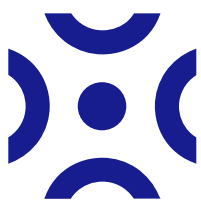


# CONFROID

APPLICATION ANDROID ET BIBLIOTHEQUE DE GESTION CENTRALISEE DE CONFIGURATIONS

MANUEL D'UTILISATION



**Université  
Gustave Eiffel**

## PREREQUIS

*Note : Un fichier Readme est disponible à la racine du projet, comprenant un « Getting started » expliquant comment mettre en place rapidement les applications.*

Récupérez une copie du projet sur le dépôt GitHub. La racine contient 3 répertoires : l'application Confroid, l'application web service et une application de démo.

## MISE EN PLACE DU SERVICE WEB

Le service web est fournie et utilise Spring et Hibernate pour stocker les données dans une base de données. Par défaut la base de données est PostgreSQL et les identifiants de connexion sont **postgres** et **postgres**.

Il est possible de modifier ces informations dans le fichier *application.properties* situé dans « *Confroid storage service/src/main/resources/application.properties* ».

- Si vous souhaitez changer de système de gestion de base de données vous devez renseigner un driver pour les champs **spring.datasource.driver-class-name** et **spring.jpa.properties.hibernate.dialect**
- Le champ **spring.datasource.url** contient l'URL de connexion à la base de données
- Les champs **spring.datasource.username** et **spring.datasource.password** correspondent aux identifiants de connexion à la base de données
- Le champ **server.port** correspond au port auquel le serveur sera lancé, par défaut **8080**

Lorsque ces informations sont renseignées, il suffit d'exécuter le programme qui tournera sur votre adresse IP et sur le port renseigné.

Un dump des données est disponible dans le répertoire Documentation du projet avec comme un user dont le login est « **Toto** » et le mot de passe « **demo** ».

Il contient également quelques configurations que vous pourrez afficher dans l'application Confroid.

## APPLICATION CONFROID

Le répertoire Confroid contient les sources de l'application. Le plus simple pour le compiler et le lancer est de l'exécuter avec Android Studio. La version minimum du SDK est **24**.

## CONNEXION AU SERVICE WEB

Pour connecter l'application au service web, dirigez vous vers la page de paramètres (icone réglages en haut à droite du menu principal) puis dans « Configure access / Configurer l'accès » de la partie « Cloud Service ». Sur cette page vous pourrez renseigner le lien du serveur précédemment lancé (sans slash à la fin) ainsi que les identifiants de connexion au service (pas ceux de la base de données).

Par exemple :

**Base address** : `http://12.123.56.2:8080`

**Login** : Toto

**Password** : demo

Cliquez ensuite sur « Test connexion » pour tester la connexion et vous connecter en cas de succès.

*Note : La connexion au service web est optionnelle et n'empêche pas d'utiliser l'application si la connexion n'est pas établie.*

## UTILISATION DE L'APPLICATION

L'interface graphique est simple et intuitive, néanmoins il faut savoir que lors d'un appui long sur une configuration (une version d'un nom) a pour effet d'afficher un menu contextuel permettant d'effectuer diverses actions.

Lors de l'exploration d'une configuration il est également possible de faire un appui long sur une entrée pour la supprimer et/ou renommer si la valeur correspond à la clé d'un dictionnaire.

Un bouton « + » en haut à droite permet d'ajouter une entrée si la valeur correspond à un dictionnaire ou une liste. La création consiste à dupliquer le dernier élément, qui est ensuite modifiable.

L'onglet Local affiche les configurations de la base de données **SQLite** de l'application, l'onglet Cloud affiche les configurations du service web et l'onglet Files permet d'importer un fichier (avec un bouton prévu à cet effet) et d'explorer les configurations.

Les fonctionnalités précédemment énoncées sont les mêmes quelque soit la source de données.

## APPLICATION DE DEMO

Une application de démo est également disponible dans le répertoire ConfroidDemo de la racine du projet. Elle inclut la bibliothèque Confroid et possède une activité dont le layout contient des boutons permettant d'exploiter les fonctionnalités de la bibliothèque.

Elle possède notamment une classe démo contenant des données de tests. Il est tout à fait possible de créer ses propres données afin de les envoyer aux méthodes fournies par la bibliothèque pour les tester.