

B2 - Introduction to Web Development

B-WEB-200

Workshop

Création d'un messenger avec Ionic 4 et socket.io





Workshop

language: Html - Css - TypeScript

IONIC

Ionic est un framework open-source créé en 2013 par Max Lynch, Ben Sperry et Adam Bradley. Basé initialement sur AngularJS et Apache Cordova2, Ionic permet de créer un code multisupport en utilisant des outils Web comme HTML, CSS, JavaScript, afin de générer des applications iOS, Android, Chrome, Windows Phone et bien d'autres.

Ionic permet de construire des applications natives en utilisant les technologies web.

Par natives on doit comprendre : disponibles via les stores et pouvant utiliser les fonctionnalités natives des OS.

Ionic nous permet de créer nos applications avec les frameworks Angular, React ou bientôt Vue.

La puissance de ionic est de nous proposer une librairie complète de composants graphiques à utiliser dans notre application.

ANGULAR

Angular est un framework coté client opensource basé sur typescript dirigée par l'équipe du projet Angular de GOOGLE et par une communauté de particuliers et de sociétés. Angular est une réécriture complète de AngularJS, framework construit par la même équipe.



INSTALLATION

Etape 1, NPM :

Avant de pouvoir développer notre messenger, nous allons d'abord télécharger nos outils. Tout d'abord il vous faudra installer npm.

```
Terminal
~/B-WEB-200> sudo apt install npm
```

Etape 2, Ionic:

L'étape intermédiaire est simple, il vous suffit d'exécuter la commande suivante :

```
Terminal
~/B-WEB-200> sudo npm install -g @ionic/cli
```

Etape 3, Clone Project

Maintenant, je vous propose de télécharger le projet, je vous propose donc d'exécuter la commande suivante.

```
Terminal
~/B-WEB-200> git clone https://github.com/Puigs/Serveur-messenger-Socket.io.git
```

Etape 4, Socket.io

Maintenant afin de finaliser l'installation de nos outils il vous suffit de vous placer dans le dossier Serveur-messenger-Socket.io/Front et d'exécuter la ligne de commande suivante :

```
Terminal
~/B-WEB-200> npm install ngx-socket-io
```

Si Erreur:

Si après ces commandes vous ne pouvez pas lancer Ionic et que vous obtenez le message suivant :

```
Terminal
~/B-WEB-200> ERR: Your Node.js version is v8.10.0. Node.js 8 reached end-of-life on 2019-12-31 and is no longer supported. Please update to the latest Node LTS version.
```

Vous devez faire les commandes :

```
Terminal
~/B-WEB-200> curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
```

Puis



```
Terminal
~/B-WEB-200> sudo apt-get install -y nodejs
```

Si Erreur 2, le retour

Si après ces commandes vous ne pouvez pas lancer Ionic et que vous obtenez le message suivant :

```
Terminal
~/B-WEB-200> [ng] An unhandled exception occurred: Could not find module
"@angular-devkit/build-angular" from "/home/puigs/project/Workshop/test/Serveur-mess
[ng] See "/tmp/ng-kJj03E/angular-errors.log" for further details.
[ERROR] ng has unexpectedly closed (exit code 127).

The Ionic CLI will exit. Please check any output above for error details.
```

Vous devez faire la commande suivante :

```
Terminal
~/B-WEB-200> npm install --save-dev @angular-devkit/build-angular
```

UTILISATION

Pour lancer le projet il vous suffit de vous rendre dans Serveur-messenger-Socket.io/Front et d'effectuer la commande suivante :

```
Terminal
~/B-WEB-200> ionic serve
```

La page 1 est le home, nous n'y toucherons pas.

La page 2 est notre chat.

La partie 3 est vide.



PARTIE 1, IONIC 4

Pour lancer le programme vous devez executer dans le dossier front la commande suivante :

```
Terminal
~/B-WEB-200> ionic serve
```

L'exercice est le suivant :

Modifier le pseudo de l'utilisateur!

Pour cela nous allons modifier trois pages :

src/app/app.module.ts

src/app/tab2/tab2.page.html

src/app/tab2/tab2.page.ts

src/app/tab2/tab2/module.ts

Voici quelques petites explication :

Le fichier app.module.ts est notre fichier contenant nos imports, il s'agit d'un NgModule.

Le fichier tab2.page.html est le fichier de notre deuxième page de l'application, comme vous pouvez vous en douter il s'agit de l'html de notre page.

Le fichier tab2.page.ts est notre fichier contenant notre js, ou plutôt dans notre cas le typescript.

Le Typescript est une surcouche de js, le code typescript est transcompilé en js.

TypeScript permet un typage statique optionnel des variables et des fonctions, la création de classes et d'interfaces, l'import de modules, tout en conservant l'approche non-contraignante de JavaScript.

Première étape

Tout d'abord nous allons rajouter à notre fichier app.module.ts le code suivant

```
import { ReactiveFormsModule } from '@angular/forms';
import { Validators, FormBuilder, FormGroup, FormControl } from '@angular/forms';
```

Puis transformer une ligne pour qu'elle devienne comme ceci :

```
imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule,
  SocketIoModule.forRoot(config), ReactiveFormsModule],
```

Pour l'instant c'est simple non? Attendez de voir la suite ;) (Rire maléfique)

Partie 2

Maintenant, allons modifier le fichier tab2.page.ts

Si nous voulons modifier le pseudo de l'utilisateur, nous aurons besoin d'un formulaire afin que l'utilisateur puisse choisir son pseudo.

Cependant, ça sera beaucoup trop simple de seulement enregistrer la valeur d'un input. C'est pourquoi on va rajouter un système de validation à notre formulaire.

Tout d'abord, ajoutons les imports

```
import { Validators, FormBuilder, FormGroup, FormControl, FormArray } from '@angular/forms';
```

Faites de même pour tab2.module.ts et modifiez le fichier pour qu'il ressemble à ceci :



```
imports: [  
  IonicModule,  
  CommonModule,  
  FormsModule,  
  ReactiveFormsModule,  
  RouterModule.forChild([{ path: '', component: Tab2Page }])  
],  
declarations: [Tab2Page]
```

Maintenant, dans le fichier tab2.page.ts, ajoutez les variables suivantes :

```
data: FormGroup;  
pseudo: any;  
check : Boolean = false;
```

N'hésitez pas à ajouter ou enlever des variables selon votre goût.

Partie 3

Dans cette partie vous devez modifier le fichier tab2.page.html et le fichier tab2.page.ts afin d'y intégrer un formulaire.

Pour cette partie, je ne vous aiderai pas, ce lien le fera pour moi ;)

<https://ionicframework.com/docs/v3/developer-resources/forms/>

Il contient des explications et des **exemple** vous permettant de réussir cette exercice.

Voici un bout de code permettant de récupérer la valeur de votre input :

```
this.pseudo = this.data.value;
```

Conseil

Pour tester votre code, utilisez console.log().

Partie 4

Vous avez dû le remarquer mais le côté client du chat se lançant en même temps que votre formulaire impossible d'utiliser votre nouveau pseudo.

Je vous conseille donc de garder votre formulaire dans le ngOnInit et de déplacer code client serveur dans une nouvelle fonction.

Pour pouvez utiliser le code suivant à partir de votre bouton afin de lancer le chat.

```
(click)="fonction()"
```

Partie 5

La dernière partie est la plus simple, pour gardez notre page propre il sera sympa d'afficher à l'écran seulement le formulaire ou seulement le chat.

La dernière variable ajouté précédemment devrai vous être utile.