

Neuligent : Test technique

26 août 2021

Indications :

Le test est composé de 2 questions javascript, d'une question à développement, ainsi que 2 questions optionnelles en c++ et python.

Dans le cas des questions avec squelettes, un code source vous est disponible. Chacun d'entre eux sont compilables et permettent de tester vos réponses. Des indications sur comment les compiler seront transcrits dans chaque section.

Si vous ne comprenez pas une question et ne pouvez pas y répondre, il n'y a pas de problème. Décrivez tout simplement votre analyse faite sur la problématique et les détails qui vous ont bloqués.

Ce test peut être fait avec l'aide de tout outils. Seul détail : si vous vous inspirez ou vous utilisez du code pris en ligne, veuillez noter sa source.

Question 1 : JS front end

Dans le dossier Front-end, vous pouvez trouver un dossier pour un site web client programmé en Javascript avec le framework React.

Le site web contient des champs d'entrées, un bouton et un formulaire de texte. Les deux champs d'entrées doivent pouvoir contenir des chiffres. (Pour fins de test avec l'api choisi, seuls les chiffres de 1 à 10 seront considéré important comme valeur en entrée.)

On cherche à implémenter la fonctionnalité suivante :

Lorsque le bouton est cliqué, le système doit envoyer 2 requêtes HTTP sous format suivant : url : `https://reqres.in/api/users` type : GET data : `'id' : var1` ou `var2`

où var1-2 sont les valeurs reçues en entrée dans le formulaire.

Lorsque les deux réponses du serveur ont été reçues, le système doit extraire un champ de chacune des réponses, nommé ***first_name***, afin de concatener les deux valeurs reçues et d'afficher leur concaténation dans le champ de texte, nommé answer.

FIGURE 1 – Question 1

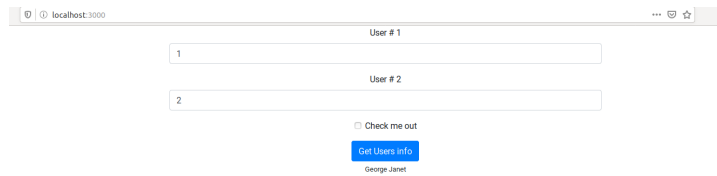


FIGURE 2 – Retour attendu du programme pour les valeurs 1 et 2.

Exemple : Lorsque le système reçoit les valeurs 1 et 2 en entrées, la page web devrait afficher George Janet.

Installation et démarrage :

Pour installer et démarrer le site sur ubuntu, il suffit d'effectuer les commandes suivantes à partir de la racine du dossier :

- npm install
- npm start

Le code à modifier est situé dans le fichier **App.js**, qui est dans le dossier **src**.

Question 2 : JS back end

Écrivez, par le biais de **NodeJS**, une application web qui écoute sur le port 8000 pour recevoir des messages afin de pouvoir les chiffrer et déchiffrer (encrypt and decrypt.)

Pour ce faire, cette application doit pouvoir recevoir des requêtes GET aux urls

- `http://localhost:8000/api/encrypt`
- `http://localhost:8000/api/decrypt`

avec les paramètres suivants :

encrypted : "encrypted_message", key : key_number.

encrypt doit pouvoir chiffrer le message reçu et **decrypt** doit pouvoir le déchiffrer.

Le chiffrement et le déchiffrement des messages doit être implémenté par le chiffrement Caesar, qui est un décalage de chaque lettre du message n fois, où n est la clé reçue en paramètre de la requête. Veuillez prendre note que Caesar est calculé de la manière suivante : $\text{lettre}_{\text{encrypte}} = \text{lettre} + \text{key} \bmod 26$.

Par exemple :

`/api/encrypt`

Parametre :

encrypted_message : Message de test

key : 2

Transformation

Message de test

$M + 2 = o$

$e + 2 = g$

$s + 2 = u$

$s + 2 = u$

$a + 2 = c$

```

etc ...
Retour :
Oguucig fg vguv

/api/decrypt
Parametre :
    encrypted_message : Oguucig fg vguv
    key : 2
Transformation
Message de test
O - 2 = M
g - 2 = e
u - 2 = s
u - 2 = s
c - 2 = a
etc ...
Retour :
Message de test

```

Question 3 : Cas d'utilisation

Cette question est une question à développement. Pour y répondre, inscrivez dans un fichier texte, nommé question3.txt ou question3.docx, les étapes que vous suivriez pour résoudre le problème.

Vous êtes un(e) développeur(euse) en charge de la maintenance d'une application web composé d'un front-end en NodeJS et d'un backend hybride en NodeJS,PHP et Python suivant l'architecture suivante : Backend : NodeJs (Gateway et fonctions de connexion) -> PHP (Fonctions d'interaction avec les bases de données) -> Python (Fonctions d'analytiques).

Un jour, votre superviseur vous demande de se débarrasser du serveur PHP, citant des fautes de sécurités, et de recréer ses fonctions soit dans le serveur NodeJS ou le serveur Python, selon votre choix.

Quelle serait votre approche, décrites en étapes, pour réaliser ce projet, afin d'assurer les points suivants ? (Veuillez prendre note dans ce scénario que vous n'avez pas de limitations sur les facteurs de temps et d'outils technologiques à votre disposition.)

- Qualité
- Robustesse
- Assurance de la qualité de transition
- Stabilité du code

Question 4 : C++ (Question optionnel)

Dans le dossier c++/src, vous pouvez trouver un fichier dénommé main.cpp avec 2 squelette de fonction. La première fonction prend en entrée une liste d'éléments et doit la ré-ordonner afin d'avoir une alternance entre l'ordre initial de la liste et son ordre inverse.

Exemple :

A-B-C-D-E-F -> A-F-B-E-C-D

1-4-2-3-7 -> 1-7-4-3-2

q-w-e-r-t-y-u -> q-u-w-y-e-t-r

La seconde fonction reçoit en entrée deux listes chaînées non-vides représentant deux chiffres positifs. Chacune des listes détient les chiffres en ordre inverse, c'est-à-dire par exemple que le chiffre 715 est stocké sous forme de liste 5-1-7. Compléter la fonction afin d'additionner les deux chiffres et de retourner son résultat sous forme d'une nouvelle liste chaînée.

Installation et démarrage :

Pour compiler et exécuter le programme, il suffit d'utiliser les commandes suivantes sur ubuntu :

- make
- ./build/apps/test x y
- x et y sont les deux chiffres utilisé par la fonction 2 en entrée.

Question 5 : Python (Optionnel)

Le fichier test.py détient une implémentation d'un arbre binaire de recherche. Ajouter les opérateurs manquants suivants :

1. +
2. =
3. ==