

December Labs

Banking System

Para instalar la webApi precisaremos tener una Base de Datos MySql local y una instancia de IIS con el .Net Core Hosting Bundle instalado.

Para ello podemos ver la documentación de Microsoft

<https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/iis/index?view=aspnetcore-3.0#direct-download-current-version>

Crearemos una instancia de Base de Datos en la que ejecutaremos los dos scripts adjuntados:

CreateTables.sql

Seeder.sql

Hay un script más que se llama TransactionSeeders.sql que es para crear 10000 transacciones, no lo ejecutaremos de momento.

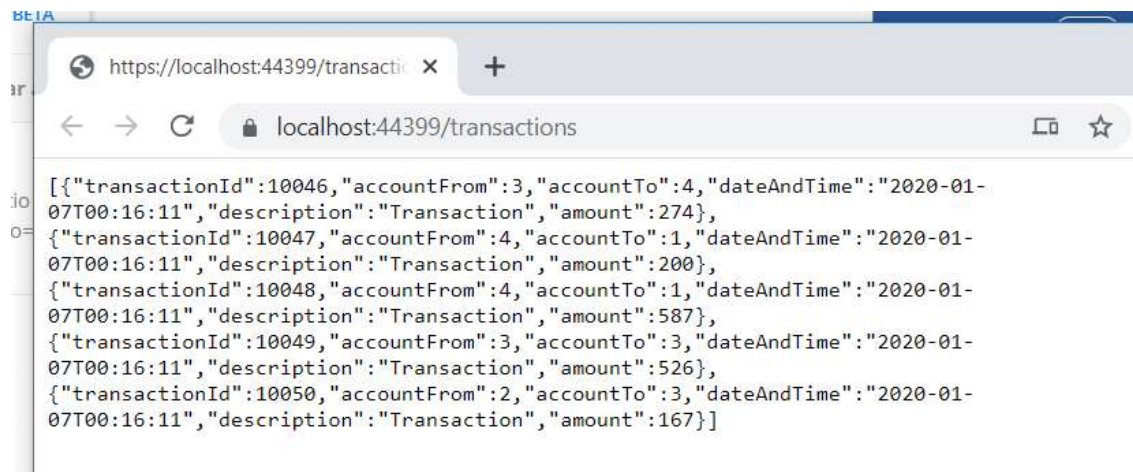
Con esto ejecutado nuestra Base de datos debería verse algo así:



Para comprobar rápidamente si está funcionando bien podemos ejecutar

<https://localhost:44399/transactions>

Y ver que nos devuelva 5 transacciones



Además de adjuntar una colección de postman con unos ejemplos aquí podemos ver unas imágenes:

GET

GET <https://localhost:44399/transactions?dateFrom=2012-12-31&dateTo=2020-12-31&accountFrom=2>

Params **Authorization** Headers (7) Body Pre-request Script Tests Settings

Query Params

| KEY | VALUE |
|---|------------|
| <input checked="" type="checkbox"/> dateFrom | 2012-12-31 |
| <input checked="" type="checkbox"/> dateTo | 2020-12-31 |
| <input checked="" type="checkbox"/> accountFrom | 2 |
| Key | Value |

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize **BETA** JSON

```
1 [
2   {
3     "transactionId": 10050,
4     "accountFrom": 2,
5     "accountTo": 3,
6     "dateAndTime": "2020-01-07T00:16:11",
7     "description": "Transaction",
8     "amount": 167
9   }
10 ]
```

POST

POST <https://localhost:44399/transfer>

Params **Authorization** Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **BETA** JSON

```
1 {
2   "AccountFromId":1,
3   "AccountToId":4,
4   "Amount":100,
5   "DateAndTime":"2020-02-06T00:50:15",
6   "Description":"Cambio a dólares"
7 }
```

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize **BETA** JSON

```
1 {
2   "transactionId": 10051,
3   "accountFromId": 1,
4   "accountFrom": null,
5   "accountToId": 4,
6   "accountTo": null,
7   "dateAndTime": "2020-02-06T00:50:15",
8   "description": "Cambio a dólares",
9   "amount": 100
10 }
```

Limitaciones

El producto por ahora no está haciendo uso de Usuario para la autenticación y el filtro al consultar. Esto pensaba implementarlo.

Prioricé el poder realizar transferencias y que estas hagan el cambio de divisa.

No se ha probado levantar en dos nodos. No tuve tiempo de poder probar esto a pesar de que todo indicaría que se puede realizar.

Mejoras

Las dos limitaciones mencionadas anteriormente creo que se podrían realizar sin tantos cambios.

Me gustaría poder realizar un mucho mejor manejo de errores ya que pensaba dedicar una feature completa a hacerlo.

También me gustaría poder hacer un refactor y revisar bien que el código sea legible.

En materia de diseño creo que hay varias cosas para pulir también.

Tiempo

Teniendo en cuenta la limitante del tiempo creo que se utilizó demasiado en investigación sobre tecnologías en las que no estoy tan familiarizado. Tardé un poco más de lo esperado configurando bien la conexión entre EF y MySQL. En total fueron unas 10 horas entre investigación y desarrollo.