***Projet Remis Par Patrick Oliveira et Michael Tremblay***

***Grille d’évaluation du projet final***

1. Développement de votre API et utilisation de méthodes de requêtes http (GET, PUT, POST, DELETE).

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api

-Base Donnée SQL Server vers Node.js

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>Data>CartesEvents>index.js

-Node.js vers Express

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>Controler>CartesControler.js

1. Développement d’un client qui consomme votre API créée de type application bureau avec la plateforme .NET (Projet WPF par exemple).

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-app

-Express vers React

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-app>src>Page>PageHearthStone.js.

1. Développement d’un deuxième client qui consomme votre API créée de type application Web (Projet REACT par exemple).

-Retrouver dans le laboratoire finale du Cours Dev. Application Console

1. Utilisation des principes de *« routing »* côté client et serveur.

- Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-app>src>Page>PageHearthStone.js.

1. Test de votre API avec Postman ou REST Client par exemple –écrire au moins 10 tests .

-Installer extension rest

- Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>testRest.rest

-Test Postman

-Retrouver dans : https://documenter.getpostman.com/view/24381246/2s93JnVSdu

6-Organisation de votre application et respect des bonnes pratiques.

-Chaque application est décomposer de façon à être le plus claire et simple possible, pour pouvoir permettre de future collaboration.

-my-heartstone-api est composé des répertoires suivants :

-Controller

-Data

-Routes

-Tous les fichiers de configuration sont retrouvées dans le répertoire

my-heartstone-api

--my-heartstone-app est composé des répertoires suivants :

-src

-Components

-NavigationPrincipale

-Pages

-Ressource

-UI

7-Gestion des erreurs pour les requêtes défectueuses

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>Data>CartesEvents>index.js

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>Controler>CartesControler.js

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-app>src>Page>PageHearthStone.js.

8- Ajout de commentaires à votre code.

- Des commentaires pertinent sont retrouver aux débuts de chaque module

-Expliquant la responsabilité de chacun de c’est module

9-Génération de la documentation avec « Swagger » par exemple.

-Documentation Postman

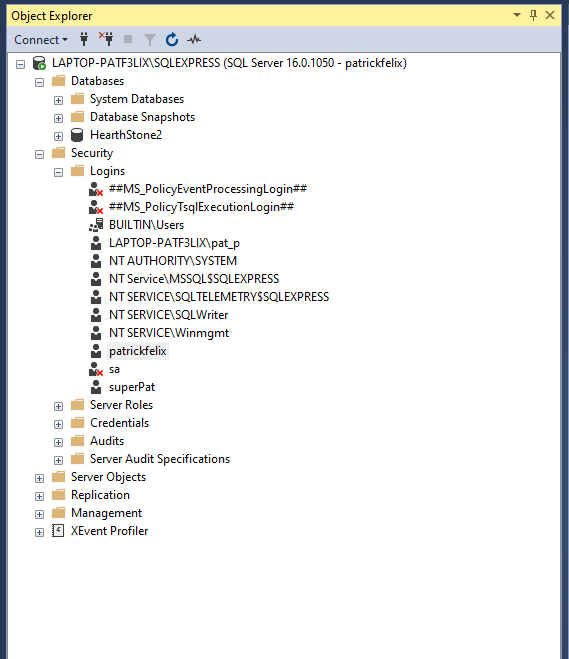
--Retrouver dans : <https://documenter.getpostman.com/view/24381246/2s93JnVSdu>

10- Présentation orale du projet (Présentation de l’ API ) et des perspectives de votre projet. Par exemple, les étudiant.e.s peuvent critiquer leur travail et présenter les parties à améliorer dans le futur.

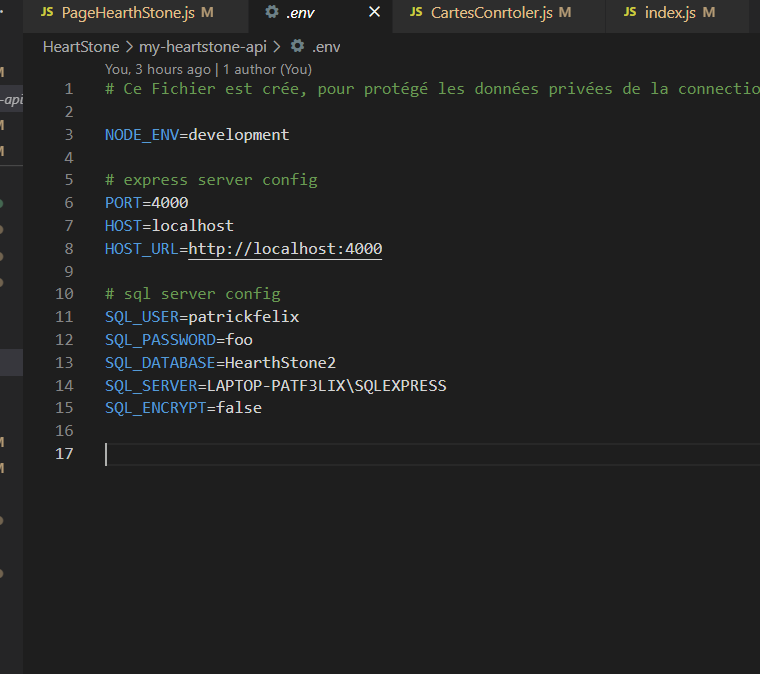
**Guide de l’utilisation**

1-Ouvrir sql serveur et se connect sur son compte utilisateur.

2-Crée un nouveau utilisateur

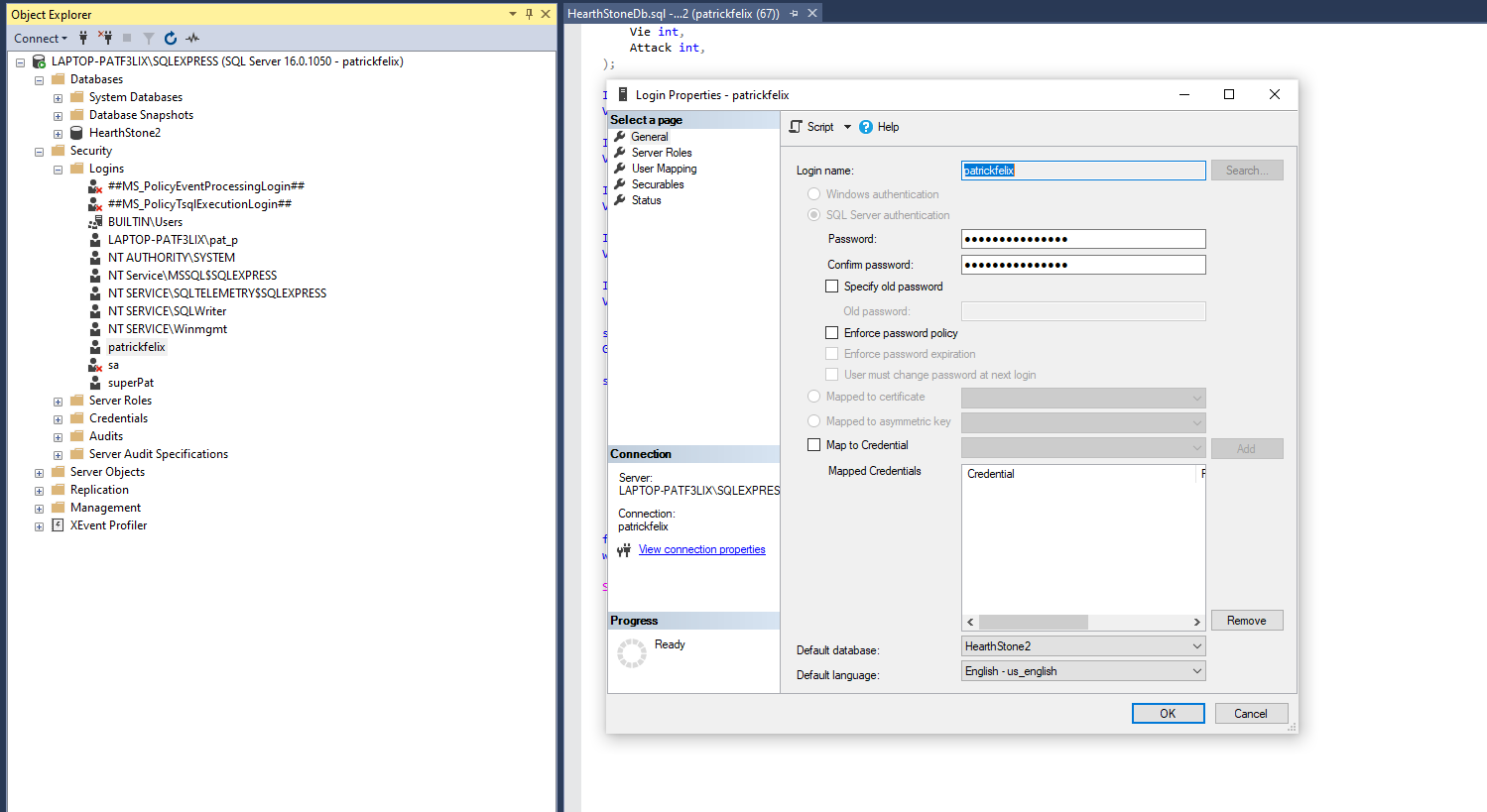


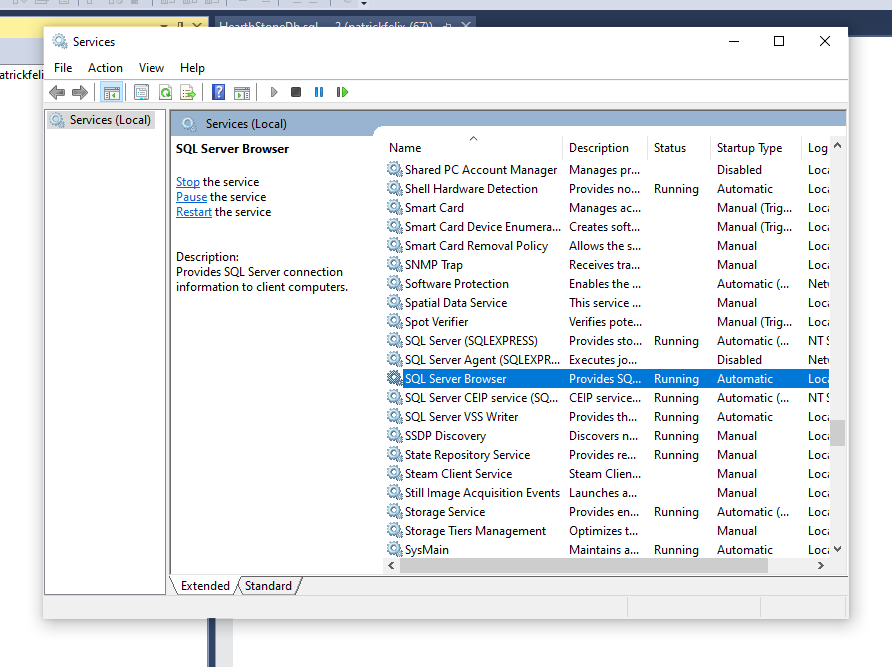
***3-Les information doivent correspondre à celle retrouver dans la photo suivante***

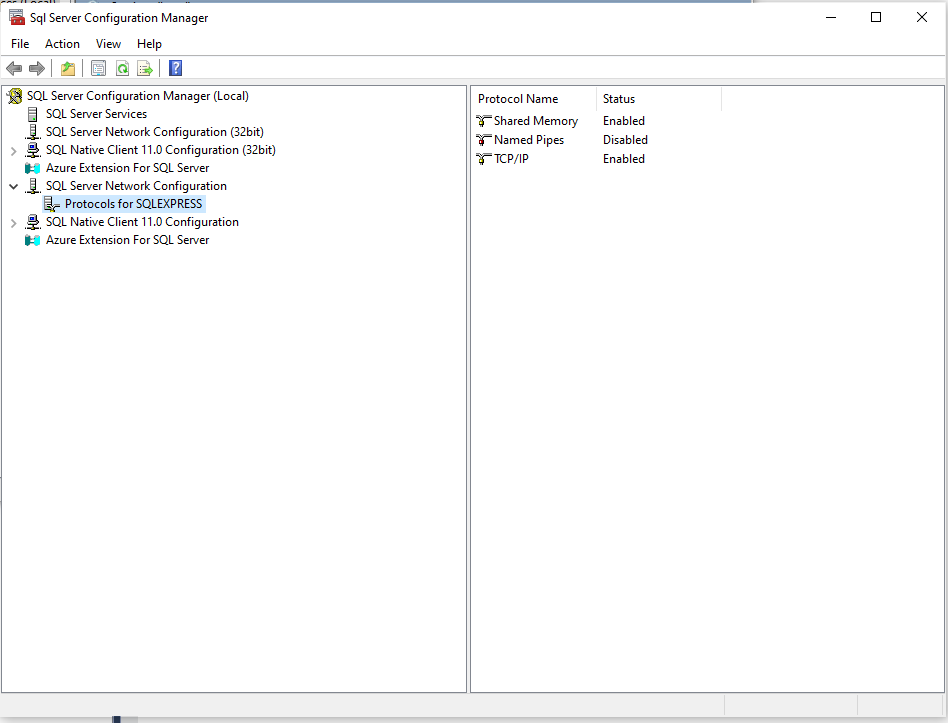
******

***4-assurer vous de choisir identification par server sql***

***Graphical user interface, application

Description automatically generated***

***5-Redémaré dans service sql express ou sql de votre choix et sql browserÊtre sur que sql broswer est enabled***

***6-Dans SqlServerConfiguration Manager être sur que Protocol express ou sql server ou installer et que TCIP est Enabled***

***7-Retourner dans sql server broswer et right-click sur la base de donner et aller dans properties***

***Assurer vous que sql server authentication est cocher***

***Graphical user interface, text, application, email

Description automatically generated***

***Retourner dans server et redémarer sql server( express) et broswer***

***8-Ouvrir une nouvelle Page de query***

***Et Utiliser le Script dans le Projet FullStackHeartStone***

***HearthStoneDb.sql***

create database HearthStone2

Go

use HearthStone2

Go

CREATE TABLE Cartes (

Id int NOT NULL IDENTITY PRIMARY KEY,

Rareter varchar(50),

Nom varchar(100),

Cout int,

Vie int,

Attack int,

);

INSERT INTO Cartes(Rareter, Nom, Cout, Vie, Attack)

VALUES ('LEGENDAIRE', 'Leeroy Jenkins', 5, 2, 6);

INSERT INTO Cartes(Rareter, Nom, Cout, Vie, Attack)

VALUES ('RARE', 'Dragon Azure', 5, 5, 4);

INSERT INTO Cartes(Rareter, Nom, Cout, Vie, Attack)

VALUES ('COMMUN', 'Morticien nécrotique', 2, 3, 2);

INSERT INTO Cartes(Rareter, Nom, Cout, Vie, Attack)

VALUES ('EPIQUE', 'Hachoir à viande', 3, 4, 3);

INSERT INTO Cartes(Rareter, Nom, Cout, Vie, Attack)

VALUES ('LEGENDAIRE', 'Jace Sombretisserand', 8, 5, 7);

select \* from Cartes

Go

select

'data source=' + @@servername +

';initial catalog=' + db\_name() +

case type\_desc

when 'WINDOWS\_LOGIN'

then ';trusted\_connection=true'

else

';user id=' + suser\_name() + ';password=<<YourPassword>>'

end

as ConnectionString

from sys.server\_principals

where name = suser\_name()

SCOPE\_IDENTITY()

Étape setup aplication API

1-Ouvrir dans Vs code le Projet FullStackHeartStone

2-Ouvrir un terminal Ctrl+j

3-ls+enter

4- cd HeartStone

5-ls+enter

6-cd my-heartstone-api

7- Installation nécessaire :

-npm install

-npm install express mssql cors body-parser

-npm install nodemon

-npm install dotenv

-npm install fs-extra

8-npm start

Rooting coter server devrais être maintenant actif

Étape setup aplication coté client

1-Ouvrir dans Vs code le Projet FullStackHeartStone

2-Ouvrir un terminal Ctrl+j

3-ls+enter

4- cd HeartStone

5-ls+enter

6-cd my-heartstone-app

7- Installation nécessaire :

-npm install

-npm install react-router-dom@6

-npm install bootstrap react-bootstrap

-npm start

Rooting coter client devrais être maintenant actif

Et l’Interface graphique devrais vous apparaitre