TextField: retrieve and validate

Steps:

1. Create a TextEditingController.

2. Supply the TextEditingController to a TextField.

3. Display the current value of the text field.

Step-1: Create a TextEditingController.

- To retrieve the text a user has entered into a textfield, create a TextEditingController and supply it to a TextField.

  Note: Call dispose method of the TextEditingController when you're finished using it. This ensures that you discard any resources used by the object.

```
class _MyHomePageState extends State<MyHomePage> {
  // step 1. creating a TextEditingController
  final myController = TextEditingController();

  // step 1. contnuing... dispose method
  @override
  void dispose() {
    myController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
```

Step-2: Supply the TextEditingController to a TextField.

```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.all(15.0),
          child: Column(
            children: [

              TextField(
                //step 2. Supply the TextEditingConroller to a TextField.
                controller: myController,
                decoration: const InputDecoration(
                  hintText: 'Enter your text',
                  labelText: 'Text',
                ), // InputDecoration
              ), // TextField
```

Step-3: Display the current value of the TextField.

- Use the **text () method** provided by the **TextEditingController** to retrieve the string that the user has entered the text field.
-

```
      FloatingActionButton(
        child: const Icon(
          Icons.text_fields,
          size: 40.0,
        ), // Icon
        onPressed: () {
          showDialog(
            context: context,
            builder: (context) {
              return AlertDialog(
                content: Text(myController.text),
                ); // AlertDialog
              },
            );
          },
        ), // FloatingActionButton
```

The above code displays an alert dialog with the current value of the text field.

Reference: https://docs.flutter.dev/cookbook/forms/retrieve-input

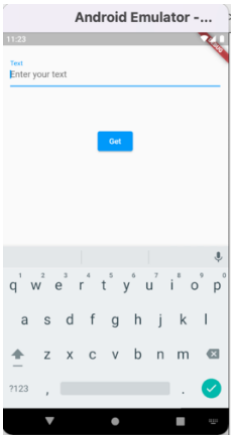
TextField: Validation

Implement the code as below:

main.dart

lib > main.dart > ...

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

Run | Debug | Profile
void main() {
  runApp(
    const MaterialApp(
      home: MyHomePage(),
    ), // MaterialApp
  );
}

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

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  // step 1. creating a TextEditingController
  final myController = TextEditingController();
  String value = '';

  // step 1. contnuing... dispose method
  @override
  void dispose() {
    myController.dispose();
    super.dispose();
  }

```



```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SafeArea(
      child: Padding(
        padding: const EdgeInsets.all(15.0),
        child: Column(
          children: [
            TextField(
              //step 2. Supply the TextEditingConroller to a TextField.
              controller: myController,
              decoration: const InputDecoration(
                hintText: 'Enter your text',
                labelText: 'Text',
              ), // InputDecoration
            ), // TextField
            const SizedBox(
              height: 30,
            ), // SizedBox
            Text(value),
            const SizedBox(
              height: 30,
            ), // SizedBox
            ElevatedButton(
              onPressed: () {
                setState(() {
                  value = myController.text;
                });
              },
              child: const Text(
                'Get',
              ), // Text
            ), // ElevatedButton
          ],
        ), // Column
      ), // Padding
    ), // SafeArea
  ); // Scaffold
}
}
```

Run the code- we will get the output as above where the text field is not validated.

1. check whether value in text field is null or not?

- make the bool value named **_validate** and set the default value as **false**.

```dart
class _MyHomePageState extends State<MyHomePage> {
  // step 1. creating a TextEditingController
  final myController = TextEditingController();
  String value = '';

  //validaton -1 making a boolean variable
  bool _validate = false;
```

2. when anyone presses on button then we will check if the value is empty then we will set **_validate** to **true**. We will use our Controller to check textField value.

```dart
ElevatedButton(
  onPressed: () {
    setState(() {
      value = myController.text;
      value.isEmpty ? _validate = true : _validate = false;
    });
  },
```

3. use **_validate** inour **errorText** if **_validate =true** then **errorText** will be displayed.

```dart
TextField(
  //step 2. Supply the TextEditingConroller to a TextField.
  controller: myController,
  decoration: InputDecoration(
    hintText: 'Enter your text',
    labelText: 'Text',
    errorText: _validate ? "field cannot be empty" : null,
  ), // InputDecoration
), // TextField
```