

Task:

1) Create and design DB, using SQL language that contains:

- 1) users table -> username, id, second name, email, phone number, address;
- 2) position of user table -> user can have a high or low position (for example manager or subordinate)
- Tables should be connected

2) Create API using C# language, you can install Microsoft Visual Studio Community. It is free

- Get the list of users
- Post new User; - add new user;
- Get the position of the user by userId;

3) Optional Add Swagger

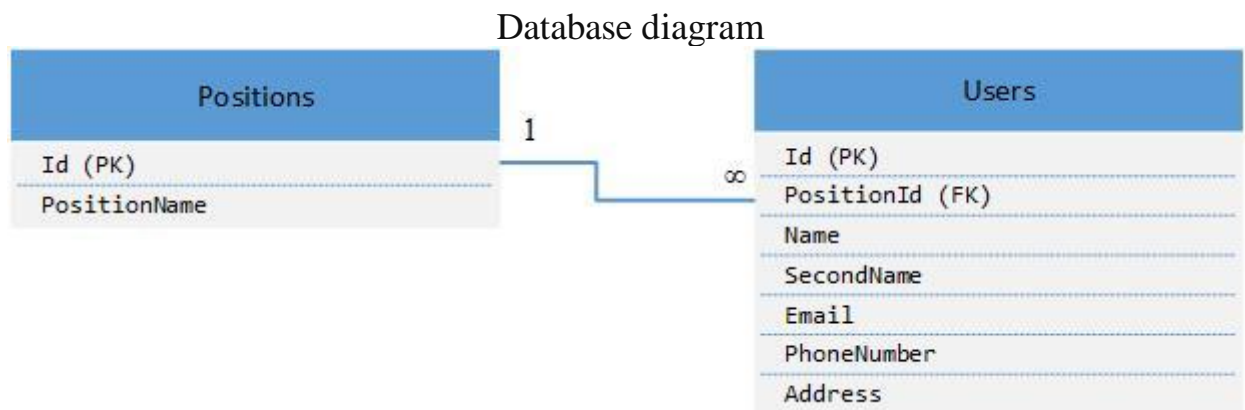
<https://code.msdn.microsoft.com/Swagger-integration-in-da408b29>

4) add 3 unit tests, basic;

Project was created in monolithic architecture.

1. Create and design DB, using SQL language.

I used Code First to existing database (I didn't use automatically creating DB).



```
CREATE TABLE Positions
(
    Id INT IDENTITY PRIMARY KEY,
    PositionName NVARCHAR(50) NOT NULL,
)
```

GO

```
CREATE TABLE Users
(
    Id INT IDENTITY PRIMARY KEY,
    PositionId INT NOT NULL FOREIGN KEY REFERENCES Positions(Id) ON DELETE CASCADE,
    Name NVARCHAR(20) NOT NULL,
    SecondName NVARCHAR(20) NOT NULL,
    Email VARCHAR(50) NOT NULL,
    PhoneNumber VARCHAR(20) NOT NULL,
    Address NVARCHAR(100) NOT NULL
)
```

GO

```
INSERT INTO Positions(PositionName)
VALUES
('manager'),
('subordinate')
```

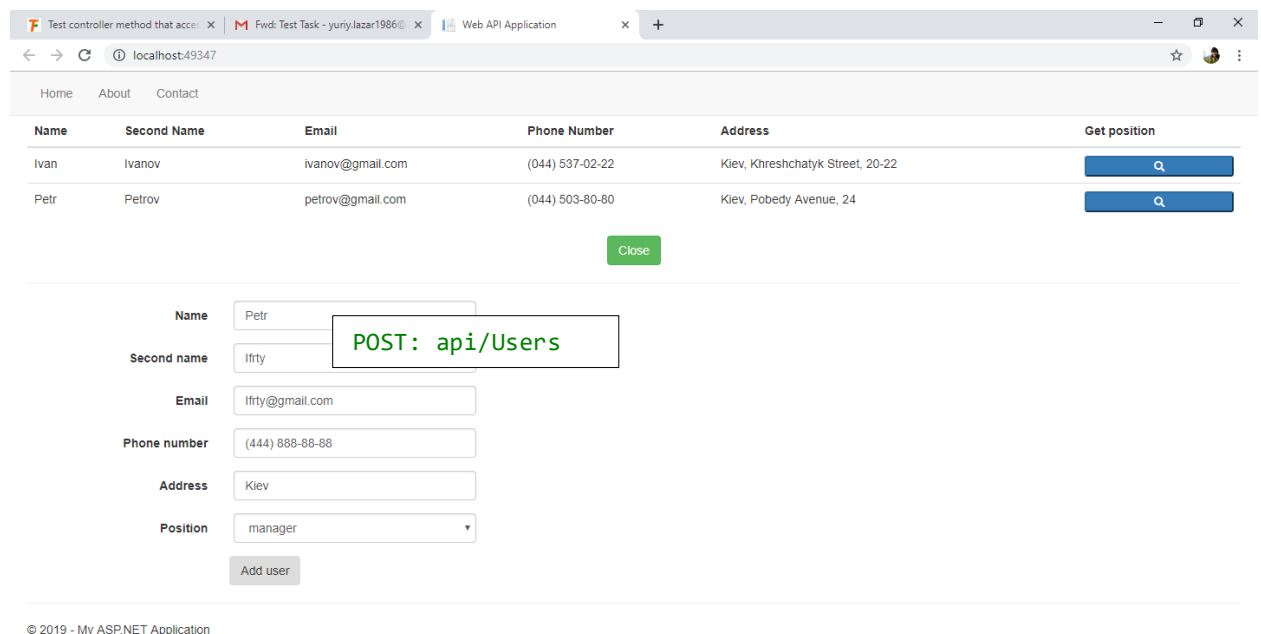
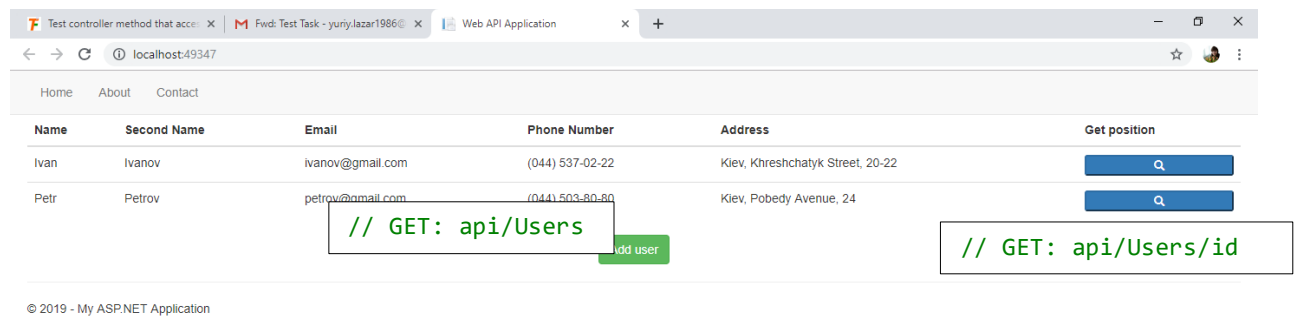
GO

```
INSERT INTO Users(PositionId,Name,SecondName,Email,PhoneNumber,Address)
VALUES
(1,'Ivan','Ivanov','ivanov@gmail.com','(044) 537-02-22', 'Kiev, Khreshchatyk Street, 20-22'),
(2,'Petr','Petrov','petrov@gmail.com','(044) 503-80-80', 'Kiev, Pobedy Avenue, 24')
```

2. Create API using C# language, you can install Microsoft Visual Studio Community.

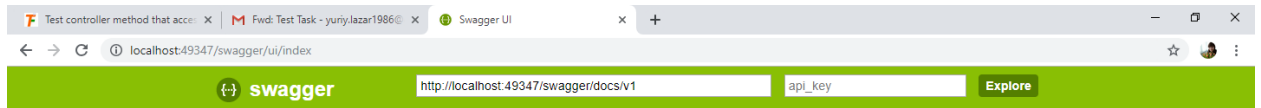
- Get the list of users
- Post new User; - add new user;
- Get the position of the user by userId;

I add UsersController and add this methods. Data validation is performed on client and server side.



3) Optional Add Swagger

<https://code.msdn.microsoft.com/Swagger-integration-in-da408b29>



WebApiAppl

Users

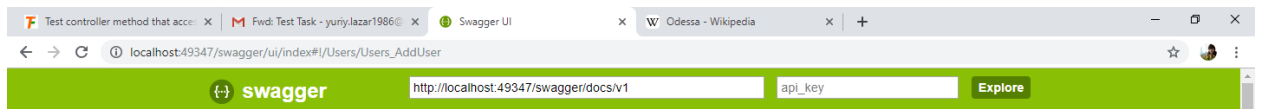
Show/Hide | List Operations | Expand Operations

GET /api/Users

POST /api/Users

GET /api/Users/{id}

[BASE URL: , API VERSION: V1]



WebApiAppl

Users

Show/Hide | List Operations | Expand Operations

GET /api/Users

POST /api/Users

Response Class (Status 200)

OK

Model Example Value

{}

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type				
user	<div><pre>{ "PositionId": 2, "Name": "querty", "SecondName": "querty", "Email": "qvfhhk@gmail.com", "PhoneNumber": "string", "Address": "Odesa"}</pre></div> <div>Parameter content type: <div>application/json</div></div>		body	<table><tr><th>Model</th><th>Example Value</th></tr><tr><td></td><td><pre>{ "Id": 0, "PositionId": 0, "Name": "string", "SecondName": "string", "Email": "string", "PhoneNumber": "string", "Address": "string"}</pre></td></tr></table>	Model	Example Value		<pre>{ "Id": 0, "PositionId": 0, "Name": "string", "SecondName": "string", "Email": "string", "PhoneNumber": "string", "Address": "string"}</pre>
Model	Example Value							
	<pre>{ "Id": 0, "PositionId": 0, "Name": "string", "SecondName": "string", "Email": "string", "PhoneNumber": "string", "Address": "string"}</pre>							

Parameter content type:

application/json

Try it out!

4) add 3 unit tests, basic;

