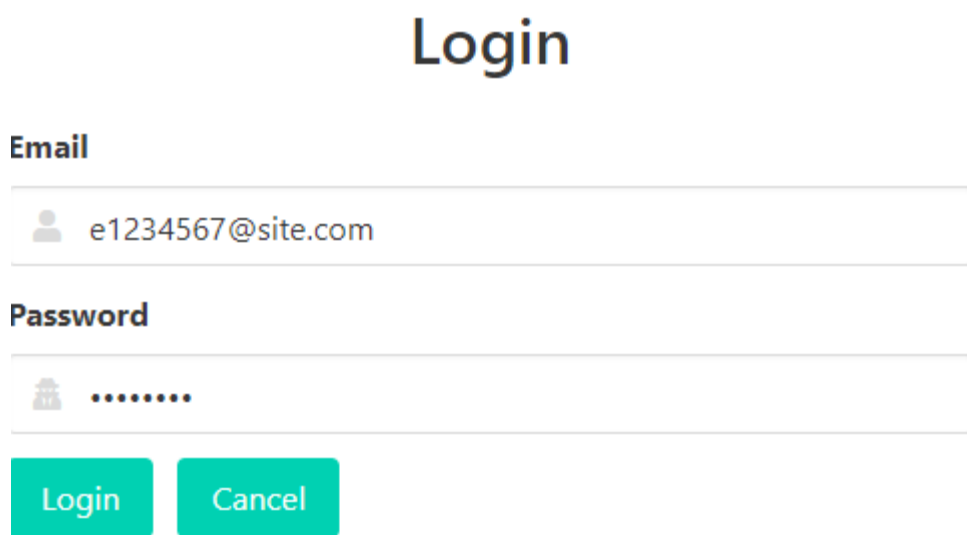


Objectifs

- Écrire une application React permettant aux utilisateurs et utilisatrices de consulter et écouter les podcasts disponibles à travers l'API.

Login

- Permet à un/une utilisateur/utilisatrice de s'authentifier.
- Utilisez e1234567@site.com/e1234567 comme nom d'utilisateur et mot de passe.



The image shows a login form with a title 'Login' in a large, bold, orange font. Below the title, there are two input fields. The first field is labeled 'Email' and contains the text 'e1234567@site.com' with a user icon on the left. The second field is labeled 'Password' and contains a series of dots with a lock icon on the left. At the bottom of the form, there are two buttons: a green 'Login' button and a light blue 'Cancel' button.

- Utilisez la méthode [/auth/token](#) de l'api pour obtenir un jeton.
- Vous devez envoyer, dans le BODY de la requête, les données d'authentification

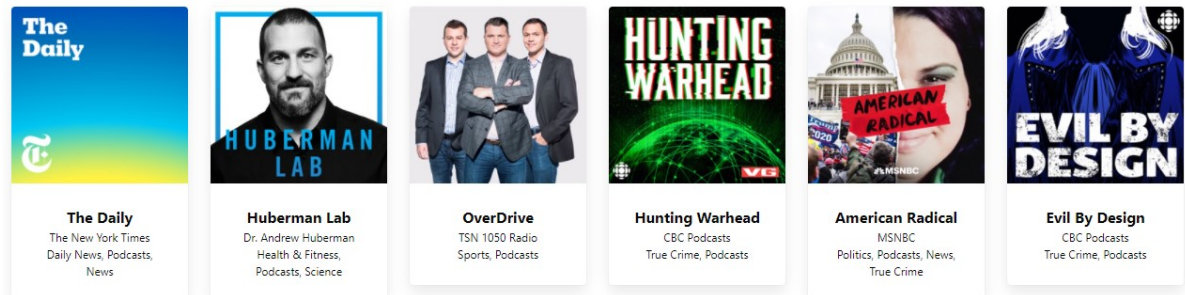
```
{  "email": "e1234567@site.com",  "password": "e1234567"}
```
- Vous recevrez le résultat sous le format suivant:

```
{  "token": "eyJhbGciOiJI[...]"}
```
- Notez que l'api est écrite en nodejs/express, et conséquemment, vous n'utilisez pas le header url-form-encoded pour obtenir un jeton.
- Un message d'erreur est affiché sur l'authentification échoue.
- Après une authentification réussie, l'utilisateur est redirigé vers la page subscriptions.

Subscriptions

- La page des abonnements (subscriptions) permet de visionner les abonnements de l'utilisateur.

Subscriptions



- La liste des abonnements est obtenue en appelant la méthode [/subscriptions](#) (méthode GET). C'est une méthode authentifiée (nécessite un bearer token).
- Les données retournées ont le même format que ceux de la méthode [/podcasts/top](#).

Ajouter/supprimer abonnements

- Dans le fragment où on trouve les informations détaillées concernant un podcast, ajoutez un (ou des) boutons permettant d'ajouter ou de supprimer un abonnement.
- Pour modifier les téléséries favorites de l'utilisateur présentement authentifié, on utilise l'api `/subscriptions?id=X` où X est le podcastId du podcast.
- Ceci est une api authentifiée, elle nécessite un jeton d'authentification bearer token pour fonctionner.
- Appelez `/subscriptions?id=X` avec la méthode GET retourne un json contenant la propriété `isSubscribed`. C'est une propriété booléenne.

```
{"isSubscribed":true} ou {"isSubscribed":false}
```

- Un appel à `/subscriptions?id=X` avec la méthode POST permet d'ajouter le podcast dont le id est spécifié en paramètre aux abonnements de l'utilisateur.
- Un appel à `/subscriptions?id=X` avec la méthode DELETE permet de supprimer le podcast dont le id est spécifié en paramètre aux abonnements de l'utilisateur.
- Lorsque l'utilisateur s'abonne ou se désabonne, la page courante reste ne change pas, mais le bouton approprié (abonnement ou désabonnement) est affiché.

Meilleures pratiques

- Un maximum de points sera accordé aux applications écrites à l'aide des meilleures pratiques (réutilisation de composants, clarté du code, pas de warnings ou erreurs REACT, etc.).