**Getting Started with Flutter Frontend Setup**

1. **Ensure Flutter Version Compatibility**\ Confirm that you are working with **Flutter version 3.8.0**.
2. **Clone the Frontend Repository**\ Clone the Flutter frontend project from the following Bitbucket repository:\ <https://bitbucket.org/enphaseembedded/enl-notification-core-service/src/frontend/>
3. **Add Required Packages and Dev Dependencies**\ Update your pubspec.yaml file to include the following packages and dev dependencies:
   * build\_runner: ^2.5.4
   * flutter\_gen\_runner: ^5.10.0
   * retrofit\_generator: ^9.1.9
   * json\_serializable: ^6.9.4
   * flutter\_launcher\_icons: ^0.14.4
4. **Save and Fetch Dependencies**\ After updating pubspec.yaml, save the file and run flutter pub get to fetch all dependencies. Ensure there are no version conflicts.
5. **Prepare for Code Generation**\ In the respective Dart files, add the following directives:
   * In api\_client.dart: part 'api\_client.g.dart';
   * In models.dart: part 'models.g.dart'; These directives are necessary for generating the associated classes.
6. **Run Build Runner for Code Generation**\ Execute the following command to generate assets, fonts, and Retrofit-related classes: **dart run build\_*runner build*** *If you encounter conflicts, use:* ***dart run build\_*runner build --delete-conflicting-outputs**
7. **Verify Generated Files**\ After successful execution, verify the following:
   * A gen folder is created inside the lib directory containing fonts and assets.
   * api\_client.g.dart and models.g.dart files are generated in the network and model folders respectively.
8. **Begin Development**\ Once all setup steps are complete, you can proceed with your assigned tasks.