

Projet - WebMobile (petits gazouillis) 420-P5C 20% de la note finale
--

Description

Créer un projet web mobile offrant les mêmes fonctionnalités que celles offertes par le projet `petit gazouillis`. Ce projet comprend une application web mobile développée dans un environnement `expo/React Native` et un serveur `REST API` développé à partir du projet `petits gazouillis`.

L'application web mobile devra fonctionner en version `web`. De plus, l'interface de l'application web mobile devra montrer les autres utilisateurs qui sont "en ligne" et s'actualiser quand un changement survient. Exemple de changement: établir une session, terminer une session, nouvel utilisateur, nouvelle publication, modification profil, modification publication, suppression utilisateur, suppression publication, suivre, ne plus suivre.

- Ce projet se fait seul.
- Présentation de l'énoncé : 23 octobre 2020
- La date de remise est lundi 7 décembre 2020 avant 23H59 (- 10% par jour à partir de 23h59) .
- Vous devrez remettre votre projet `petits_gazouillis_par_votre_nom.zip` sur LÉA contenant une application `React Native` dans un dossier `web_mobile` et un serveur `REST API` dans un dossier `serveur_rest_api`.
- Afin de faciliter la correction, vous devrez lors d'une rencontre Zoom démontrer que votre projet respecte les critères de correction (durant la semaine 15 et 16)
- La correction du projet se fera à partir de votre projet (partage d'écran).
- Puisque ce projet est réalisé pour la première fois, il est possible que des modifications y soient apportées en cours de route.
- L'exploration est un des objectifs de ce projet. Vous devrez démontrer une très grande autonomie dans la réalisation du projet. Cependant, sur demande de votre part, je pourrai produire des capsules vidéo sur des aspects spécifiques.

Par où débiter? Je vous suggère ce tutoriel <https://docs.expo.io/> (Slow start)

PARTIE 1- (35%) Développer un serveur REST API

En se basant sur le projet `petits_gazouillis`, ce serveur aura les caractéristiques suivantes:

A => Gérer les jeton

(5%) `/api/jeton`

POST - demander un jeton avec `utilisateur:mot de passe`

DELETE - supprimer un jeton avec `jeton`

B => Gérer les utilisateurs

(5%) `/api/utilisateur`

GET - Lister les informations d'un utilisateur (`<id>`) avec `jeton`

GET - Lister les informations des utilisateurs avec `jeton`

POST - Créer un utilisateur - sans `jeton`

PUT - Modifier votre profil avec `jeton` spécifique

SUPPRIMER - Supprimer votre compte (pas celui d'un autre) avec `jeton` spécifique

GET - Suivre un utilisateur avec `jeton` spécifique

GET - Ne plus suivre un utilisateur avec `jeton` spécifique

C => Gérer les publications

(5%) `/api/publications`

GET - Lister les informations d'une publication (`<id>`) avec `jeton`

GET - Lister les informations des publications avec un `jeton`

POST - Créer une publication avec un `jeton` spécifique

PUT - Modifier votre publication (pas celle d'un autre) avec `jeton` spécifique

SUPPRIMER - Supprimer votre publication (pas celle d'un autre) avec `jeton` spécifique

* `jeton` spécifique appartient à un utilisateur spécifique. C'est à vous de déterminer comment réaliser ceci.

D => Gérer les erreurs

(5%) Gestion de tous les messages d'erreurs par des flux `JSON` contenant les messages d'erreurs.

E => Gérer les événements demandant une actualisation de l'interface utilisateur.

(5%) Les utilisateurs sont informés de toutes modifications par des messages transmis par `websocket`.

F => Tester votre API

(10%) Un module de test permet de vérifier l'ensemble des fonctionnalités offertes par les APIs `/api/jeton`, `/api/utilisateur`, `/api/publications`

Note : Vous êtes libre d'ajouter toutes fonctionnalités que vous jugerez nécessaires à la réalisation du projet.

PARTIE 2- (65%) Développer une application web mobile

- Vous êtes libre de déterminer le format de l'interface. Vous serez jugés sur la convivialité et l'ergonomie de l'interface.
- L'application web mobile est obligatoirement un projet expo REACT NATIVE
- L'application web mobile doit obligatoirement fonctionner en version Web
- L'application web mobile est obligatoirement informée qu'elle doit actualiser son interface par des messages transmis par websocket

Votre application web mobile devra avoir les caractéristiques suivantes:

A => Actualisation

(20%) Événements qui génèrent la transmission des messages d'actualisation

Pour chacun de ces événements, votre application doit réagir en actualisant l'affichage avec l'intervention la plus minimale possible. Par exemple : Le contenu d'une publication est modifié, alors uniquement le contenu de cette publication est actualisé sur l'affichage. Pas l'ensemble des publications.

1. Établir une session
2. Terminer une session
3. Nouvel utilisateur
4. Supprimer utilisateur
5. Nouvel publication
6. Modification d'une publications
7. Suppression d'une publication
8. Modification d'un profil
9. "Suivre" ou "Ne plus suivre"

B => Fonctionnalités

(40%) Votre application devra offrir les fonctionnalités ci-dessous. C'est à vous de déterminer de quelles manières ces fonctionnalités seront offertes à l'utilisateur par l'intermédiaire de l'interface.

1. Établir une session
2. Terminer une session
3. Créer un nouvel utilisateur
4. Supprimer un utilisateur (uniquement notre compte, pas celui d'un autre!)
5. Publier un message

6. Modifier une de nos publications (pas celle d'un autre)
7. Supprimer une de nos publications (pas celle d'un autre)
8. Afficher les informations tous les profils
9. Modifier les informations de notre profil
10. "Suivre" ou "ne plus suivre" un utilisateur
11. L'affichage les listes sur plusieurs page et suivant, précédent
12. Afficher une liste de messages (filtrer: mes messages, messages suivis par des utilisateur, tous les messages, etc.)

C => Erreurs

(5%) Gérer avec élégance les erreurs

En cas d'erreur, l'interface réagit avec "élégance". Voici une liste d'erreurs auxquelles votre interface doit réagir.

1. établir une session: utilisateur existe pas
2. établir une session: erreur mot de passe
3. création utilisateur: utilisateur existe déjà
4. modifier utilisateur: nouveau nom existe déjà

Voilà!