



Web programming test

(A french version is available under the english version)

Description

The project provided to you for this test is called **OvaWebTest**. This is a simple Rest API to create new users, view their information and delete existing users. Upon receipt of the project, only a few features will be provided to you as a starting point (see **specifics and requirements**)

The project is divided into two parts, one part containing all the source code (Core) and the other part containing all the unit tests (Tests). A Swagger is built at application startup for API documentation purposes.

The project was built in Dotnet (C #) and uses the Dotnet Core 3.1LTS framework. We strongly recommend using Visual Studio 2019 (Community) tools for development and Postman for testing your HTTP requests.

Specifics and requirements

An initial implementation of the project is provided to you as a starting point. It contains the first user creation functionality, as well as all the unit tests associated with it. In addition, the signatures of the other functions to be programmed are provided to you.

Here is what is required of you to complete this test:

- Program all error handling for the user creation functionality.
- Program the entire functionality to fetch user information.
- Program the entire user deletion functionality.
- Set up a database and integrate it into the project in order to replace the “in memory” persistence layer already there, while respecting the software architecture in place.
- Develop a frontend (using React) which allows you to interact with the three basic functionalities of the backend, namely to be able to create users, see their information and delete existing users.
- Develop all the necessary unit tests in order to have adequate coverage of your code.

Additional notes

- You must respect the standards of software quality and clean code.
- HTTP codes returned by your API must meet industry standards.
- You are allowed to use the classes and functions already present in the project to complete your functionalities.
- You are allowed to add all the classes and private functions that you deem necessary.
- You are prohibited from modifying the unit tests already present in the project. In addition, they must all be able to pass when handing over your test.
- We want your frontend to be functional above all, so don't spend too much time with your CSS.
- We suggest you use PostgreSQL as your database. You can install the latest version for Windows 64 Bits at the following address:
<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>
- Instructions for using your database and frontend must be provided when submitting your test.

Installation and deliverable

For the development and delivery of your test, we ask you to create a gitlab repository. Make sure your code is correctly sent to the server when you're done (*commit* and *push*).



Test de programmation Web

Description

Le projet qui vous est fourni pour ce test est intitulé **OvaWebTest**. Il s'agit d'un simple Rest API permettant de créer des utilisateurs, voir leur information et supprimer des utilisateurs existants. Au moment de la réception du projet, seuls quelques fonctionnalités vous seront fournies en guise de point de départ (voir **spécificités et exigences**)

Le projet est divisé en deux parties, soit une partie contenant tout le code source (Core) et l'autre partie contenant tous les tests unitaires (Tests). Un Swagger est construit au démarrage de l'application à des fins de documentation de l'API.

Le projet a été construit en Dotnet (C#) et utilise le framework Dotnet Core 3.1 LTS. Nous recommandons fortement d'utiliser les outils Visual Studio 2019 (Community) pour le développement et Postman pour tester vos requêtes HTTP.

Spécificités et exigences

Une base du projet vous est fournie comme point de départ. Elle contient la première fonctionnalité de création d'utilisateurs, ainsi que tous les tests unitaires qui y sont associés. De plus, les signatures des autres fonctions à programmer vous sont fournies.

Voici ce qui vous est demandé pour la réalisation de ce test:

- Programmer toute la gestion d'erreurs pour la fonctionnalité de création d'utilisateurs.
- Programmer dans son entièreté la fonctionnalité d'obtention des informations de l'utilisateur.
- Programmer dans son entièreté la fonctionnalité de suppression d'un utilisateur.
- Mettre en place une base de données et l'intégrer au projet afin de remplacer la persistance mémoire déjà en place, tout en respectant l'architecture logicielle mise en place.
- Développer un frontend (avec React) qui permet d'interagir avec les trois fonctionnalités de base du backend, soit de pouvoir créer des utilisateurs, voir leur information et supprimer des utilisateurs existants.
- Développer tous les tests unitaires nécessaires afin d'avoir une couverture adéquate de votre code.

Notes supplémentaires

- Vous devez respecter les standards de qualité logicielle et de code propre (Clean Code).
- Les codes HTTP renvoyés par votre API doivent respecter les standards de l'industrie.
- Il vous est permis d'utiliser les classes et fonctions déjà présentes dans le projet pour compléter vos fonctionnalités.
- Il vous est permis de rajouter toutes les classes et fonctions privées que vous jugerez nécessaires.
- Il vous est interdit de modifier les tests unitaires déjà présents dans le projet. De plus, ils doivent tous être en mesure de passer avec succès lors de la remise de votre test.
- Nous voulons que votre frontend soit fonctionnel avant tout, donc ne passez pas trop de temps avec votre CSS.
- Nous vous suggérons d'utiliser PostgreSQL comme base de données. Vous pouvez installer la dernière version pour Windows 64 Bits à l'adresse suivante:
<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>
- Les instructions pour pouvoir utiliser votre base de données et votre frontend doivent être fournies lors de la remise de votre test.

Installation et livrable

Pour le développement et la livraison de votre test, nous vous demandons de créer un dépôt (repository) Gitlab. Assurez-vous que votre code est correctement envoyé sur le serveur lorsque vous aurez terminé (*commit* et *push*).