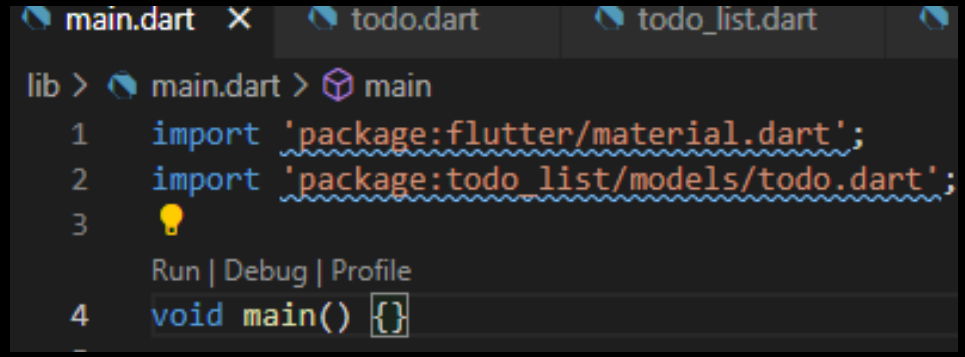
Step 1 – App Structure

Creating the app, the basic structure and the Todo class

Declutter the Flutter

- Start by opening VSCode and creating a new flutter project using the flutter create command.
- Call your app todo_list.
- Remove the templated counter app code from the main.dart file.
- Leave the main() entry point that dart requires, it should look like this:



Basic App Structure

- For flutter we need to initialise the flutter application by running a built-in command runapp()
- To this we pass our main app, which is a widget we create.
- The easiest way to do this with VS Code is to type "st" which should open an intellisense window with options.
- The second item should be stateless widget, press enter to insert the snippet.
- Now enter the name of your app: TodoApp.
- Now we can call runApp(const TodoApp()); in main to launch the application.

```
class TodoApp extends StatelessWidget {
  const TodoApp({ Key? key }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(

    );
  }
}
```

Material App

- Now we can create an application using a helpful widget called Material App.
- This sets up some theming and lets us set the main home page.
- Replace Container() with Material app and press return.
- Remember to use the Ctrl + Space to get a helper of what parameters can be passed to this material app constructor.
- Add title & home to the parameters.
- Home accepts the page (Still a widget) we are going to display as our homepage.
- Enter a string in the title for the app.

Home Page

- Now we need to create our Stateful Widget for the home page.
- This is going to be managing its own internal state through the built-in setstate() function.
- Like we did before create a new stateful widget this time called TodoHomePage.
- Stateful widgets are split up into two parts, the widget and the state.

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

  @override
  _TodoHomePageState createState() => _TodoHomePageState();
}
```

Home Page

- The state controls the build method and any dynamic properties that might define or influence its state.
- As the app is rebuilt it calls the state to determine what to display.

```
class _TodoHomePageState extends State<TodoHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold();
  }
}
```

- Now go back to our MaterialApp widget and add the TodoHomePage into its home parameter. *(You may need to apply const appropriately here)*

Adding to Material App

```
class TodoApp extends StatelessWidget {
  const TodoApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      title: 'Todo\'s!',
      home: TodoHomePage(),
    ); // MaterialApp
  }
}
```

Final Setup

- Now you should have the following elements:
- Main with runApp() function passing in our stateless TodoApp widget.
- TodoApp that builds a material app that takes a title and a homepage of a stateful TodoHomePage widget.
- Inside that then we finally have our build method that will display the widgets we want to build on our page.
- Replace the build methods container with a scaffold with an app bar that has a title of a Text Widget.
- Test that we have implemented this correctly by running the app and seeing if it launches correctly.

Todo Class

- To the project inside the lib folder create a new folder called models.
- This is going to hold our todo class.
- Create a new class by creating a new file don’t forget to name it appropriately, including the .dart extension.
- Open the file and create a class inside called Todo.
- Create a constructor and three properties for name, description and complete.
- Finally, we are going to override an inbuilt toString method.
- Type String and press space and the override will pop up, press enter.
- This will add the @override annotation for us.
- Remove the default code and return a string that is the name and the description in a single string.

Todo Class – Example

```dart
class Todo {
  final String name;
  final String description;
  final bool complete;

  Todo({required this.name, required this.description, this.complete = false});

  @override
  String toString() {
    //return name + " (" + description + ") "; old operation on string
    return "$name - ($description) "; //String interpolation
  }
}
```