

Routes and Navigator in Flutter

Last Updated : 06 Mar, 2025

Route: Apps are the new trend. The number of apps available in the Play Store and App Store nowadays is quite a lot. The apps display their content in a full-screen container called **pages** or screens. In flutter, the pages or screens are called **Routes**. In android, these pages/screens are referred to as Activity and in iOS, it is referred to as ViewController. But, in a flutter, routes are referred to as Widgets. In Flutter, a Page / Screen is called a Route.

Creating routes in Flutter

A route can be written in the form of a “Class” in Dart using object-oriented concepts. Each route can be written as a separate class and has its own contents and UI.

Now let’s create two routes, each having unique App Bars and Elevated Buttons. The code is as follows:

```
class HomeRoute extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Geeks for Geeks'),  
        backgroundColor: Colors.green,  
      ),  
      body: Center(  
        child: ElevatedButton(  
          child: Text('Click Me!'),  
          onPressed: () {  
  
            // Contains the code that helps us  
            // navigate to the second route.  
  
          },  
        ),  
      ),  
    );  
  }  
}
```



```

    ),
  ),
);
}
}

class SecondRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Click Me Page"),
        backgroundColor: Colors.green,
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {

            // Contains the code that helps us
            // navigate to first route.
          },
          child: Text('Home!'),
        ),
      ),
    );
  }
}

```



Navigator in Flutter

As the name suggests, Navigator is a widget that helps us to navigate between the routes. The navigator follows stack method when dealing with the routes. Based on the actions made by the user, the routes are stacked one over the other and when pressed back, it goes to the most recently visited route. Navigator is a widget that follows a stack discipline.

– **Defining Home:** While navigating, the first thing that we need to do is to define or initialize the “home page”. The home page can be any route according to our needs. The home usually will be placed at the bottom of the navigator stack. Now let’s see how to initialize our `HomeRoute()` as our home page:



```
void main() {  
  runApp(MaterialApp(  
    home: HomeRoute(),  
  ));  
}
```

– **Navigating to a Page:** Since we have defined our Home, all the remaining is to navigate from home to another route of the app. For that the navigator widget has a method called `Navigator.push()`. This method pushes the route on top of the home, thereby displaying the second route. The code for pushing a route into the stack is as follows:

```
// Within the `HomeRoute` widget  
onPressed: () {  
  Navigator.push(  
    context,  
    MaterialPageRoute(builder: (context) => const  
SecondRoute()),  
  );  
},
```

– **Navigating Back to Home:** Now we have arrived at our destination, but how do we go back home? For that, the navigator has a method called `Navigator.pop()`. This helps us to remove the present route from the stack so that we go back to our home route. This can be done as follows:

```
// Within the SecondRoute widget  
onPressed: () {  
  Navigator.pop(context);  
}
```

Example of Routes and Navigator in Flutter

So, this is how we can navigate between two pages in an app. The whole code for the above flutter app is as follows:

main.dart:

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MaterialApp(
    debugShowCheckedModeBanner: false,
    home: HomeRoute(),
  ));
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Geeks for Geeks'),
        backgroundColor: Colors.green,
        foregroundColor: Colors.white,
      ),
      body: Center(
        child: ElevatedButton(
          style: ButtonStyle(
            backgroundColor: WidgetStateProperty.all(Colors.green),
            foregroundColor: WidgetStateProperty.all(Colors.white)),
          child: const Text('Click Me!'),
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => const
SecondRoute()),
            );
          },
        ),
      ),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Click Me Page"),
        backgroundColor: Colors.green,
        foregroundColor: Colors.white,
      ),
      body: Center(
        child: ElevatedButton(
          style: ButtonStyle(
            backgroundColor: WidgetStateProperty.all(Colors.green),
            foregroundColor: WidgetStateProperty.all(Colors.white)),
```



```
        onPressed: () {  
            Navigator.pop(context);  
        },  
        child: const Text('Home!'),  
    ),  
),  
);  
}  
}
```



To know more about `ElevatedButton` refer this article: [Flutter – ElevatedButton Widget](#)

Output: