Welcome to AdEstate Documentation

AdEstate is a complete real estate management platform with mobile apps (Android/iOS), web interface, and powerful admin tools. Featuring comprehensive property listings, agent management systems, appointment booking, floor plan visualization, and comparison tools, AdEstate empowers real estate businesses to streamline operations. With robust features like subscription plans, multi-language support, KYC verification, and responsive customer care, our platform helps you manage properties efficiently while delivering exceptional client experiences. AdEstate combines cutting-edge technology with user-friendly design to transform property management, helping you convert opportunities into successful transactions. Our solution offers enterprise-grade capabilities at competitive pricing, enabling you to stand out in the competitive real estate market.

Server Setup Requirements >

Server Requirements

Before installing the software, please ensure that your server meets the following requirements:

- **PHP Version:** Ensure that your server has PHP 8.1.25 or a later version installed.
- **Laravel Framework:** The software requires Laravel 9.1. Make sure your server is compatible with this version.
- **Database:** MySQL 5.1 or later version is required. Ensure MySQL is installed and accessible.
- **Terminal Access:** You need terminal access to execute commands during the installation process.
- **Composer:** Composer is required for managing PHP dependencies. Make sure it's installed on your server.
- **PHP Extensions:**
  - openssl
  - pdo
  - mbstring
  - tokenizer
  - json
  - gd
  - curl
  - zip
  - zlib
  - fileinfo
  - exif
  - bcmath
- **Apache Configuration:** Ensure mod_rewrite is enabled in Apache for URL rewriting.

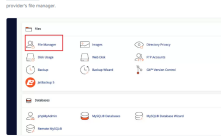
‹ Welcome

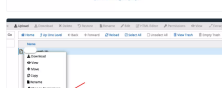
Server Configuration ›

Server Setup and Installation Guide

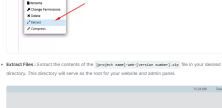
Follow these steps to set up and install the website and admin panel on your server:

- **Step 1**: Upload Files
  - **Download and Upload**: After downloading the code from Codecanyon, upload the `[project_name]-web-[version_number].zip` file to your server using your preferred method, such as FTP or through your hosting provider's file manager.

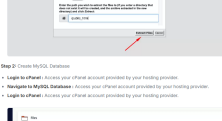


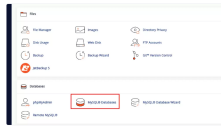


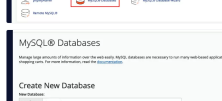
  - **Extract Files**: Extract the contents of the `[project_name]-web-[version_number].zip` file in your desired directory. This directory will serve as the root for your website and admin panel.

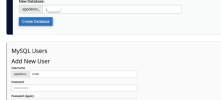


- **Step 2**: Create MySQL Database
  - **Login to cPanel**: Access your cPanel account provided by your hosting provider.
  - **Navigate to MySQL Database**: Access your cPanel account provided by your hosting provider.
  - **Login to cPanel**: Access your cPanel account provided by your hosting provider.

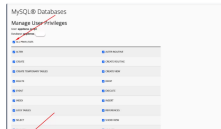








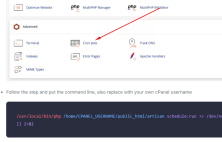
- **Step 3**: Setup Cron Jobs
  - Click on Cron Jobs



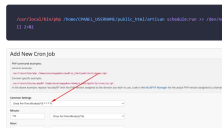
  - Follow the step and put the command line, also replace with your own cPanel username

```
/usr/local/bin/php /home/CPANEL_USERNAME/public_html/artisan schedule:run >> /dev/null 2>&1
```

- **Step 4**: Installation
  - **Run Installation Script**: Run your domain URL in a web browser. You'll be prompted with the installation steps. Follow the instructions carefully.
  - **Start Installation**: Click the "Start" button to initiate the installation process.



  - **Follow Instructions**: Follow each step of the installation wizard, providing necessary information when prompted, such as database connection details.
  - **Complete Installation**: Once all steps are completed successfully, you'll receive a confirmation message indicating that the installation is complete.

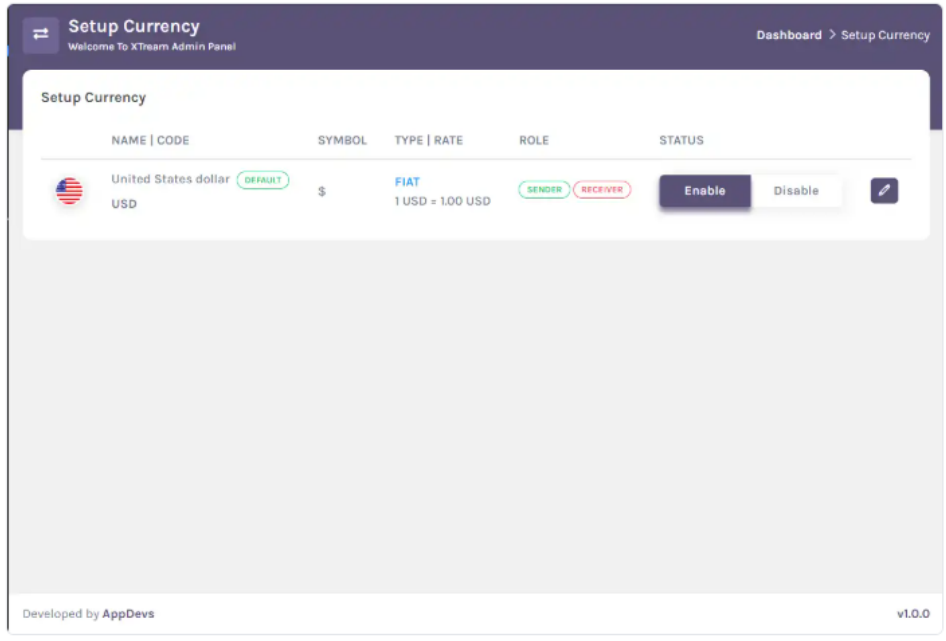


Conclusion: Congratulations! You have successfully set up and installed the website and admin panel on your server. You can now access and manage your website through your domain. If you encounter any issues during the installation process, refer to the documentation provided or seek assistance from your hosting provider. Enjoy using your new website and admin panel!

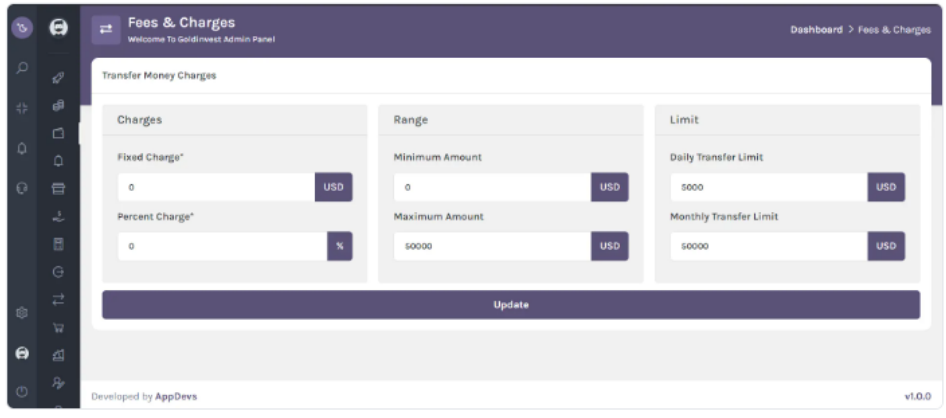
Admin Panel Setup and Configuration Guide

Follow these steps to configure the admin panel and make necessary changes to run the site:

- **Default Settings:**
  - **Setup Currency**
    - Define the default currency for transactions and display throughout the site. Read Article >



  - **Fees and Charges**
    - Specify fees and charges applicable to transactions. Read Article >



  - **Events**
    - Promote your event, manage attendee registrations, and provide updates.

Conclusion: Once you have completed the above steps, your admin panel will be configured with the necessary settings to run the site smoothly. Make sure to review and update these settings regularly to ensure optimal performance and security. If you encounter any issues during the setup process, refer to the documentation provided or seek assistance from your system administrator. Enjoy managing your site with the configured admin panel!

< Server Configuration

Admin Panel Setup (Settings) >

Setup and Configuration Admin Panel Setting

Follow these steps to configure the admin panel and make necessary changes to run the site:

- **Web Settings:**
  - **Basic Settings:**
    - Set up business and basic information such as company name, address, and contact details. Read Article >
  - **Image Assets:**
    - Upload and manage image assets such as logos, favicon.
  - **Setup SEO:**
    - Optimize the site for search engines by configuring SEO settings such as meta tags, keywords, and descriptions.
- **App Settings:**
  - **Splash Screen:**
    - Set up or manage your app splash screen and app version. Read Article >
  - **Onboard Screen:**
    - Upload and manage onboard screen for your app. Read Article >
  - **App URLs:**
    - Set your website app download url from here. Read Article >
- **Language Settings:**
  - **Language Manager:**
    - Set up and manage your mother language for your website and app. also you can edit, enable or disable your added language. Read Article >

Conclusion: Once you have completed the above steps, your admin panel will be configured with the necessary settings to run the site smoothly. Make sure to review and update these settings regularly to ensure optimal performance and security. If you encounter any issues during the setup process, refer to the documentation provided or seek assistance from your system administrator. Enjoy managing your site with the configured admin panel!

< Admin Panel Setup (Default)

Admin Panel Setup (Verification) >

Setup and Configuration Verification Center

Follow these steps to configure the admin panel and make necessary changes to run the site:

- **Setup Email:**
  - Configure the email method for verification purposes. Use SMTP webmail for email configuration.
    Read Article >
- **Setup KYC:**
  - Enable and configure Know Your Customer (KYC) verification for user authentication and security.
    Read Article >

Conclusion: Once you have completed the above steps, your admin panel will be configured with the necessary settings to run the site smoothly. Make sure to review and update these settings regularly to ensure optimal performance and security. If you encounter any issues during the setup process, refer to the documentation provided or seek assistance from your system administrator. Enjoy managing your site with the configured admin panel!

< Admin Panel Setup (Settings)

Admin Panel Setup (Web Content) >

Setup Web Content

Follow these steps to configure the admin panel and make necessary changes to run the site:

- **Setup Section**
  - Fill the inputs you needed along with fileholder and then submit all. You can also create, update and delete data.
- **Setup Pages**
  - You can Enable/Disable your page.

Conclusion: Once you have completed the above steps, your admin panel will be configured with the necessary settings to run the site smoothly. Make sure to review and update these settings regularly to ensure optimal performance and security. If you encounter any issues during the setup process, refer to the documentation provided or seek assistance from your system administrator. Enjoy managing your site with the configured admin panel!

‹ Admin Panel Setup (Verification)

Admin Panel Setup (Payment Methods) ›

Setup Payment Methods

Follow these steps to configure the admin panel and make necessary changes to run the site:

- **Automatic:**
  - Input your payment credentials for automatic payment processing.
- **Manual:**
  - Add manual payment methods for transactions, allowing admin to handle transactions manually.

Conclusion: Once you have completed the above steps, your admin panel will be configured with the necessary settings to run the site smoothly. Make sure to review and update these settings regularly to ensure optimal performance and security. If you encounter any issues during the setup process, refer to the documentation provided or seek assistance from your system administrator. Enjoy managing your site with the configured admin panel!

‹ Admin Panel Setup (Web Content)

Admin Panel Setup (Notification) ›

Setup and Configuration Notification

Follow these steps to configure the admin panel and make necessary changes to run the site:

- **Push Notification:**
  - Integrate the Pusher for push notifications and real-time chat with users. Read Article >

Conclusion: Once you have completed the above steps, your admin panel will be configured with the necessary settings to run the site smoothly. Make sure to review and update these settings regularly to ensure optimal performance and security. If you encounter any issues during the setup process, refer to the documentation provided or seek assistance from your system administrator. Enjoy managing your site with the configured admin panel!

< Admin Panel Setup (Payment Methods)

Flutter Environment Setup >

App Environment Setup Guide

To customize and work on this project, ensure that you have the necessary tools and configurations in place. Follow these steps to set up your development environment:

- **Prerequisites**
  - **Android Studio/VS Code:** Install Flutter and Dart Plugins in your preferred Integrated Development Environment (IDE), such as Android Studio, Visual Studio Code, or IntelliJ.
  - **Flutter SDK:** Install the latest stable version of Flutter on your system.
  - **Dart SDK:** Ensure the latest stable version of Dart is installed alongside Flutter.
  - **Java (JDK):** Install java JDK version 11 (LTS) according to your operating system.

```
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.24.0, on macOS 14.5 23F79 darwin-arm64, locale
    en-BD)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Xcode - develop for iOS and macOS (Xcode 15.4)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2023.2)
[✓] VS Code (version 1.92.2)
[✓] Connected device (4 available)
[✓] Network resources

• No issues found!
```

- **Project Setup**
  - **Download and Extract Files :** Unzip the project files folder. Inside, you'll find the source code named as `[project name]-app-[version number]`.
  - **Open Project in IDE :** Use your preferred IDE (Android Studio, Visual Code, IntelliJ) to open the project.
  - Please run this command:

```
flutter pub get
```

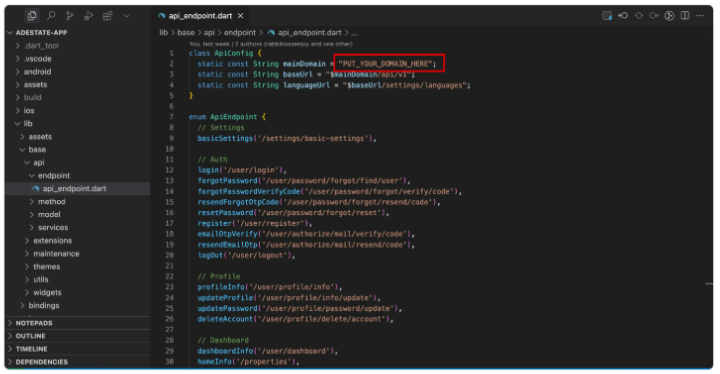
‹ Admin Panel Setup (Notification)

App Installation ›

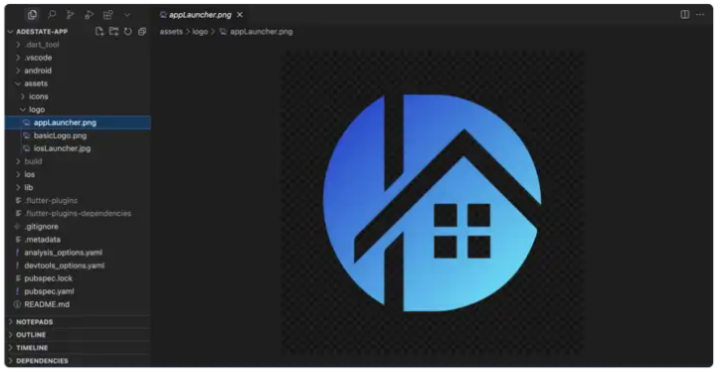
User App Configuration Guide

Follow these steps to set up the app with your information inside the project files:

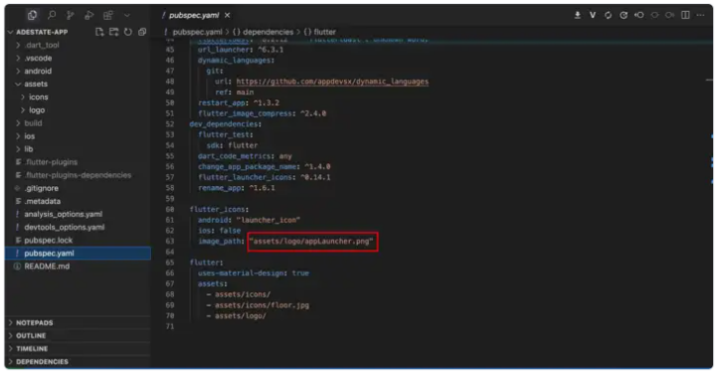
- **Set Domain**
  - **Domain Configuration:** Navigate to `lib/base/api/endpoint/api_endpoint.dart` file and replace the domain with your own. Ensure to update all API endpoints with your domain.
    Replace "**PUT_YOUR_DOMAIN_HERE**" with "**https://adestate.appdevs.net**" as the demo testing URL in the mainDomain field. The app won't work without a valid project website link. To run the application smoothly, make sure to deploy the web application first.



- **Logo/Icon Change**
  - **Logo/Icon Replacement:** Go to the `/assets/logo/` directory and replace all image files with your custom logos/icons. Ensure to keep the file names unchanged.



- **Package Name and App Launcher Name-Icon Change**
  - **Package Name and App Launcher Configuration:** Open the `/pubspec.yaml` file and update the **flutter_icons** section with the path to your launcher icon. Also, update the **name** field with your desired app name.



  - **Run Commands:** Execute the following commands in your terminal:

```
flutter pub get
flutter pub run flutter_app_name
dart run rename_app:main all="Your App Name"
flutter pub run change_app_package_name:main com.new.package.name
```

Note: replace `com.new.package.name` with your desired package name and your app name with the title of your application.

- **Additional Note**
  - **Device Installation:** It's recommended to use a real device for installing the app, as emulators/simulators may not always function correctly.

Conclusion: By following these steps and executing the provided commands, you can configure the app with your own domain, logos/icons, package name, and app launcher name. Ensure to test the app thoroughly after making these changes to ensure everything is functioning as expected. If you encounter any issues during the configuration process, refer to the documentation or seek assistance from the Flutter community. Happy app customization!

‹ Flutter Environment Setup

Development & Testing ›

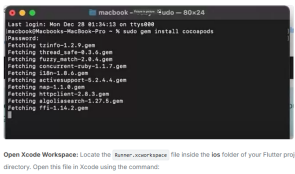
Build & Run Guide

- **Run the Android Application:**
  - **Select Android Device:** In Android Studio, select an Android device from the target selector for running the app. If no device is available, create one using `Tools > Android > AVD Manager`.
  - **Run the App:** Click the run icon in the toolbar or select `Run > Run from the menu`. This will initiate the app's deployment and execution on the selected Android device.
  - If you prefer the command line:

```
flutter run
```

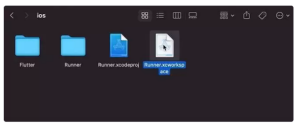
- **Run the iOS Application:**
  - **Prepare Flutter App on Mac:** Ensure you have a test Flutter app set up on your Mac.
  - **Device Trust:** On your iPhone, navigate to `Settings > General > Device Management` or Profile, and trust "Your Developer Name" to enable testing.
  - **Xcode Setup:** Install CocoaPods using the following command in the terminal:

```
$ sudo gem install cocoapods
```

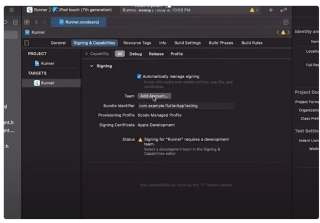


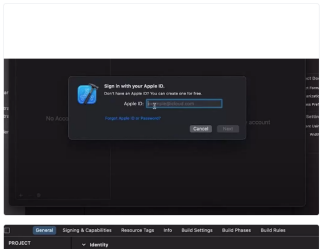
  - **Open Xcode Workspace:** Locate the `Runner.xcworkspace` file inside the **ios** folder of your Flutter project directory. Open this file in Xcode using the command:

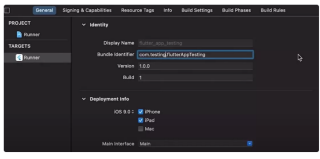
```
$ open ios/Runner.xcworkspace
```



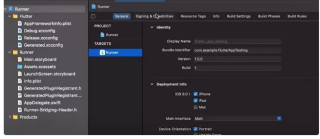
  - **Configure Signing & Capabilities:** In Xcode, navigate to the Runner file in the left menu bar and go to the Signing & Capabilities page. Click "Add an Account" to add your Apple ID and password. Then, change the Bundle Identifier to a unique value.

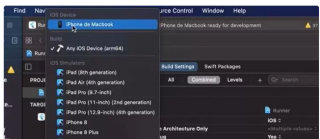




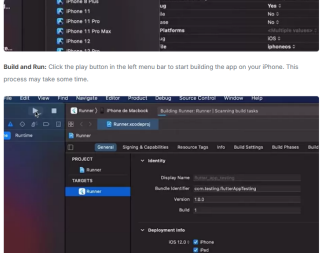


  - **Connect iPhone:** Connect your iPhone to your Mac. In Xcode, select your iPhone device from the device list.

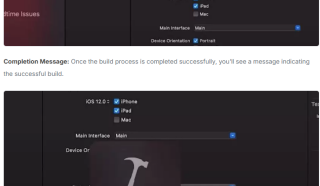




  - **Build and Run:** Click the play button in the left menu bar to start building the app on your iPhone. This process may take some time.



  - **Completion Message:** Once the build process is completed successfully, you'll see a message indicating the successful build.



Conclusion: By following these steps, you can build and run your Flutter app on both Android and iOS devices. Ensure that your development environment is properly configured and that you've trusted your device for testing on iOS. If you encounter any issues during the process, refer to the Flutter documentation or seek assistance from the Flutter community. Happy app development!

Building and Signing APK for Submission to Play Store: A Step-by-Step Guide

Building and submitting your Flutter app to the Google Play Store involves several steps, including building, signing, and uploading the APK. Below is a comprehensive guide on how to accomplish this:

- **Step 1:** Prepare Your Flutter App
  - Ensure that your Flutter app is fully developed and tested before proceeding with the submission process. Make sure all necessary configurations, assets, and dependencies are in place.
- **Step 2: Build the APK**
  - Open your terminal or command prompt.
  - Navigate to your Flutter project directory.
  - Run the following command to build the release version of your APK:

```
flutter build apk --release
```

Note: This command generates an APK file ready for release in the `build/app/outputs/flutter-apk` directory.

- **Step 3: Generate a Signing Key**

Before you can upload your app to the Play Store, you need to sign it with a signing key. Follow these steps to generate a signing key:

  - Open your terminal or command prompt.
  - Navigate to your Flutter project's **android** directory.
  - Run the following command to generate a signing key:

```
keytool -genkey -v -keystore key.jks -keyalg RSA -keysize 2048 -validity 10000 -alia
s key
```

Note: Follow the prompts to enter the required information, such as passwords and aliases.

- **Step 4: Configure Gradle Build Script**

To sign your app using the generated signing key, you need to configure the Gradle build script. Follow these steps:

  - Open the `android/app/build.gradle` file in your Flutter project.
  - Navigate to your Flutter project's **android** directory.
  - Add the following code snippet at the end of the file:

```
groovy:

android {
 signingConfigs {
  release {
    keyAlias 'your_alias'
    keyPassword 'your_key_password'
    storeFile file('path_to_your_keystore_file')
    storePassword 'your_keystore_password'
  }
 }
 buildTypes {
  release {
    signingConfig signingConfigs.release
  }
 }
}
```

Note: Replace `'your_alias'`, `'your_key_password'`, `'path_to_your_keystore_file'`, and `'your_keystore_passwor d'` with the appropriate values.

- **Step 5: Build Signed APK**

To sign your app using the generated signing key, you need to configure the Gradle build script. Follow these steps:

  - Open your terminal or command prompt.
  - Navigate to your Flutter project directory.
  - Run the following command to build the signed APK:

```
flutter build apk --release
```

Note: This command generates a signed APK file ready for submission to the Play Store.

- **Step 6: Create a Google Play Developer Account**

If you don't already have one, create a Google Play Developer account. You'll need to pay a one-time registration fee to access the Google Play Console.

- **Step 7: Log in to Google Play Console**
  - Visit the Google Play Console website.
  - Log in using your Google account credentials.
- **Step 8: Create a New App Release**
  - In the Google Play Console dashboard, select your app.
  - Navigate to the **"Release"** tab on the left sidebar.
  - Click on **"Create Release"** or **"Manage Production"** depending on your app's current release status.
- **Step 9: Upload Your APK**
  - Click on **"Upload"** and select the signed APK file generated earlier.
  - Wait for the upload to complete. Google Play Console will validate the APK for any errors or policy violations.
- **Step 10: Configure Release Details**
  - Fill in the release details, including the release name, release notes, and any other relevant information.
  - Verify that all information is accurate and up-to-date.
- **Step 11: Rollout Options (Optional)**

You can choose how you want to roll out your release:

  - **Gradual rollout:** Release the update to a percentage of your users.
  - **Immediate rollout:** Release the update to all users immediately.
- **Step 12: Review and Publish**
  - Review all the details and configurations carefully.
  - Click on **"Review"** or **"Publish"** to submit your release for review.
- **Step 13: Review Process**
  - Your release will undergo review by Google Play team, which may take a few hours to several days depending on their workload.
- **Step 14: Distribution**
  - Once your release passes review, it will be available to users on the Google Play Store.
- **Step 15: Monitor Performance**
  - Monitor your app's performance using the metrics provided in the Google Play Console. This includes installation metrics, user feedback, and app ratings.

Conclusion: By following these steps, you can successfully upload your APK to the Google Play Store and make your app available to millions of users worldwide. Remember to keep your app updated and respond promptly to user feedback to ensure a positive user experience.

Installing and Releasing iOS App on App Store: Step-by-Step Guide

Before building and releasing your app on the App Store, you need to set up a place for it using App Store Connect and register a unique bundle ID for your app. Here's how you can accomplish this:

- **Step 1:** Set Up Bundle ID on Apple Developer Account
  - Log in to your [App Store Connect account]
  - Open the **"App IDs"** page.
  - Click on the **"+"** button to create a new Bundle ID.
  - Fill in the necessary information, including the App Name and Explicit App ID.
  - If your app requires specific services, select them and click Continue.
  - Review the details and click Register to finish.
  - In the account dashboard, select **"My Apps"**.
  - Click on "+" and select **"New App"**.
  - Fill in your app details, ensuring iOS is selected, then click Create.
  - From the sidebar, select **"App Information"**.
  - In the General Information section, select the Bundle ID that you registered above.
- **Step 2:** Adjust Xcode Project Settings
  - Open `Runner.xcworkspace` located inside your app's iOS folder using Xcode.
  - In the Xcode project navigator, select the Runner project.
  - Go to the General tab.
  - Fill out the information in the Identity section and ensure the Bundle Identifier matches the one registered on App Store Connect.
  - In the Signing section, ensure **"Automatically manage signing"** is checked and select your team.
  - Fill out the rest of the information as needed.
- **Step 3:** Update App's Icon
  - Select `Assets.xcassets` in the Runner folder from Xcode's project navigator.
  - Update your app's icon as required.
- **Step 4:** Build and Release
  - From the command line, run:

    ```
    flutter build ios
    ```

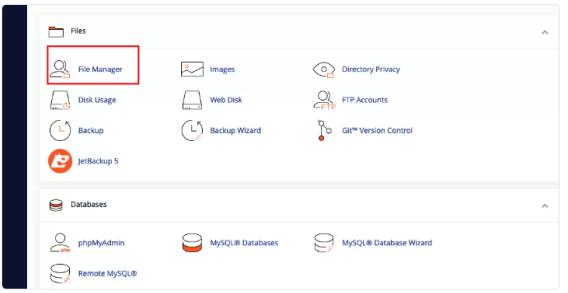
  - Reopen `Runner.xcworkspace` in Xcode.
  - Select Product → Scheme → Runner.
  - Select Product → Destination → Generic iOS Device.
  - Select Product → Archive to produce a build archive.
  - From the Xcode Organizer window, select your iOS app from the sidebar, then select the build archive you just produced.
  - Click the Validate... button to build.
  - Once the archive is successfully validated, click Upload to App Store.
- **Step 5:** Fill in App Details
  - From the command line, run:
  - Select your newly created app from the list.
  - Go to the **"App Information"** section.
  - Fill in all required details such as the app name, subtitle, privacy policy URL, and app icon.
- **Step 6:** Upload App Screenshots
  - Go to the **"App Store"** section.
  - Upload screenshots of your app for different device sizes.
  - Ensure that the screenshots comply with Apple's guidelines regarding content and dimensions.
- **Step 7:** Configure App Metadata
  - Go to the **"App Store"** section.
  - Fill in the app's metadata, including the description, keywords, categories, and ratings.
  - Provide any additional information required by Apple, such as contact information and support URL.
- **Step 8:** Set Pricing and Availability
  - Go to the **"Pricing and Availability"** section.
  - Set the pricing tier for your app.
  - Choose the territories where you want your app to be available.
  - Set the release date and availability options for your app.
- **Step 9:** Submit App for Review
  - Go to the **"App Store"** section.
  - Review all the details of your app to ensure accuracy.
  - Click on **"Submit for Review"** to submit your app for review by Apple's App Review team.
- **Step 10:** Await Review
  - Once you've submitted your app for review, it will undergo a review process by Apple's App Review team. This process may take several days.
- **Step 11:** Review and Respond to Feedback
  - During the review process, Apple may provide feedback or request additional information about your app. Make sure to respond promptly and address any issues raised by the review team.
- **Step 12:** Release Your App
  - Once your app has been approved, you will receive a notification from Apple. You can then proceed to release your app on the App Store by clicking on the **"Release"** button in the App Store section of your app's dashboard.

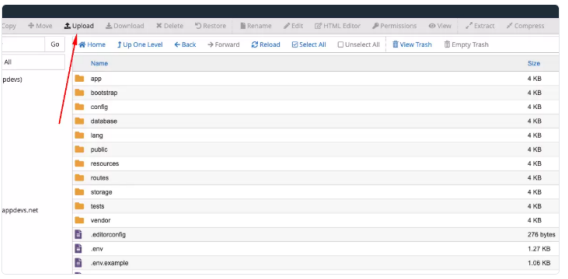
Conclusion: By following these steps, you can successfully submit your app for review and release it on the App Store Console. Make sure to adhere to all App Store guidelines and requirements to ensure a smooth review process. Good luck with your app submission!

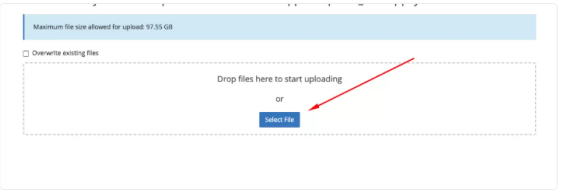
Website Version Update

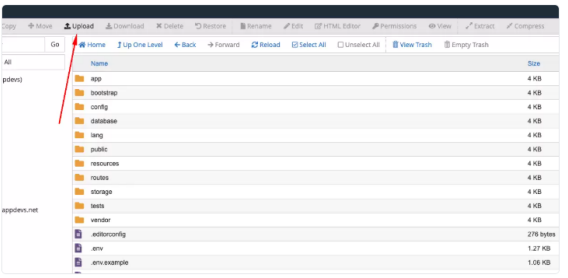
To update your website version for old buyers, please follow these steps:

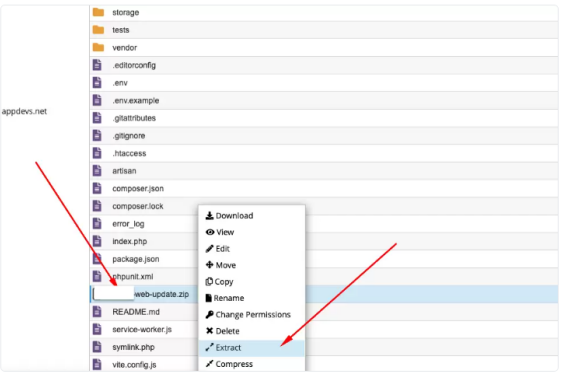
- **Step 1:** Upload and Extract Update Files
  - Re-download the updated code from CodeCanyon.
  - Upload the `[project name]-web-update-file-[version-to-version]` files to your server in the expected directory.



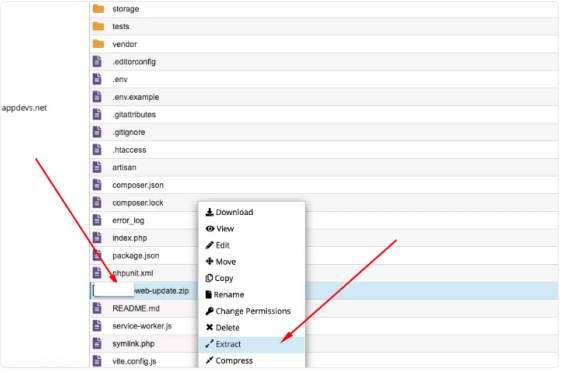








- Extract the zip file inside the `public_html` path.



Note: Make sure to replace `/path/to/your/project` with the actual path to your project directory.

- **Step 2: Execute Update Commands in Terminal**

Conclusion: By following these steps and executing the provided commands, you can successfully update your website version for old buyers. Remember to configure the update files properly before executing the commands to ensure they work as intended. If you encounter any issues, double-check your configurations and seek assistance if needed.

‹ iOS App Deployment

App Version Update ›

Flutter App Version Update

To update your Flutter app version for old buyers, follow these steps:

- **Step 1:** Download and Extract Update Files
  - Re-download the updated code from CodeCanyon.
  - Open the current version file on **Android Studio** or **VS Code**.
  - Extract the `[project name]-app-update-file-[version-to-version]` files in your root project folder or directory.
- **Step 2: Ensure Flutter and Dart Packages are Updated**
  - Ensure that Flutter and Dart packages are updated to the latest version by running the following commands in your terminal:
  - Run Flutter upgrade command

```
flutter upgrade --force
```

  - Run Flutter pub get command

```
flutter pub get
```

Conclusion: By following these steps, you can successfully update your Flutter app version for old buyers. Make sure to extract the update files in your project directory and update Flutter and Dart packages to the latest version to ensure compatibility and functionality. If you encounter any issues during the update process, refer to the documentation or seek assistance from the Flutter community.

‹ Website Version Update