**Documentació UsersManagement**

Autor del crim: Eduard Moret Ruz

**Una breu descripció**  
  
En aquest exercici poso en pràctica les meves habilitats com a programador. En el context de l’exercici he optat per crear un projecte amb el frontend i el backend totalment desacoplat per a un major manteniment i independència en el desenvolupament entre les parts del projecte.

Les especificacions tècniques són les següents:

Backend:

* Projecte MVC WepAPI2
* Entity Framework 4.6.1
* Linq
* Arquitectura N-Capas (Presentació, Negoci, Accés a dades)
* Desacoblament de dependències amb Ninject
* Securitzaió de la API mitjançant OAuth.
* Projecte SQL per desplegar en una LocalDB
* Capa de Tests unitaris
* Patró repositori

Front-End

* Framework SPA: Angular 7
* Html5
* Css3
* TypeScript

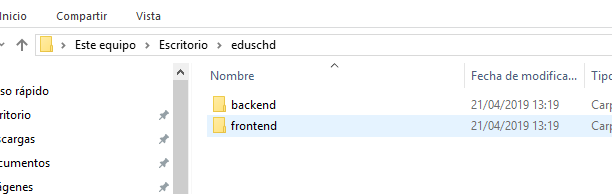
IDES utilitzats:

* Visual Studio 2017
* Visual Studio Code

**Accés al codi font**

Per obtenir el codi Font ho farem en dues parts:

Jo personalment optaria per crear un directori amb el nom del projecte, i dins d’aquest dues carpetes. Una d’elles li direm Backend i l’altre Frontend com mostro a la següent imatge:

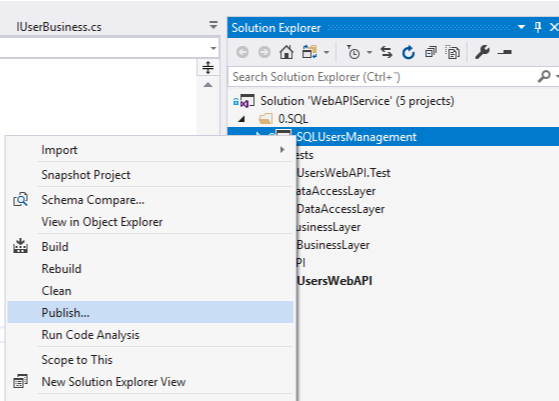


Entrarem dintre de la carpeta Backend i mitjançant un client de Git ens descarregarem el codi del repositori GitLab. Jo com a costum utilitzo tortoise git però això va a gust del consumidor…  
  
Fem clone del següent repositori:

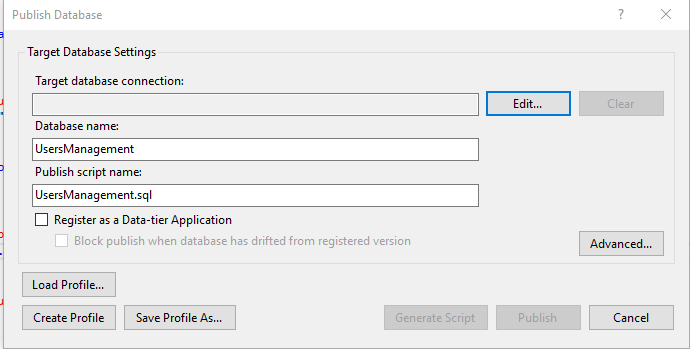
<https://gitlab.com/edward_catacrocker/webapi2-test.git>

**Desplegament del backend**

Un cop tinguem el codi font al nostre abast clicarem en el fitxer **WebApiService.sln** i se’ns obrirà el Visual Studio . Veurem aquesta estructura i fem clic al botó dret sobre el projecte **SQLUsersManagement** tal i com mostro a la imatge i seleccionem l’opció de **publish**:

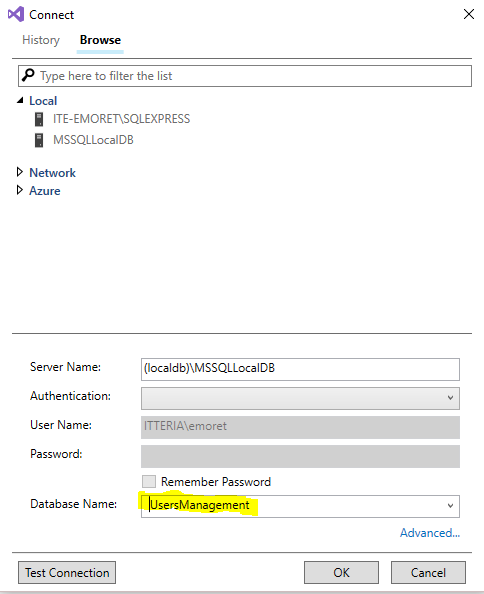


Apareixerà el següent assistent per desplegar la base de dades



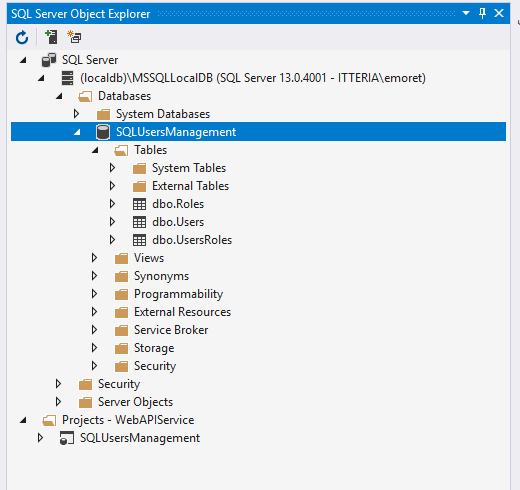
Posem *UsersManagement* al camp *Database name* (important) i premem el botó *Edit*…

Seleccionem *browse local* i la opció de *LocalDB*. Això ens desplegarà una base de dades local sense haver de necessitar gestionar-ho amb un SQL Server Management Studio.



A la nostra base de dades li hem de dir **UsersManagement (és important que es digui igual)** provarem la connexió si s’escau amb “*Test Connection*” i si ha anat bé clicarem *OK*.

Ara s’haurà desplegat una base de dades amb les dades dels usuaris incloses en local.  
  
Com veieu tenim la base de dades desplegada i la podem gestionar des del propi Visual Studio a través del **SQL Server Object Explorer.**



**Nota important**

**Si al compilar el nostre codi backend tenim algun problema amb les .dll, podem fer un *clean* i un *rebuild* de toda la solució. A vegades cal esborrar les carpetes bin i obj dels projectes de la solució i fer rebuild de tot el projecte.**

Desplegament de la SPA

Angular és un framework que s’aixeca sobre un entorn Node.js per tant és important que abans de seguir amb els següents passos ens instal·lem el Node a la nostra màquina. El podrem descarregar aquí:

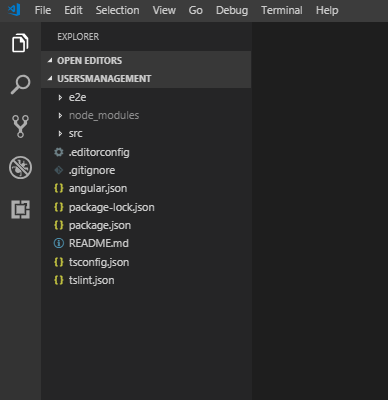
<https://nodejs.org/en/>

Jo recomano instal·lar la versió LTS (long term support) per a un correcte funcionament.

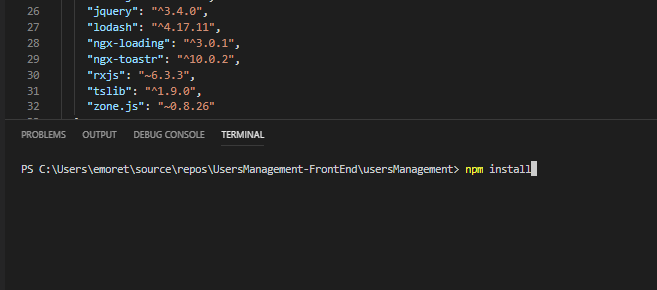
Un cop tenim el backend totalment operatiu haurem d’accedir al codi font del projecte frontend. Tal i com hem seguit amb el pas anterior fem un clone del projecte a la carpeta FrontEnd que hem creat inicialment. La url del gitlab és la següent:

<https://gitlab.com/edward_catacrocker/users-frontend.git>

Un cop tenim el codi descarregat i amb el Visual Studio Code instal·lat obrirem el projecte des de l’arrel de l’aplicació Angular. Ara en el nostre IDE podrem visualitzar el projecte d’aquesta manera:

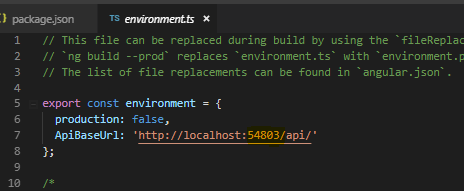


Instal·lant dependències

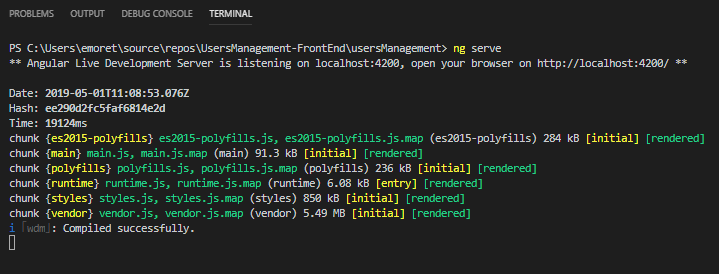
Per a que la nostra aplicació Angular funcioni correctament haurem d’instal·lar les dependències que necessita, les per defecte i les instal·lades per mi. Com ja sabeu aquestes dependències no venen incloses al node\_modules però si que estan declarades al fitxer package.json. Per instal·lar les dependències ho podem fer des de la terminal que ens proveeix el IDE i escrivim *npm install*. Això llegirà el nostre fitxer package.json i les instal·larà.  
  


És possible que aparegui algun *warning* al acabar d’instal·lar-les però no és un problema.

Un cop tinguem les dependències instal·lades navegarem fins el fitxer *environment.ts* del projecte. N’hi ha prou amb escriure Ctrl P i buscar *environment.ts*. Un cop el tenim obert ens assegurem que el paràmetre *ApiBaseUrl* apunta al port on s’ha aixecat el nostre backend. En el meu cas ho tenia d’aquesta manera:



Un cop arribat a aquest punt podem aixecar la nostra aplicació de Angular a través de Node. Per això escrivim des de la mateixa ruta on hem fet el *npm install* un *ng serve* o si no tinguéssim l’intèrpret de comandes d’Angular (Angular CLI) fem un *npm build* i tot seguit un *npm start*però en principi hauríem de tenir Angular CLI instal·lat com a dependència.

Si tot ha anat bé, després de fer el *ng serve* se’ns informarà per pantalla indicant-nos, entre d’altres coses que la compilació ha anat correctament i a quin port s’ha desplegat l’aplicació. En el meu cas sobre *localhost* i al port 4200.  
  


Obrim el navegador que volem (jo preferiblement utilitzo Chrome) i posem la url següent: [http://localhost:4200](http://localhost:4200/)

Ara tenint el backend i el frontend aixecats i accedint a aquesta url se’ns mostrarà la pantalla de login.

Joc d’usuaris:

User Password

|  |  |
| --- | --- |
| edu | siberet |
| nuria | borrissol |
| gabri | farigola |
| manu | machupichu |
| Zigor | huhui |
| Donald Duck | hehei |
| Cheewbaka | hohoi |
|  |  |

|  |  |
| --- | --- |
| **Nota important:** |  |

**Les contrasenyes en aquesta aplicació no estan encriptades per afavorir una bona manipulació de les dades des de la pròpia base de dades si fes falta.**  
Salut!