

Tests techniques frontend & backend

Table des matières

Introduction :	1
Description des professionnels :	1
Fonctionnalités attendues (sous forme d'API RESTful) :	2
Conditions du test, avertissement, délais :	2
Git :	2
Base de données :	3
API RESTful	3
Frontend.....	3
Documentation	3

Introduction :

Ce test a pour objectif d'évaluer les points suivants :

- Autonomie (ce qui n'empêche pas de demander des informations)
- Qualité du code (commentaires, lisibilité...)
- Utilisation basique du gestionnaire source git
- Conception et création d'une base de données PostgreSQL
- Création d'une API RESTful avec Spring-Boot en Java 8 ou plus
- Création d'un projet frontend utilisant l'API RESTful créée
- Documenter son travail

Ce projet consiste à créer une application de gestion des professionnels. Les utilisateurs peuvent créer, lire, modifier, supprimer des professionnels de leur application.

On part du principe que nous avons qu'un seul utilisateur pour diminuer le temps que vous consacrez à ce projet.

Description des professionnels :

- Nom
- Prénom
- Adresse e-mail (valider que c'est une adresse email : voir annotation dédiée en java)

- Un numéro de téléphone
- Une adresse (simple string)
- Une profession du domaine de la santé parmi : Médecin généraliste, Chirurgien, Infirmier, Kinésithérapeute, Assistante sociale, Psychologue. (Énumération)

Fonctionnalités attendues (sous forme d'API RESTful) :

- Lister tous les professionnels
- Rechercher des professionnels par : nom, profession
- Trier les professionnels par nom, prénom, profession
- Voir les détails/Ajouter/Modifier/Supprimer des professionnels

Il est possible de combiner les critères de tri et les critères de recherche.

La recherche doit pour le nom regarder si les noms contiennent le texte recherché par l'utilisateur.

Conditions du test, avertissement, délais :

Délais : Le délai est d'une semaine à compter de la réception de l'email.

Le projet doit bien évidemment être réalisé seul et par vos soins.

Le projet doit être réalisé en bonne intelligence :

- Le bête copié/coller est proscrit, mais le copié/coller intelligent, lui, est autorisé
- On ne réinvente pas la roue
- Il est possible (mais pas requis) d'apporter des améliorations au sujet. Laissez libre cours à votre imagination.

Git :

L'ensemble du projet doit être mis à disposition sur un gestionnaire de source Git (ex : GitHub, GitLab...) sur un repository qui sera rendu accessible à nos équipes (le repository peut être public).

Les dates des commits seront vérifiées pour valider le fait qu'il n'y ait pas de travail supplémentaire réalisé à la fin du délai imparti.

Base de données :

Créez un script SQL pour initialiser une base de données.

Un second script SQL sera créé pour remplir cette base de données de valeurs d'exemple.

Le moteur de base de données doit être Postgres.

API RESTful

Vous devez créer une API suivant la « norme » RESTful, en utilisant le framework Spring-Boot en Java 8 ou plus, avec une gestion des librairies en Maven.

Les erreurs renvoyées comporteront la bonne méthode HTTP associée à l'erreur ainsi qu'un message explicite.

L'api pourra être compilée au format JAR et lancée en tant que tel.

Il n'est pas nécessaire de mettre en place de l'authentification pour l'utilisation de votre API.

Frontend

Vous devez créer un projet frontend exploitant l'ensemble des endpoints mis à disposition par l'API que vous aurez développée. Le framework imposé : [Angular 10+](#).

Il vous est possible d'utiliser une librairie css (Ionic, Material Angular, Bootstrap).

Les erreurs de l'api seront gérées avec un message clair s'adressant aux utilisateurs.

La validation des objets est réalisée en front et en back.

Documentation

- Un manuel d'utilisation s'adressant à des développeurs désireux de tester votre projet sera fourni (cela peut être le README des repositories)
- Vous fournirez l'ensemble des requêtes pour tester toutes les fonctionnalités de votre projet sur Postman (pour cela vous pourrez utiliser la fonction d'export de collections sur Postman).