

# Paramétrages des clients

- [Nom de l'app, bundle ID et Version](#)
- [Chemins vers le backend](#)
- [Modules](#)
- [Langues](#)
- [Titre de l'application et numéro de version](#)
- [Firebase pour les notifications](#)

## Nom de l'app, bundle ID et Version

Dans le fichier `/dev/user-frontend-ionic/capacitor.ts` paramétrer :

- `appId` : Un identifiant unique de l'application qui devra correspondre à celui déclaré sur les stores
- `appName` : Le nom de l'application

```
const config: CapacitorConfig = {
  appId: 'fr.esupportail.mobile.multi',
  appName: 'Esup-Multi',
```

## Chemins vers le backend

Dans le fichier `/dev/user-frontend-ionic/src/environments/environment.ts` paramétrer :

- `apiEndpoint` : l'URL du backend. Il s'agit de celle du `µService main`.
- `cmsPublicAssetsEndpoints` : l'URL des assets (images) saisie au travers du `CMSHeadless` et adressable publiquement.

```
export const environment = {
  [...]
  apiEndpoint: 'http://localhost:3000',
  cmsPublicAssetsEndpoint: 'http://localhost:8055/assets/',
  [...]
};
```

## Modules

voir le détail pou chaque module [Paramétrage des modules](#)

## Langues

Dans le fichier `/dev/user-frontend-ionic/src/environments/environment.ts` paramétrer :

- `languages` : Les langues proposées dans l'application
- `defaultLanguage` : la langue par défaut

```
export const environment = {
  [...]
  languages: ['fr', 'en'],
  defaultLanguage: 'fr',
  [...]
};
```

## Titre de l'application et numéro de version

Dans le fichier `/dev/user-frontend-ionic/src/environments/environment.ts` paramétrer :

- `appTitle` : Titre de l'application
- `appVersion` : version de l'app

```
export const environment = {
  [...]
  appName: 'Titre de l\'application',
  appVersion: '1.0.1'
  [...]
};
```



Le numéro de version tel qu'il est spécifié dans le fichier `environment.ts` ne sera utilisé que dans la version PWA du client. Pour les clients natifs, la version sera celle spécifiée lors du build Android ou iOS et sera récupérée et affichée via un plugin Capacitor.

## Firestore pour les notifications

Pour activer **firebase** vous trouverez dans le répertoire `/dev/user-frontend-ionic/src/environments` le sous-répertoire **firebase** et l'arborescence suivante :

```
firebase
|--android
|--ios
|--web
```

Y placer les fichiers téléchargés depuis firebase pour iOS et Android

Pour les notifications web, dans `/dev/user-frontend-ionic/src/environments/firebase/web` dupliquer et renommer le fichier **firebase-environment.pwa.json.dist** en **firebase-environment.pwa.json** et compléter avec les informations fournies par firebase.

Une fois complété, il devra ressembler à cela :

```
{
  "apiKey": "",
  "authDomain": "universite-lorraine.firebaseio.com",
  "projectId": "universite-lorraine",
  "storageBucket": "universite-lorraine.appspot.com",
  "messagingSenderId": "123456789",
  "appId": "1:123456789azerty987654321qsdfg",
  "measurementId": "G-A1Z2E3R4T5",
  "vapidKey": "xxxx la cle fournie xxxx"
}
```

Décommenter les deux lignes de `/dev/user-frontend-ionic/src/environments/environment.ts`

```
import firebasePwaEnvironment from './firebase/web/firebase-environment.pwa-development.json';

export const environment = {
  [...]
  firebase: firebasePwaEnvironment,
  [...]
};
```

Ionic propose des options supplémentaires grâce à un plugin `FirebaseMessaging` déclaré dans le fichier `/dev/user-frontend-ionic/capacitor.ts`

```
plugins: {
  FirebaseMessaging: {
    presentationOptions: ['badge', 'sound', 'alert'],
  },
},
```

Enfin, ces fichiers doivent figurer dans les application clientes. L'outil **Trapèze** permet de copier ces fichiers aux bons endroit.

Dans le fichier `/dev/user-frontend-ionic/src/trapeze-config.yml` décommenter

```
platforms:
  android:
  [...]
  copy:
  - src: ../src/environments/firebase/android/google-services-dev.json
    dest: app/google-services.json

  ios:
  [...]
  copy:
  - src: ../../src/environments/firebase/ios/GoogleService-Info-dev.plist
    dest: App/GoogleService-Info.plist
```