




Applications web et mobiles

TP n°13 : La page de login du forum Ionic

À partir de maintenant, vous allez porter votre forum sur smartphone en utilisant *Ionic*, lui-même exploitant un *framework Angular*. Les parties TypeScript sous *Ionic* et *Angular* vont être très similaires et, pour cela, vous aurez principalement à faire des copiers/collers de votre forum *Angular* (vous pourrez utiliser indifféremment celui communiquant avec le *backend* en PHP ou en node/express. Mais vous devrez toujours copier/coller des morceaux du même *frontend* vers votre nouveau forum *Ionic*). Pour la partie HTML, on pourrait faire de même, mais cela n'aurait pas le *look and feel* des applications mobiles. Vous aurez donc à adapter votre code *Angular*. Enfin, vous testerez votre nouveau forum dans votre navigateur.

Étape 59 – Création du projet Ionic

Créez votre projet Ionic. Choisissez un projet « blank » fondé sur *Angular* en mode **standalone**.

 Si votre installation « freeze » sur l'instruction « `npm i --save -E @capacitor/core@latest` », c'est-à-dire que rien ne « bouge » pendant plusieurs minutes, exécutez les instructions suivantes :

1. Supprimez le répertoire créé par le « `ionic start` » qui a « freezez ».
2. Recréez le en rajoutant l'option « `--cordova` » au « `ionic start` ».
3. Faites un « `cd` » pour vous déplacer dans votre projet.
4. Exécutez les trois instructions :

```
npm install @capacitor/core
npm install -D @capacitor/cli
npx cap init
```

et répondez aux questions qui vous sont posées.

Étape 60 – Initialisation du projet

Supprimez le répertoire contenant la page « home » et créez une nouvelle page « login ». Faites en sorte que la route par défaut pointe, non plus sur home, mais sur login.

Étape 61 – Les services de messages et d'authentification

Démarrez votre serveur *backend*. Créez deux services `message` et `auth` et recopiez dedans tout le code que vous aviez écrit pour ces services dans votre forum « *Angular* ».



N'oubliez pas de rajouter la directive `provideHttpClient()` afin que vos deux services fonctionnent correctement : ici, ce n'est pas dans le fichier `app.config.ts` mais dans `main.ts` qu'il faut le faire.

Étape 62 – La page de login

Copiez-collez dans les parties TypeScript et HTML (dans la zone `<ion-content></ion-content>`) de votre page `login` tout ce que vous aviez écrit pour ce composant dans votre forum « *Angular* ».

Créez une nouvelle page pour la page `cours`, qui permettra d'afficher la liste des cours (pour pouvoir faire un maximum de copiers/collers, il est important de donner aux pages que vous créez dans *Ionic* les mêmes noms que les composants correspondants de votre forum *Angular*). Pour l'instant, laissez cette page telle qu'elle est créée par *Ionic*.

Testez que vous pouvez bien vous logger. Pour l'instant, ne vous préoccupez pas du style de la page.

Étape 63 – Le look and feel de la page de login

Dans le `<ion-header></ion-header>`, ajoutez l'attribut `color="primary"` à la balise `<ion-toolbar>`. Remplacez les balises `<input>` par des balises `<ion-input>` (cf. <https://ionicframework.com/docs/api/input>). Utilisez l'attribut `placeholder` des `<ion-input>` afin de demander à l'utilisateur de saisir son login ou password : cela permet d'éviter d'avoir, au dessus du `<ion-input>`, un label demandant la même chose, d'où un gain de place sur l'écran (de petite taille du smartphone).

Maintenant faites en sorte que les balises `<ion-input>` soient les éléments (`<ion-item></ion-item>`) d'une `<ion-list>`. Pour voir comment fonctionnent ces listes, vous pouvez vous reporter à la documentation de *Ionic* : <https://ionicframework.com/docs/components>.

Transformez le bouton de connexion en `<ion-button>` et placez-le dessous de la liste. Vous pourrez utiliser l'attribut `color="primary"` pour lui donner une apparence plus sympathique.

Étape 64 – La gestion des erreurs de connexion

Dans le TypeScript de votre `LoginComponent`, créez une méthode `openAlert` afin d'ouvrir un message d'alerte, comme vu dans le cours. Faites en sorte que, si votre *backend* vous a renvoyé une erreur, cette alerte l'affiche.