

Manuel d'installation

SIForms : Application de saisie de formulaires

Table des matières

1. Prérequis	1
2. Liste des bibliothèques à installer	1
2.1. Backend (Django)	1
2.2. Frontend (React Native avec Expo)	2
3. Configuration de PostgreSQL	2
3.1. Démarrer le service PostgreSQL	2
3.2. Accéder au terminal PostgreSQL	2
4. Configuration du Projet Django	2
4.1. Créer un fichier .env	2
5. Commandes pour initialiser le projet	3
5.1. Créer et appliquer les migrations	3
5.2. Créer un super utilisateur (admin)	3
5.3. Lancer le serveur de développement	3
6. Lancer le projet React Native	3
7. Générer un fichier APK avec Expo	3

1. Prérequis

Avant de commencer, assurez-vous d'avoir les éléments suivants installés sur votre système :

```
Python 3.x
PostgreSQL
pip
Node.js
Expo CLI
Git
```

2. Liste des bibliothèques à installer

2.1. Backend (Django)

Bibliothèque	Description
Django	Framework web pour Python.
djangorestframework	Framework pour créer des API REST avec Django.
psycopg2	Adaptateur PostgreSQL pour Python.
python-dotenv	Gestion des variables d'environnement via un fichier .env.
qrcode[pil]	Génération de codes QR avec support d'images via PIL (Pillow).

Commande d'installation :

```
pip3 install --break-system-packages Django djangorestframework psycopg2 python-dotenv qrcode[pil]
```

2.2. Frontend (React Native avec Expo)

Bibliothèque	Description
expo	Framework pour développer des applications React Native.
react-native	Framework pour développer des applications mobiles natives avec React.
@react-native-async-storage/async-storage	Bibliothèque pour le stockage local sur React Native.

Commandes d'installation :

```
npm install -g expo-cli  
cd emplacement/du/projet  
npm install @react-native-async-storage/async-storage  
npm install
```

3. Configuration de PostgreSQL

3.1. Démarrer le service PostgreSQL

Démarrez le service PostgreSQL sur votre système :

```
sudo service postgresql start
```

3.2. Accéder au terminal PostgreSQL

Accédez au terminal PostgreSQL en tant qu'utilisateur postgres :

```
sudo -i -u postgres  
psql
```

4. Configuration du Projet Django

4.1. Créer un fichier .env

Créez un fichier .env à la racine de votre projet Django et ajoutez-y les variables d'environnement suivantes :

```
SECRET_KEY=django-insecure-l2!h2**v04*+so*f*w^#=v7n4ydhsxrw8bb1$tgr4-nh4=6j3  
DEBUG=True  
DB_NAME=nom_de_la_base_de_données  
DB_USER=nom_utilisateur  
DB_PASSWORD=mot_de_passe  
DB_HOST=localhost  
DB_PORT=5432
```

5. Commandes pour initialiser le projet

5.1. Créer et appliquer les migrations

Générez les fichiers de migration et appliquez-les à la base de données :

```
python3 manage.py makemigrations
python3 manage.py migrate
```

5.2. Créer un super utilisateur (admin)

Créez un superutilisateur pour accéder à l'interface d'administration de Django :

```
python3 manage.py createsuperuser
```

Suivez les instructions pour définir un nom d'utilisateur, un email et un mot de passe.

5.3. Lancer le serveur de développement

Démarrez le serveur de développement Django :

```
python3 manage.py runserver
```

6. Lancer le projet React Native

Pour démarrer votre projet React Native avec Expo, exécutez :

```
expo start
```

Scannez le code QR avec l'application Expo Go sur votre téléphone ou utilisez un émulateur pour tester l'application.

7. Générer un fichier APK avec Expo

Se connecter à Expo :

```
expo login
expo whoami
```

Créer et configurer un fichier eas.json contenant :

```
{
  "build": {
    "preview": {
      "android": {
        "buildType": "apk"
      }
    },
    "preview2": {
      "android": {
        "gradleCommand": ":app:assembleRelease"
      }
    },
    "preview3": {
      "developmentClient": true
    }
  }
}
```

```
    },  
    "production": {}  
  }  
}
```

Ajouter des variables d'environnement (si nécessaire) :

```
eas secret:create
```

Générer le fichier APK :

```
eas build -p android --profile preview
```