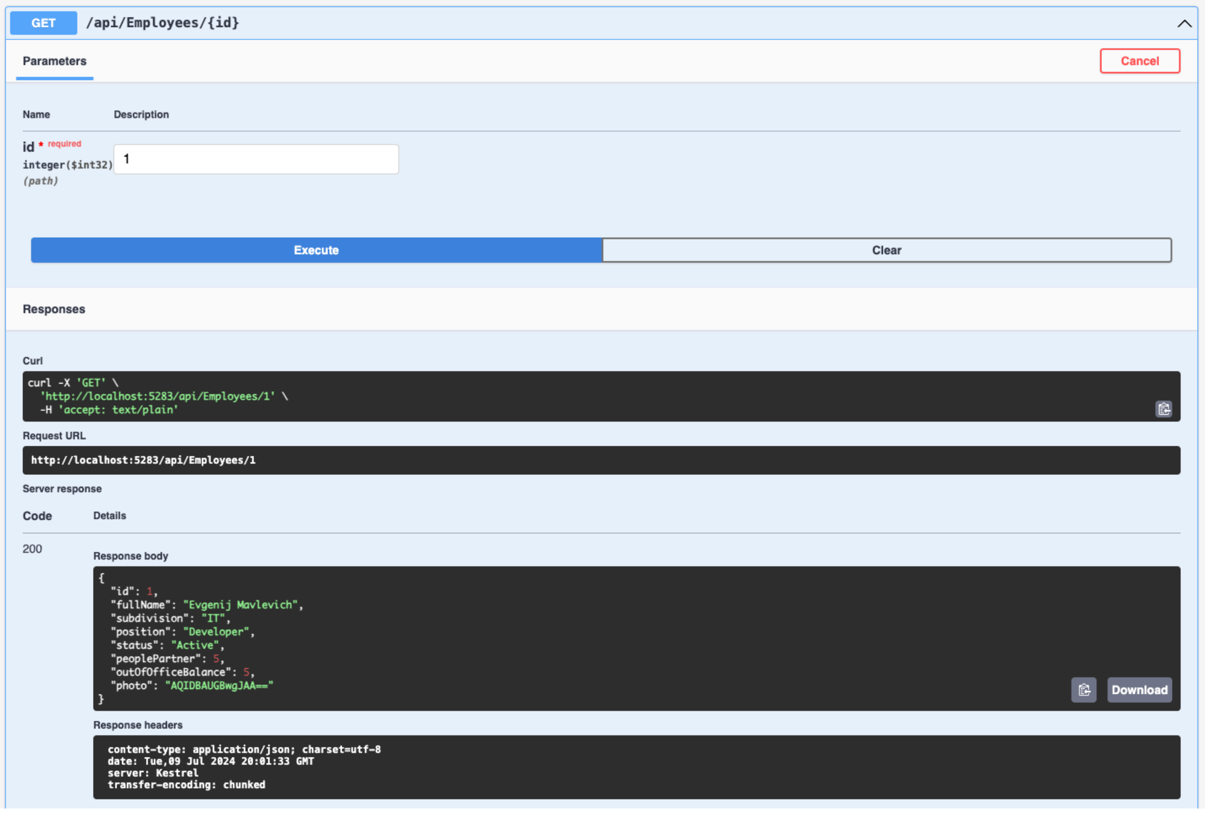
Query Params

	Key	Value	Bulk Edit
☑	sortBy	Position	
☑	sortOrder	asc	
	Key	Value	

200 OK 12 ms 802 B

```
[
    {
        "id": 7,
        "fullName": "James Bond",
        "subdivision": "GG",
        "position": "Agent",
        "status": "Active",
        "peoplePartner": 7,
        "outOfOfficeBalance": 7.0,
        "photo": "AQIDBAUGBwgJAA=="
    },
    {
        "id": 1,
        "fullName": "Evgenij Mavlevich",
        "subdivision": "IT",
        "position": "Developer",
        "status": "Active",
        "peoplePartner": 5,
        "outOfOfficeBalance": 5.0,
        "photo": "AQIDBAUGBwgJAA=="
    },
    {
        "id": 2,
        "fullName": "Ryan Gosling",
        "subdivision": "GG",
        "position": "Driver",
        "status": "Active",
        "peoplePartner": 4,
        "outOfOfficeBalance": 4.0,
        "photo": "AQIDBAUGBwgJAA=="
    },
    {
        "id": 3,
        "fullName": "Harry Potter",
        "subdivision": "GG",
        "position": "Wizard",
        "status": "Active",
        "peoplePartner": 7,
        "outOfOfficeBalance": 7.0,
        "photo": "AQIDBAUGBwgJAA=="
    }
]
```

Getting an employee by ID :



/api/Employees/{id}

Name | Description

id * required
integer($int32)
(path)

1

Curl

```
curl -X 'GET' \
  'http://localhost:5283/api/Employees/1' \
  -H 'accept: text/plain'
```

Request URL

```
http://localhost:5283/api/Employees/1
```

Server response

Code | Details

200

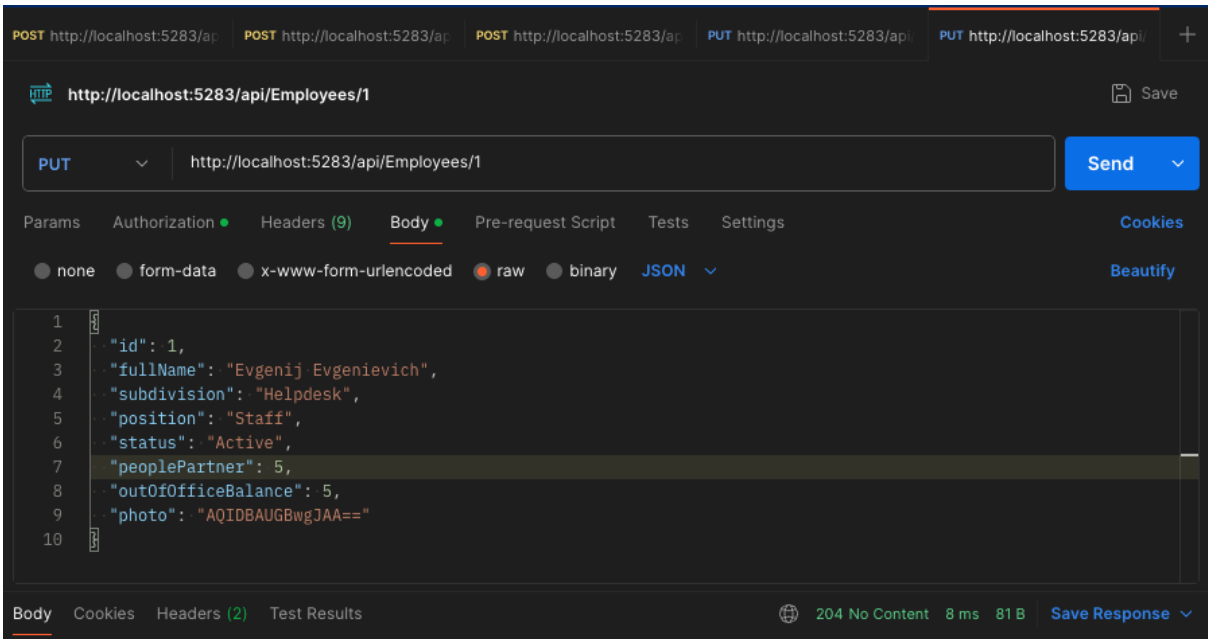
Response body

```
{
  "id": 1,
  "fullName": "Evgenij Mavlevich",
  "subdivision": "IT",
  "position": "Developer",
  "status": "Active",
  "peoplePartner": 5,
  "outOfOfficeBalance": 5,
  "photo": "AQIDBAUGBwgJAA=="
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Tue,09 Jul 2024 20:01:33 GMT
server: Kestrel
transfer-encoding: chunked
```

Changes to employee data:



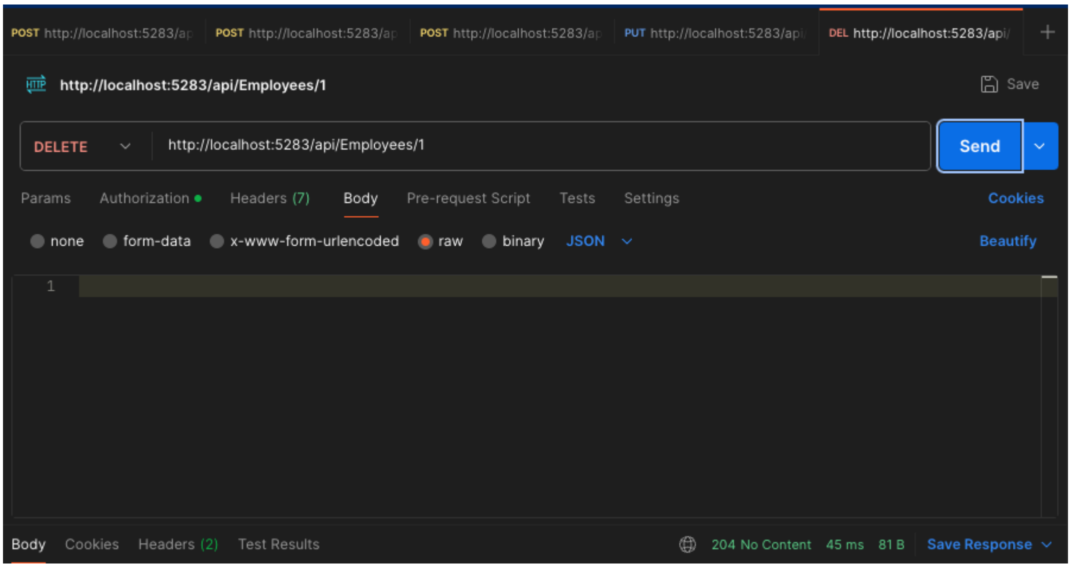
http://localhost:5283/api/Employees/1

PUT http://localhost:5283/api/Employees/1

```
{
  "id": 1,
  "fullName": "Evgenij Evgenievich",
  "subdivision": "Helpdesk",
  "position": "Staff",
  "status": "Active",
  "peoplePartner": 5,
  "outOfOfficeBalance": 5,
  "photo": "AQIDBAUGBwgJAA=="
}
```

204 No Content 8 ms 81 B

Delete employee:



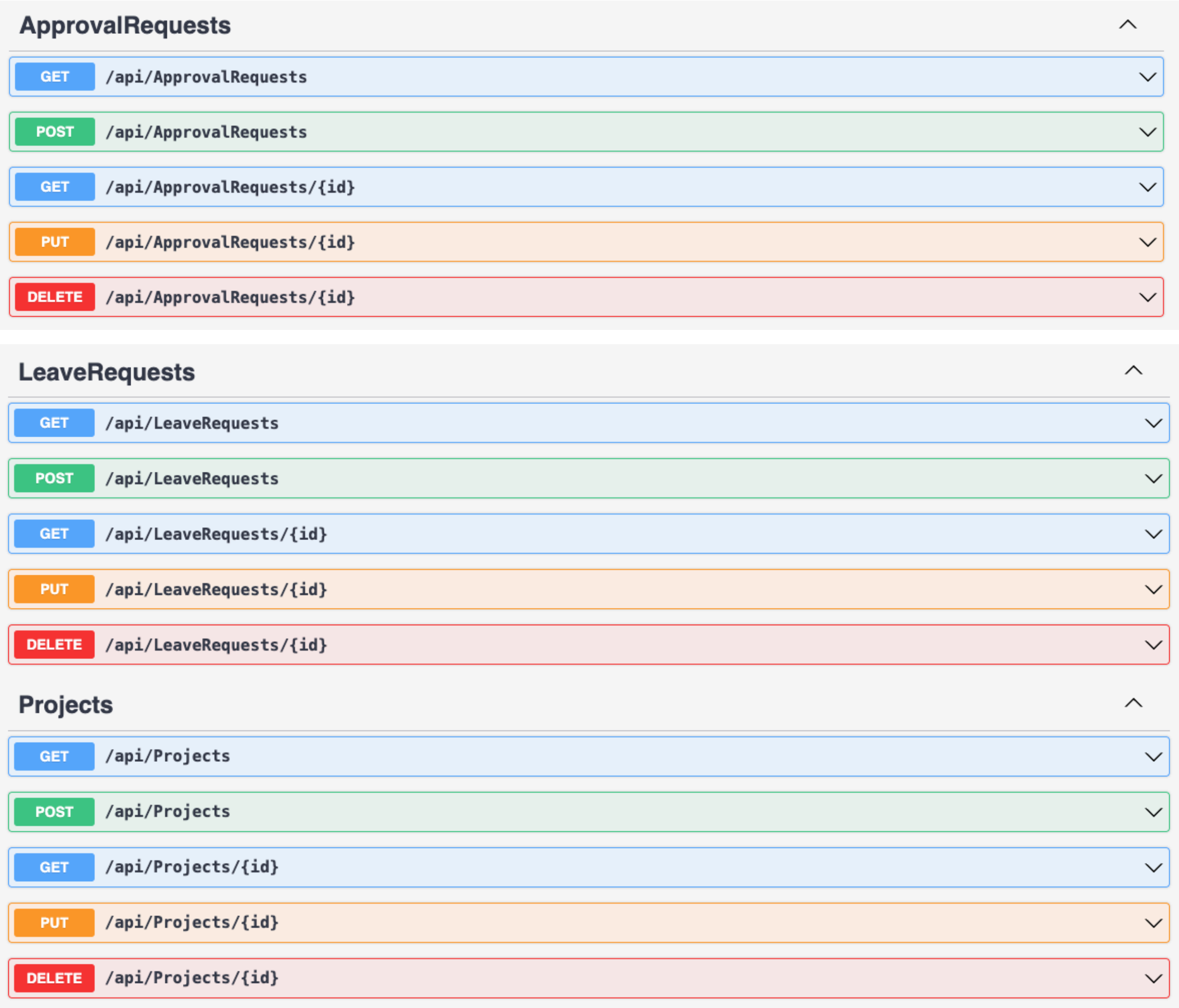
http://localhost:5283/api/Employees/1

DELETE http://localhost:5283/api/Employees/1

● raw JSON

204 No Content 45 ms 81 B

Endpoints with the same functionality have also been implemented for : Approval Request, Project, Leave Request



ApprovalRequests

GET /api/ApprovalRequests

POST /api/ApprovalRequests

GET /api/ApprovalRequests/{id}

PUT /api/ApprovalRequests/{id}

DELETE /api/ApprovalRequests/{id}

LeaveRequests

GET /api/LeaveRequests

POST /api/LeaveRequests

GET /api/LeaveRequests/{id}

PUT /api/LeaveRequests/{id}

DELETE /api/LeaveRequests/{id}

Projects

GET /api/Projects

POST /api/Projects

GET /api/Projects/{id}

PUT /api/Projects/{id}

DELETE /api/Projects/{id}