Flutter

Training Assignments

Building layouts

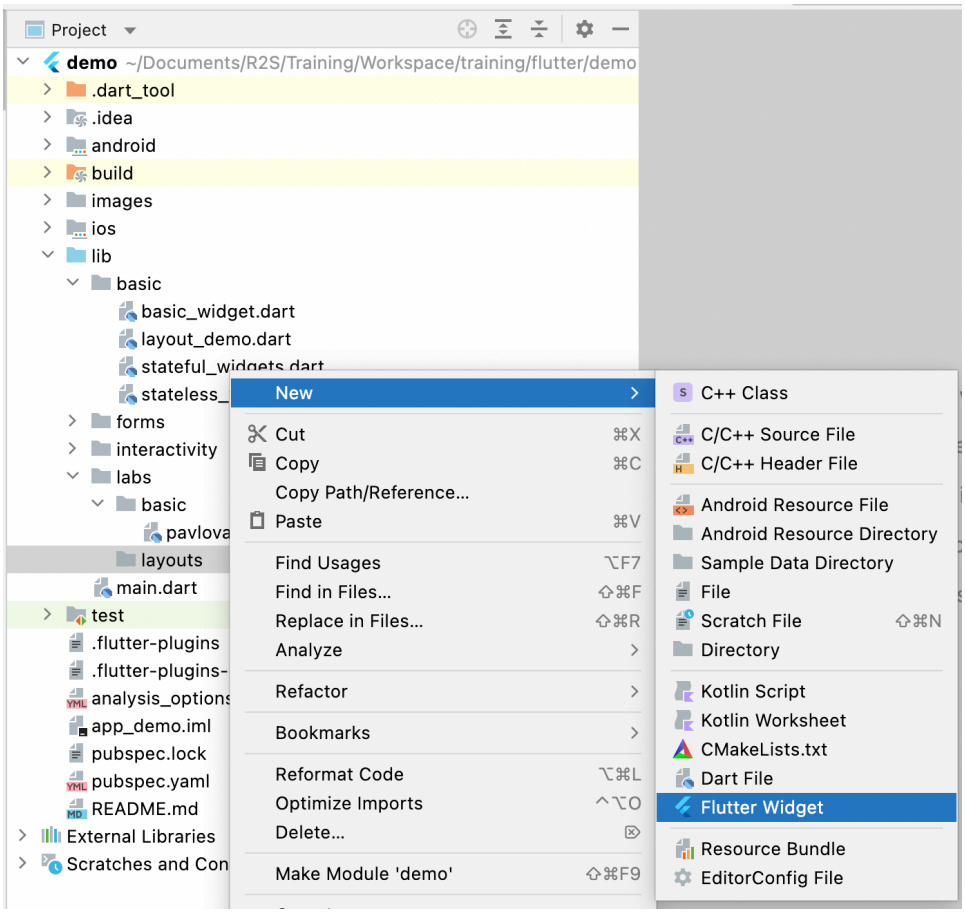
Overview

In this practice, you'll create a simple mobile Flutter app.

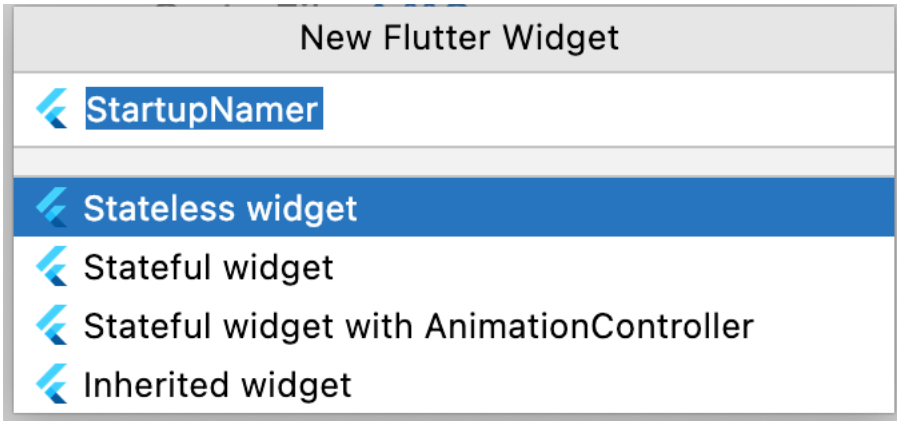
Tasks

1. Open the Android Studio, navigate to the project folder

+ Create Flutter Widget



+ Type named **StartupNamer** and press enter



2. Replace the contents of lib/startup_namer.dart

```dart
import 'package:flutter/material.dart';

class StartupNamer extends StatelessWidget {
  const StartupNamer({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ),  // AppBar
        body: const Center(
          child: Text('Hello World'),
        ),  // Center
      ),  // Scaffold
    );  // MaterialApp
  }
}
```

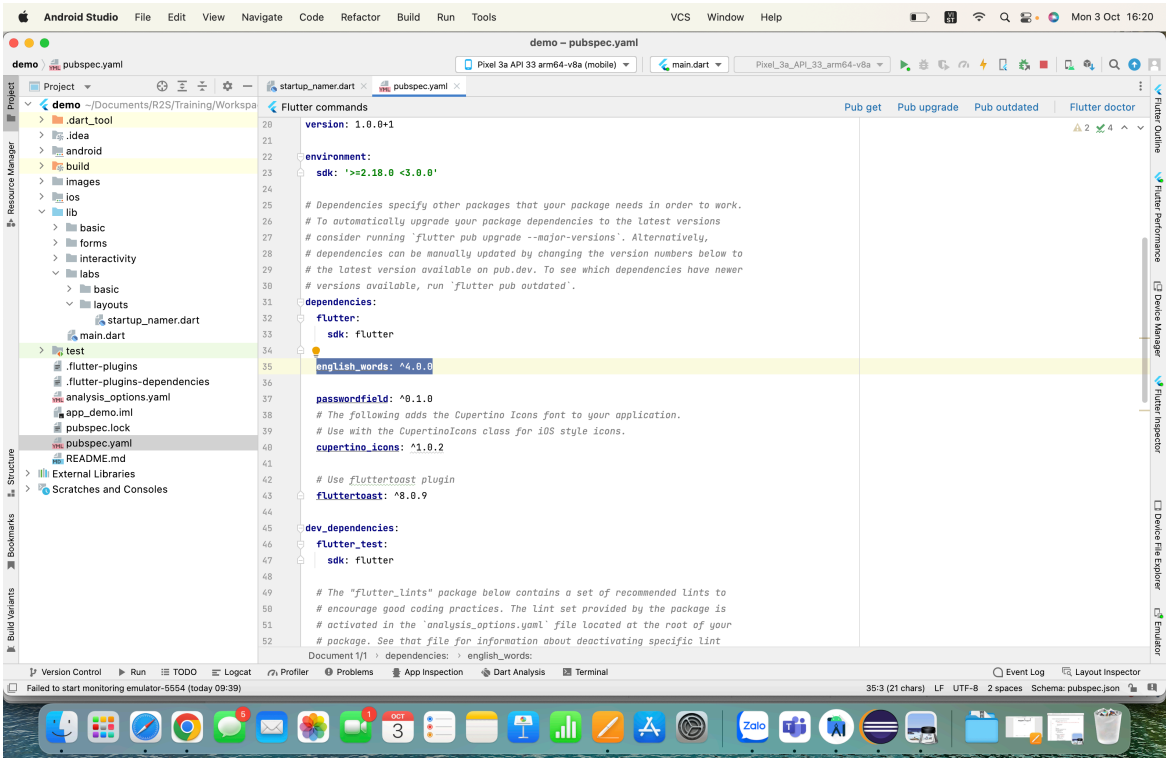
3. Use an external package

In this step, you'll start using an open-source package named english_words, which contains a few thousand of the most-used English words, plus some utility functions.

Add the **english_words** package as a dependency of this app:

+ english_words: ^4.0.0

+ Click ‘Pub get’



4. Add a stateful widget

+ In lib/startup_namer.dart, start typing **stful**, the editor asks if you want to create a **Stateful** widget. Press **Return** to accept. The boilerplate code for two classes appears, and the cursor is positioned for you to enter the name of your stateful widget. Enter **RandomWords** as the name of your widget. As you can see in the code below

```
class RandomWords extends StatefulWidget {
  const RandomWords({super.key});

  @override
  State<RandomWords> createState() => _RandomWordsState();
}

class _RandomWordsState extends State<RandomWords> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

+ Update the **_RandomWordsState**:

```
class _RandomWordsState extends State<RandomWords> {
  final _suggestions = <WordPair>[];
  final _biggerFont = const TextStyle(fontSize: 18);

  @override
  Widget build(BuildContext context) {
    _suggestions.addAll(generateWordPairs().take(50));

    return ListView.builder(
        itemCount: _suggestions.length,

        itemBuilder: (context, index) {
          if (index.isOdd) {
            return const Divider();
          }

          return ListTile(
            title: Text(
              _suggestions[index].asPascalCase,
              style: _biggerFont,
            ),  // Text
          );  // ListTile
        });  // ListView.builder
  }
}
```

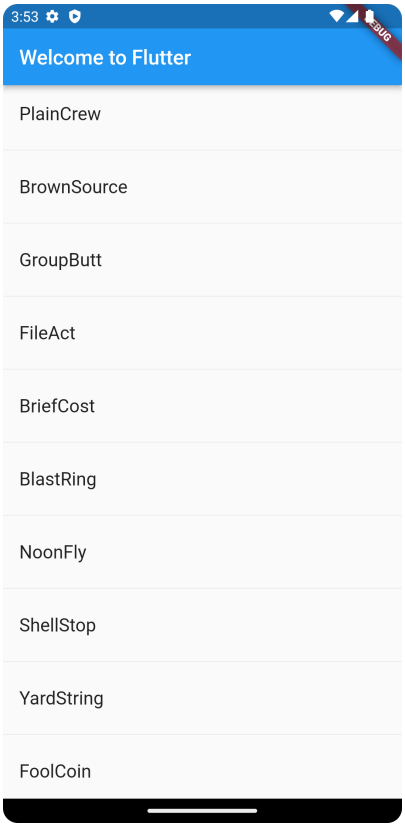
5. Update the **build** method for **StartupNamer**

```dart
class StartupNamer extends StatelessWidget {
  const StartupNamer({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ),  // AppBar
        body: const Center(
          child: RandomWords(),
        ),  // Center
      ),  // Scaffold
    );  // MaterialApp
  }
}
```

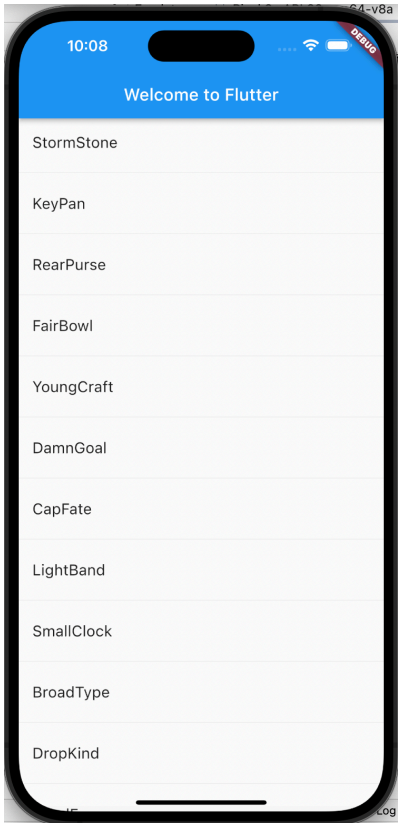
- Final source code

```dart
import 'package:english_words/english_words.dart';
import 'package:flutter/material.dart';

class StartupNamer extends StatelessWidget {
  const StartupNamer({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ),  // AppBar
        body: const Center(
          child: RandomWords(),
        ),  // Center
      ),  // Scaffold
    );  // MaterialApp
  }
}

class RandomWords extends StatefulWidget {
  const RandomWords({Key? key}) : super(key: key);

  @override
  State<RandomWords> createState() => _RandomWordsState();
}

class _RandomWordsState extends State<RandomWords> {...}
```

6. Run the app

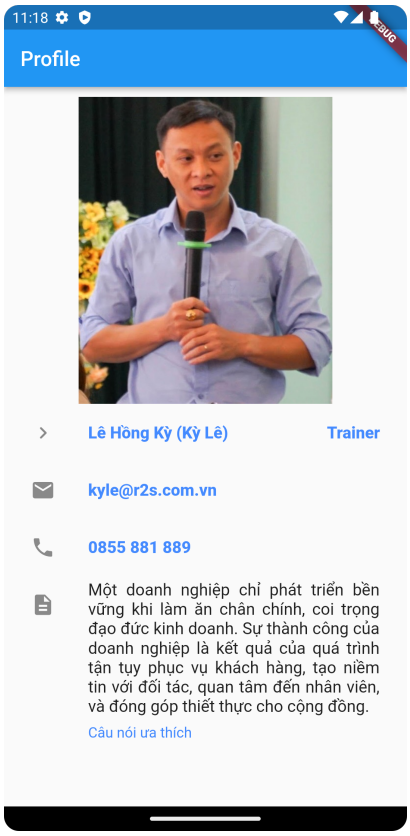


Android emulator



iOS simulator

Extra tasks



Android emulator

iOS simulator

--THE END—