



MusicStoreApi backend 202401

Levantar ambiente

1. Clonar repo → <https://github.com/mitocode21/musicstore-backend202401>
2. En la carpeta raiz del proyecto ejecutar:

```
docker compose up -d
```

```
PS C:\Users\gerso\source\repos\musicstoreapi-initial> docker compose up -d
[+] Running 2/0
 ✓ Container mydb          Running      0.0s
 ✓ Container musicstoreapi Running      0.0s
PS C:\Users\gerso\source\repos\musicstoreapi-initial> 
```

```
MINGW64/c/Users/gerso/source/mitocode-repos/MusicStore

gerso@GThinkpad MINGW64 ~/source/mitocode-repos/MusicStore (master)
$ docker compose up -d
#0 building with "default" instance using docker driver

#1 [app internal] load .dockerignore
#1 transferring context: 382B 0.0s done
#1 DONE 0.1s

#2 [app internal] load build definition from Dockerfile
#2 transferring dockerfile: 1.24kB 0.0s done
#2 DONE 0.1s

#3 [app internal] load metadata for mcr.microsoft.com/dotnet/sdk:8.0
#3 DONE 1.0s





















#4 [app internal] load metadata for mcr.microsoft.com/dotnet/aspnet:8.0
#4 DONE 0.9s
```

Verificar en docker que se haya levantado el contenedor:

Containers [Give feedback](#)

Container CPU usage ⓘ 88.00% / 800% (8 cores allocated) Container memory usage ⓘ 814.76MB / 3.61GB [Show charts](#) ▾

🔍 Search ☰ ⏻ Only show running containers

<input type="checkbox"/>	Name	Image	Status	Port(s)	Last started	CPU (%)	Actions
<input type="checkbox"/>	>  playinfra		Exited		2 months ago	0	  
<input type="checkbox"/>	▾  musicstore		Running (2/2)		2 seconds ago	88	  
<input type="checkbox"/>	 mydb 5383f7558958 	mcr.micros	Running	2590:1433 	3 minutes ago	1.88	  
<input type="checkbox"/>	 musicstoreapi 89dd2681abc9 	musicstore	Running	8080:80 	2 seconds ago	86.12	  

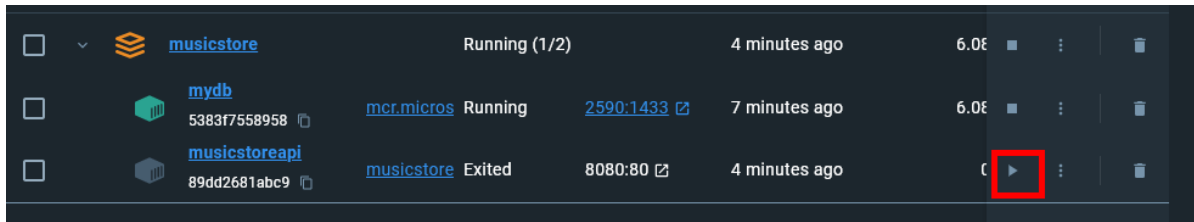
Tras unos segundos, es posible que en la primera ejecución el container `musicstoreapi` se detenga porque aun no estuvo creado el container `mydb` y haya tenido error al ejecutar cadena de conexión.

Para solucionar eso, ejecutar:

```
docker start musicstoreapi
```

```
gerso@GThinkpad MINGW64 ~/source/mitocode-repos/MusicStore (master)
$ docker start musicstoreapi
musicstoreapi
```

o alternatively se puede iniciar manualmente en docker.

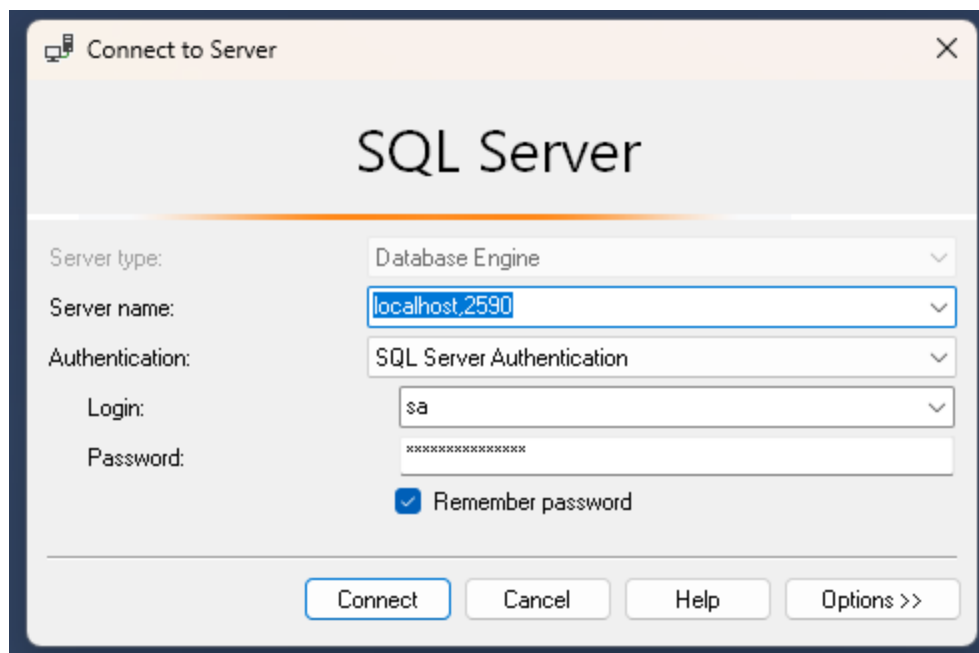


<input type="checkbox"/>		musicstore	Running (1/2)	4 minutes ago	6.08			
<input type="checkbox"/>		mydb	Running	7 minutes ago	6.08			
<input type="checkbox"/>		musicstoreapi	Exited	4 minutes ago	0			

3. Verificar en SSMS o algún gestor de BD

Utilizar las credenciales:

SERVER: localhost,2590
USER: sa
PW: MC1admin



Connect to Server

SQL Server

Server type: Database Engine

Server name: localhost,2590

Authentication: SQL Server Authentication

Login: sa

Password:

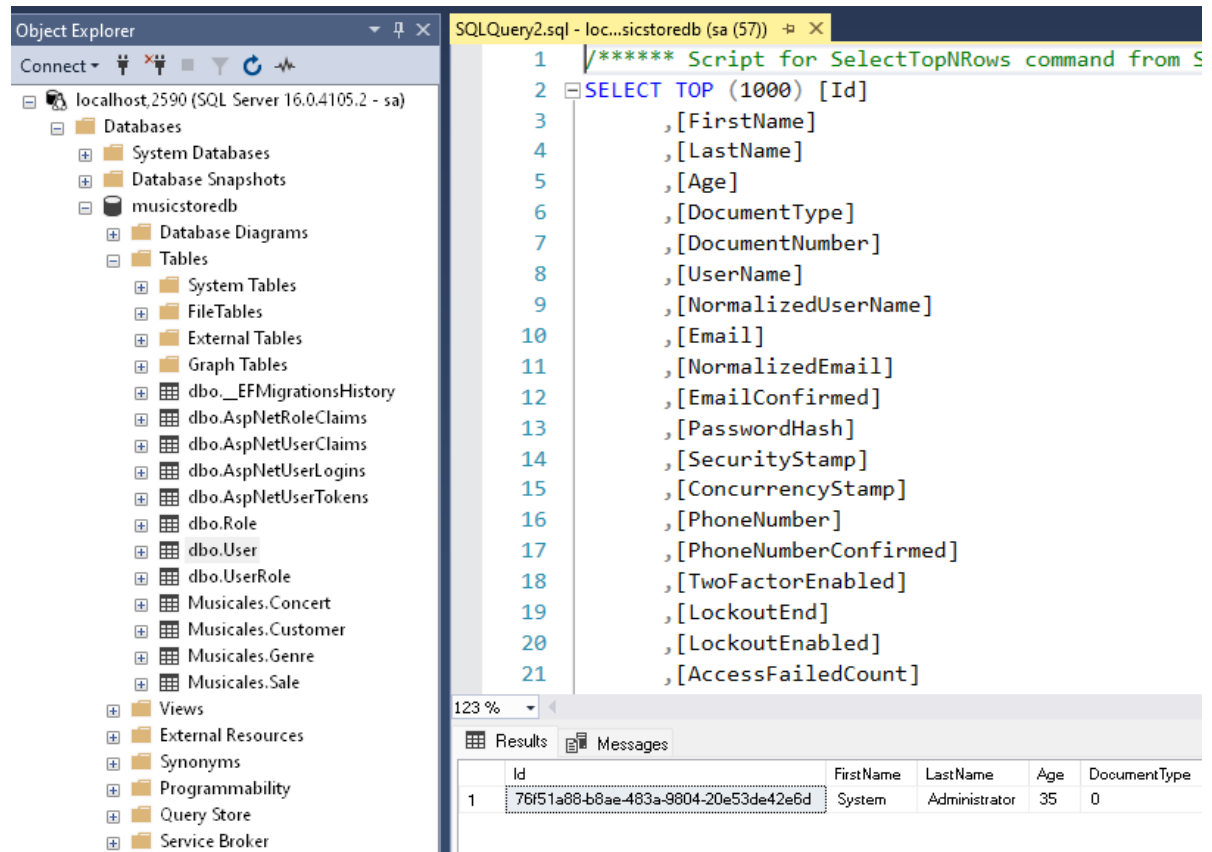
☒ Remember password

Connect Cancel Help Options >>

Se ha creado la base de datos completa, sin datos.

El único dato que se insertó como seed data es un usuario admin.

Verificando el usuario admin creado por defecto:



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the 'musicstoredb' database with various tables. The main window shows a SQL query window with the following query:

```
1 /***** Script for SelectTopNRows command from S
```

```
2 SELECT TOP (1000) [Id]
```

```
3     , [FirstName]
```

```
4     , [LastName]
```

```
5     , [Age]
```

```
6     , [DocumentType]
```

```
7     , [DocumentNumber]
```

```
8     , [UserName]
```

```
9     , [NormalizedUserName]
```

```
10    , [Email]
```

```
11    , [NormalizedEmail]
```

```
12    , [EmailConfirmed]
```

```
13    , [PasswordHash]
```

```
14    , [SecurityStamp]
```

```
15    , [ConcurrencyStamp]
```

```
16    , [PhoneNumber]
```

```
17    , [PhoneNumberConfirmed]
```

```
18    , [TwoFactorEnabled]
```

```
19    , [LockoutEnd]
```

```
20    , [LockoutEnabled]
```

```
21    , [AccessFailedCount]
```

The Results pane shows the first row of data:

	Id	FirstName	LastName	Age	DocumentType
1	76f51a88-b8ae-483a-9804-20e53de42e6d	System	Administrator	35	0

4. Validar endpoints

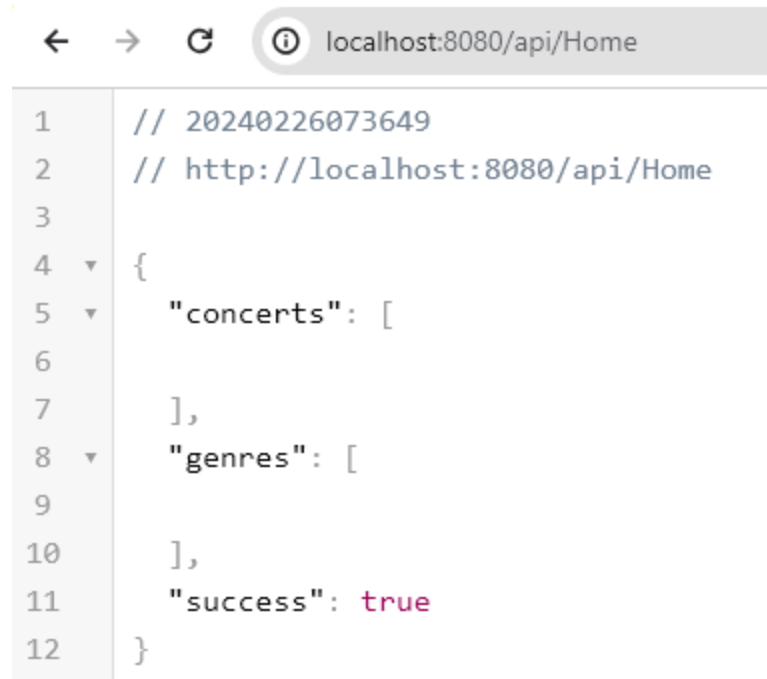
Browser

Swagger → <http://localhost:8080/swagger/index.html>

Browser

Verificar las URI home en navegador → <http://localhost:8080/api/home>

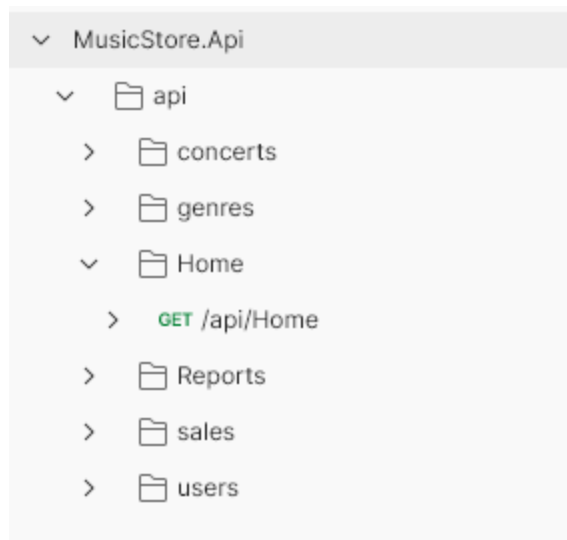
De momento no habrá data pero el api está disponible para su uso



```
1 // 20240226073649
2 // http://localhost:8080/api/Home
3
4 {
5   "concerts": [
6
7   ],
8   "genres": [
9
10  ],
11   "success": true
12 }
```

Postman

Importar en postman la colección que está en la raíz del repo



Para el login como usuario admin en el API utilizar las credenciales:

Users/Login

```
"userName": "admin@gmail.com",  
"password": "Admin1234*"
```

Eso te proporciona la JWT Bearer para que la añadas como header `authorization` en los endpoints que lo requieran

