
IMPLEMENTATION D'UNE APPLICATION DE GESTION DE STOCKS AVEC IONIC

TRAVAUX PRATIQUES

ÉCRIT PAR
NDAYOU RONALD
VINCENT

3IAC - Institut Universitaire de la Côte

\mathcal{PL}

2021-2022
NDAYOU R. VINCENT

Contents

1	Mise sur pieds de l'environnement	1
1.1	Installation de Nodejs	1
1.2	Installation de ionic	1
1.3	Installation de Cordova	1
1.4	Installation de native-run	1
1.5	Installation de firebase	1
2	Implémentation	2
2.1	Intégration de Firebase	2
2.1.1	Création du projet	2
2.1.2	Intégration au projet	2
2.1.3	Enregistrement des produits dans firestore	2
2.2	Amélioration des visuels	3
2.2.1	La page d'accueil (home)	3
2.2.2	Affichage de la météo	5
2.3	Sauvegarde en Local des coordonnées	5
3	Déploiement	6
3.1	Exécution sur le navigateur	6
3.2	Déploiement sur android	6

1 Mise sur pieds de l'environnement

1.1 Installation de Nodejs

Nodejs est un environnement de développement d'application javascript basé sur la machine virtuelle V8. Son installation dépend du système d'exploitation hôte:

- Sur linux (Ubuntu): il est installé via la commande ***apt install nodejs***
- Sur Windows il suffit d'ouvrir le lien ***<https://nodejs.org/en/download/>*** dans un navigateur, choisir la version de node, télécharger l'exécutable et l'installer. Il installe aussi son gestionnaire de package ***npm*** qui nous permettra l'installation des dépendances nécessaires pour notre application.

1.2 Installation de ionic

Ionic est un framework de développement d'application cross-platform avec le langage javascript. Il se sert d'autres frameworks de développement front-end tels que angular ou reactjs eux même basés sur javascript pour le développement de ses interfaces. Il est installé via l'utilitaire de gestion des packages npm avec la commande:

npm install -g @ionic/cli

L'argument -g permet de l'installer globalement dans notre OS.

1.3 Installation de Cordova

Cordova est un framework open-source qui nous permet de développer et transformer nos applications HTML, CSS et Javascript en application native. Il est installé via la commande:

npm install cordova

1.4 Installation de native-run

native-run est un utilitaire qui nous permet d'exécuter et déployer nos applications sur des terminaux android ou ios ou sur les simulateurs. Il s'installe avec la commande suivante:

npm install -g native-run

1.5 Installation de firebase

Firebase est un service d'hébergement d'application qui nous offre la possibilité de créer des bases de données temps réel, de gérer l'authentification, les notifications et bien d'autres services. Pour l'utiliser dans notre application ionic, nous devons installer les dépendances ***fire*** et ***@angular/fire***:

- ***npm install fire --save***
- ***npm install @angular/fire***

2 Implémentation

2.1 Intégration de Firebase

2.1.1 Création du projet

- Se connecter à son compte via le lien *firebase.google.com/console*
- Cliquer sur le bouton *Get Started*
- Ajouter un nouveau projet
- Ajouter une nouvelle application Web au projet.
- Copier les identifiants du nouveau projet

2.1.2 Intégration au projet

Une fois le projet créée sur firebase et les identifiants copiés, nous devons connecter notre application à firebase. Nous ferons ceci dans le fichier *app.modul.ts*. Mais avant, nous devons enregistrer les paramètres du projet firebase dans les fichiers *environments.ts*. Grace au module **AngularFireModule** de la dépendance **@angular/fire** installé plus tôt nous allons nous connecter à notre application.

```
AngularFireModule.initializeApp(environment.firebaseConf) ;
```

où *firebaseConf* est l'object qui contient nos identifiants firebase dans le fichier *environment.ts*.

2.1.3 Enregistrement des produits dans firestore

le service: products.service.ts Dans le service, nous devons injecter le module **AngularFirestore**

```
contructor(private firestore: AngularFireStore)
```

Nous devons ajouter à la fonction **addProduct** l'instruction **return this.firestore.collection('products').add(product)**. Cette instruction retourne une promesse que nous devons exploiter lors de son appel dans la page *new-product.page.ts*. Cette page dispose d'une fonction **createProduct()** qui est invoqué au click sur le bouton du formulaire; elle verifie la conformité des données saisies et affiche un **Toast** en cas d'erreur. Si la sauvegarde est réussie, elle nous renvoie vers la page d'accueil.

Au formulaire d'enregistrement des produits nous ajoutons un bouton qui permet d'ajouter une image au produit avec un preview. Après le choix d'une image, le chemin de l'image est extrait et enregistré avec le produit.

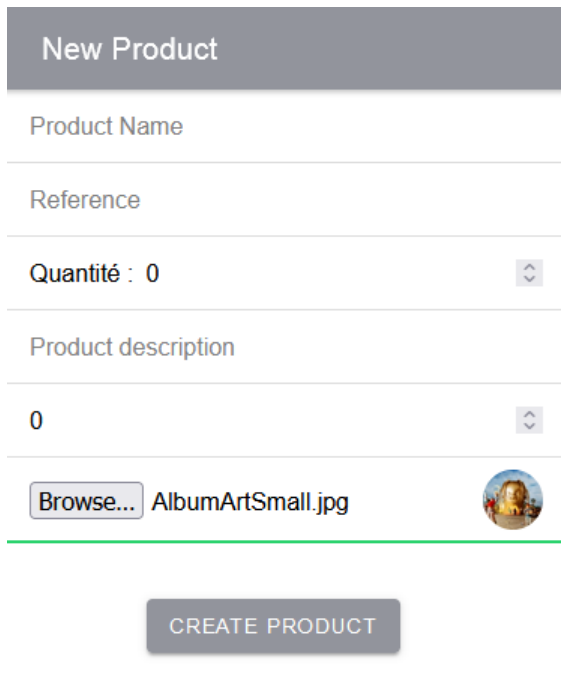
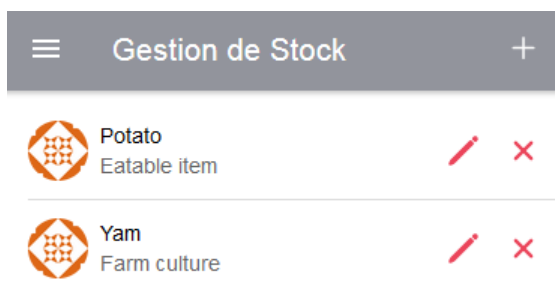


Figure 1: Ajout d'un produit

2.2 Amélioration des visuels

2.2.1 La page d'accueil (home)

Cette page liste les articles stockés dans nos bases de données. Les informations sont extraites de la base de données avec la fonction **•getProducts()**. Nous décidons d'associer une image aux produits et de changer la vue pour avoir le rendu ci-dessous:









Gestion de Stock			
	Potato	Eatable item	 
	Yam	Farm culture	 

Figure 2: Ajout d'un produit

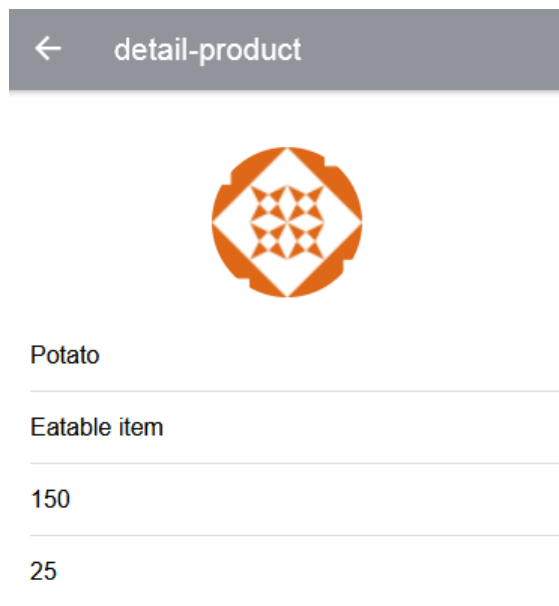


Figure 3: Détails des produits

2.2.2 Affichage de la météo

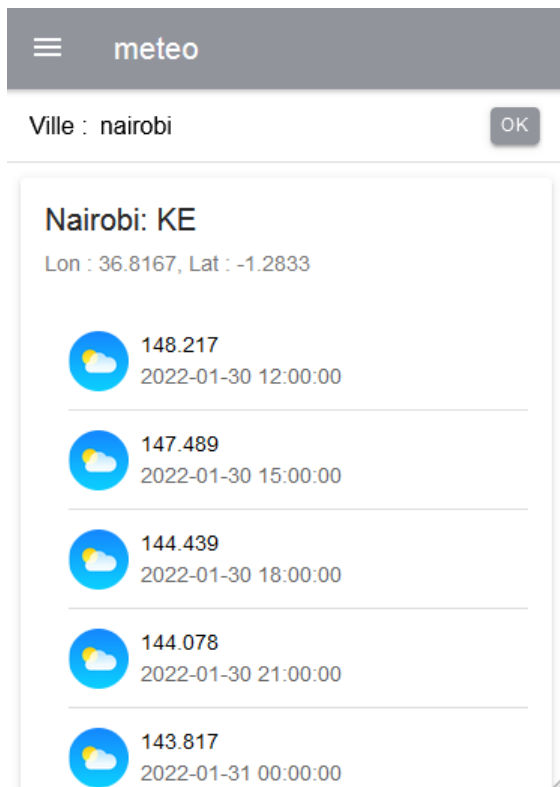


Figure 4: Affichage de la météo

2.3 Sauvegarde en Local des coordonnées

Les coordonnées gps (longitude et latitude) sont stockés en local grace à la fonction ***setItem()*** du module ***nativeStorage***

3 Déploiement

3.1 Exécution sur le navigateur

l'exécution de notre application sur le navigateur est faite par la commande **ionic serve**. Cependant cette commande affiche une erreur indiquant que cordova n'est pas disponible dans notre environnement. Pour corriger cette erreur, nous pouvons utiliser les commandes

- **ionic cordova platform add browser** qui ajoute la possibilité pour cordova de s'exécuter sur le navigateur
- **ionic cordova run browser** compile et exécute l'application sur notre navigateur.

3.2 Déploiement sur android

Grace à la commande **ionic cordova run android** notre application est compilée et exécuter utilisant la sdk installée sur notre machine. Si la sdk est absent on peut utiliser l'option **ionic cordova run android -no-native-run**