Dossier P6\_new :

Terminal vsc :

Node.js v14.15.5

Backend : npm init

Dans package.json on a :

{

"name": "backend",

"version": "1.0.0",

"description": "P6\_Openclassrooms",

"main": "server.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

Nodemon : 2.0.7

nodemon server

frontend

npm install

npm start

Angular CLI : 12.2.2 déjà installé (pour la version de Angular CLI taper dans la console ng version)

Dans le dossier frontend : npm install --save node-sass@4.14

node-sass v4.14.1

.gitignore dans backend : node\_modules

\*) Créer server.js et tester le serveur backend :

const http = require('http');

const server = http.createServer((req, res) => {

res.end('Voilà la réponse du serveur !');

});

server.listen(process.env.PORT || 3000);

ok :

\*) installer 2 en 1 : express et dotenv :

Dans backend

npm i express dotenv

+ dotenv@10.0.0

+ express@4.17.1

Installation de dotenv et express : dans backend npm i express dotenv

Remplacez les scripts existants dans package.json par le script suivant afin que nous puissions exécuter notre application :

"start": "node server.js"

{

"main": " server.js",

"scripts": {

"start": "node server.js"

},

\*) créer app.js pour l'application express et modification du fichier server.js pour tenir compte de cette appli Express:

Installation

Here are the dependancies you need to install:

* NodeJS 12.14 or 14.0.
* Angular CLI 7.0.2.
* node-sass : make sure to use the corresponding version to NodeJS. For Nodcd..
* e 14.0 for instance, you need node-sass in version 4.14+.

On Windows, these installations require to use PowerShell in administrator mode.

Then, clone this repo and run npm install.

Usage

Run npm start. This should both run the local server and launch your browser.

If your browser fails to launch, or shows a 404 error, navigate your browser to <http://localhost:8080>.

The app should reload automatically when you make a change to a file.

Use Ctrl+C in the terminal to stop the local server.

Séances mentor :

01/09 :

Q1 : variable d'environnemnt pour les données sensibles bdd etc ..

MySQL est un exemple de base de données relationnelle - vous utiliseriez un ORM pour traduire entre vos objets dans le code et la représentation relationnelle des données.

Des exemples d'ORM sont [nHibernate](http://nhibernate.info/) , [Entity Framework](http://msdn.microsoft.com/en-us/data/ef.aspx) , [Dapper](http://code.google.com/p/dapper-dot-net/) et plus ...

MongoDB est un exemple de base de données de documents - vous utiliseriez un ODM pour traduire entre vos objets dans le code et la représentation documentaire des données (si nécessaire).

[Mandango](https://github.com/mandango/mandango) est un exemple d'ODM pour MongoDB.

Un ORM mappe entre un modèle objet et une base de données relationnelle. Un ODM mappe entre un modèle objet et une base de données de documents. MySQL n'est pas un ORM, c'est une base de données relationnelle, plus précisément une base de données SQL. MongoDB n'est pas un ODM, c'est une base de données de documents.

Essentiellement, un ORM utilise un pilote de base de données SQL comme ODBC, JDBC ou OLEDB pour traduire la notation d'objet en notation relationnelle et un ODM utilise une API JSON ou JSONB pour traduire la notation d'objet en notation de document.

Il existe différents types d'implémentations sous le capot.

PS: JSONB est une notation de document texte JSON stockée dans un format binaire tel qu'utilisé par MongoDB.

01/09 :

1. ajout du middelware pour CORS
2. mise en place de la bdd mongoDB Atlas et de mongoose : new project : OC-P6\_piiquante (cluster name : OC-P6)
3. installation de mongoose dans backend :

À partir de la version 4.16 d'Express, bodyparser est inclus et vous

n'avez pas besoin de l'installer.

Utilisez ( express.json() ) pour analyser le corps de la requête.

A faire :

Étape 2 : Construire le parcours utilisateur

**30 %** d’avance

🎯**Partie du livrable :** API

📌**Recommandations :**

● Créez les éléments suivants :

○ Modèle d'utilisateur ;

○ Parcours utilisateur ;

○ Contrôleur d'utilisateur.

● L'utilisateur est en mesure d'effectuer les opérations suivantes :

○ Créer un compte ;

○ Se connecter et disposer d'un token valide.

● Consultez l'onglet réseau de Devtools pour plus d'informations.

⚠️**Les problèmes à connaître :**

● *Le mot de passe n'est pas haché.* Veillez à hacher le mot de passe.

● *Un utilisateur peut s'inscrire plusieurs fois avec la même adresse*

*électronique.* Assurez-vous que le code vérifie qu’une adresse

électronique est unique.

**Utilisateur**

**● email** : *String* — adresse e-mail de l'utilisateur **[unique]**

**● password** : *String* — mot de passe de l'utilisateur haché