MOBILE APP ENVIRONMENT SETUP

Introduction

Welcome to the Redback mobile app development team environment setup document. This guide provides a comprehensive overview of the setup process required for working on the Redback mobile app. It is designed to assist new and existing team members in setting up their development environments efficiently and effectively.

This document covers the full setup process for both the front-end and back-end of the Redback mobile app. It includes detailed instructions on installing and configuring essential development tools such as Android Studio, Flutter, and Visual Studio Code (VS Code). Additionally, it provides step-by-step instructions on cloning the app and back-end repositories from GitHub, setting up virtual environments, installing dependencies, and running the applications.

The front-end setup section will guide you through configuring Android Studio, cloning the front-end repository, and running the application on an emulator or physical device. It also explains how to create a `.env` file for app configurations and install necessary project dependencies.

The back-end setup section outlines the process of cloning the back-end repository, setting up a Python virtual environment, installing the required dependencies, running database migrations, and starting the back-end server.

In addition to setup instructions, this document provides guidance on using version control to commit and push changes to the repositories. It also addresses potential issues you may encounter during the setup process and offers solutions to resolve them.

By following this environment setup guide, you can ensure a smooth and consistent development experience across the team. This document serves as a reference for all Redback mobile app development team members as they work on the project.

Development Tools - Front End Setup

Android Studio

Installation:

- Visit the Android Studio download page.
- Choose the appropriate download link for your operating system (Windows, macOS, or Linux).
- Once downloaded, follow the installation instructions provided on the website for your operating system.

Configuration:

- After installation, launch Android Studio.
- Follow the setup wizard to install necessary SDKs and tools.

Flutter SDK

Installation:

- Visit the Flutter installation page.
- Choose the appropriate download link for your operating system (Windows, macOS, or Linux).
- Once downloaded, follow the installation instructions provided on the website for your operating system.

Configuration:

- After extracting the SDK, add the **flutter** binary to your PATH.
- Verify the installation by running **flutter --version** in a terminal or command prompt.

Cloning the Front-End App from GitHub

Repository URL: Front-End App Repository.
Cloning Instructions:

- Open Android Studio.
- Go to **File > New > Project from Version Control.**
- In the "URL" field, paste the repository URL: **https://github.com/alexbaar/app.git**.
- Choose a location to save the project on your computer and click **Clone** to download the project.

Setting Up the Environment - Front End

Add a **.env** File:

* Create a .env file in the root project directory.
* Inside the .env file, add the following configuration:

  API_URL_BASE = http://<your_machine_network_address>:8000

* Replace **<your_machine_network_address>** with your machine's network address (e.g., **http://192.168.3.103:8000)**.

Install Dependencies:

* Open **pubspec.yaml** in the root project directory.
* Click the "**Pub Get**" button to install project dependencies.

Development Tools - Back End Setup

Python 3

Python 3 Installation:

* Go to the Start menu and type "**Microsoft Store**" to open it, or you can find it in the list of installed apps.
* In the Microsoft Store, use the search bar to search for "**Python 3**".
* In the search results, locate and select the Python 3 app (e.g., "**Python 3.9**" or "**Python 3.10**" depending on the available version).
* Click the "**Get**" or "**Install**" button to start the installation.
* Once the installation is complete, you can find Python 3 in the list of installed apps.

Verify Python Installation:

* Open a terminal (e.g., Command Prompt or PowerShell).
* Type the following command to verify the installation and check the version:

  python --version

* If Python 3 is installed successfully, you should see the version number.

Visual Studio Code

Download VS Code:

- Go to the Visual Studio Code download page.
- Click the "**Download**" button to download the VS Code installer for Windows.

Run the Installer:

- Locate the downloaded installer file and double-click it to start the installation process.
- Follow Installation Wizard:
- Follow the installation wizard to install VS Code.
- You can customize the installation options according to your preferences.

Complete Installation:

- Once the installation is complete, you can launch Visual Studio Code from the Start menu or desktop shortcut.

Install Python Extensions:

- In the VS Code Extensions marketplace (**View > Extensions or Ctrl + Shift + X**), search for "**Python**".
- Install the Python extension by Microsoft for additional Python support, such as IntelliSense, debugging, and more.

Setting Up the Environment - Back End

Repository URL: Back-End App Repository.
Cloning Instructions:

- Open VS Code (with no project open).
- Click on the "**Source Control**" icon in the vertical panel on the left.
- Click on 'Clone Repository' and paste the repository URL (**https://github.com/alexbaar/app_backend.git**).
- The project will open in VS Code.

Navigate to Backend Server Directory:

- In VS Code, open a terminal (**View > Terminal**).

Navigate to the backend server directory: **cd backend_server**

- Run ls to list files in the directory; you should see files like **__init__.py, admin.py, pycache,** and others.

Create a virtual environment with the following command:

python3 -m venv .venv

Activate the virtual environment:

- On Windows: **.venv\Scripts\activate**
- On macOS/Linux: **source .venv/bin/activate**

Installing Dependencies:

- After activating the virtual environment, install Django and Django REST framework:

  pip install Django djangorestframework

Running Migrations:

- Navigate one level up to the project root directory: **cd .**.

Run migrations to create and update the database schema:

- Run the following command to make migrations: **python manage.py makemigrations**
- Run the following command to apply migrations: **python manage.py migrate**

Running the Back-End Server

- Start the server using the following command: **python manage.py runserver 0.0.0.0:8000**

Running the Front-End App

- Connect a physical Android device via USB or open an emulator in Android Studio.
- In Android Studio, navigate to the project directory and select the run configuration for the app.
- Click the "Run" button to start the app.

Version Control

Android Studio

Commit and Push Changes:

- In Android Studio, go to **Git** > **Commit** and select the files you want to commit.
- Add a commit message and commit your changes.
- Then, go to **Git** > **Push** to push your changes to the repository.

Visual Studio Code

Commit and Push Changes:

- In Visual Studio Code, go to the Source Control panel (click the third icon on the left sidebar).
- Stage the files you want to commit by clicking the + icon next to each file in the Changes section, or stage all changes by clicking the + icon next to "**Changes**."
- In the commit input field, type a commit message describing your changes.
- Click the "**Commit**" button to commit the staged changes with the provided message.
- After committing, you can push your changes to the remote repository.
- In the Source Control panel, click the three dots (More Actions) and select "**Push**".
- This will push your committed changes to the remote Git repository.

Troubleshooting

Common Issues:

Error in Primary Color:

- If you encounter an error related to the primary color in your Android app:
  - Look for the error in Android Studio.
  - Click on the **red bulb** icon next to the error.
  - Select **Show Configuration** to change the primary color to a different color.
  - Apply the changes and check if the error is resolved.

Pillow Installation Error:

- Possible Error: `backend_server.acc_details.image: (fields.E210) Cannot use ImageField because Pillow is not installed.

- Solution:
  - You need to install the Pillow library, which is used for image handling in Django.

- In the terminal of Visual Studio Code, run one of the following commands depending on your Python version: **python -m pip install Pillow** or **python3 -m pip install Pillow**
- Once Pillow is installed, rerun the code that caused the error.

Helpful Resources:

- Android Studio Documentation: Refer to the [official Android Studio documentation](https://developer.android.com/studio) for guides, troubleshooting, and tips.
- Flutter and Dart Documentation: If you are working with Flutter and Dart, check out the [official Flutter documentation](https://flutter.dev/docs).
- Python Documentation: For Python-related issues, refer to the [official Python documentation](https://docs.python.org/3/).
- VS Code Documentation: For issues with Visual Studio Code, refer to the [official VS Code documentation](https://code.visualstudio.com/docs).
- Django Documentation: If you are using Django, refer to the [official Django documentation](https://docs.djangoproject.com/).
- GitHub Forums: For any Git-related issues, check out the [GitHub community forums](https://github.community/).

By referring to these resources and following the provided solutions, you should be able to resolve common issues that may arise during setup and development.

This guide should help you set up both the front-end and back-end of the app, including cloning the repositories from GitHub, setting up a virtual environment, installing dependencies, and running the applications. It also includes version control instructions for saving changes and pushing them to the repository.