



FLUTTER TUTORIALS

BY: **TECMAN**

TECMAN Lesson 9

Navigate to a new screen and back

Most apps contain several screens for displaying different types of information. For example, an app might have a screen that displays products. When the user taps the image of a product, a new screen displays details about the product.

Navigate to a new screen and back

Terminology: In Flutter, *screens* and *pages* are called *routes*.
The remainder of this recipe refers to routes.

Navigate to a new screen and back

In Android, a route is equivalent to an Activity. In iOS, a route is equivalent to a ViewController. In Flutter, a route is just a widget.

Navigate to a new screen and back

Navigate to a new route using the [Navigator](#). This recipe uses the following steps:

The next few sections show how to navigate between two routes, using these steps:

1. Create two routes.
2. Navigate to the second route using `Navigator.push()`.
3. Return to the first route using `Navigator.pop()`.

1. Create two routes

First, create two routes to work with. Since this is a basic example, each route contains only a single button. Tapping the button on the first route navigates to the second route. Tapping the button on the second route returns to the first route.

First, set up the visual structure:

```

class FirstRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Route'),
      ),
      body: Center(
        child: RaisedButton(
          child: Text('Open route'),
          onPressed: () {
            // Navigate to second route when tapped.
          },
        ),
      ),
    );
  }
}

class SecondRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Second Route'),
      ),
      body: Center(
        child: RaisedButton(
          onPressed: () {
            // Navigate back to first route when tapped.
          },
          child: Text('Go back!'),
        ),
      ),
    );
  }
}

```

2. Navigate to the second route using `Navigator.push()`

To switch to a new route, use the [`Navigator.push\(\)`](#) method. The `push()` method adds a Route to the stack of routes managed by the Navigator. Where does the Route come from? You can create your own, or use a [`MaterialPageRoute`](#), which is useful because it transitions to the new route using a platform-specific animation.

In the `build()` method of the `FirstRoute` widget, update the `onPressed()` callback:

2. Navigate to the second route using Navigator.push()

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

Animate a widget across screens

Use the [Hero](#) widget to animate a widget from one screen to the next. This recipe uses the following steps:

1. Create two screens showing the same image.
2. Add a Hero widget to the first screen.
3. Add a Hero widget to the second screen.

1. Create two screens showing the same image

- This example, display the same image on both screens.

Animate the image from the first screen to the second screen when the user taps the image. For now, create the visual structure; handle animations in the next steps.

```

class MainScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Main Screen'),
      ),
      body: GestureDetector(
        onTap: () {
          Navigator.push(context, MaterialPageRoute(builder: (_) {
            return DetailScreen();
          }));
        },
        child: Image.network(
          'https://picsum.photos/250?image=9',
        ),
      ),
    );
  }
}

class DetailScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: GestureDetector(
        onTap: () {
          Navigator.pop(context);
        },
        child: Center(
          child: Image.network(
            'https://picsum.photos/250?image=9',
          ),
        ),
      ),
    );
  }
}

```

2. Add a Hero widget to the first screen

To connect the two screens together with an animation, wrap the Image widget on both screens in a Hero widget. The Hero widget requires two arguments:

``tag`` An object that identifies the ``Hero``. It must be the same on both screens.

``child`` The widget to animate across screens.

2. Add a Hero widget to the first screen

```
Hero(  
  tag: 'imageHero',  
  child: Image.network(  
    'https://picsum.photos/250?image=9',  
  ),  
);
```

3. Return to the first route using `Navigator.pop()`

How do you close the second route and return to the first? By using the [`Navigator.pop\(\)`](#) method. The `pop()` method removes the current Route from the stack of routes managed by the navigator.

To implement a return to the original route, update the `onPressed()` callback in the `SecondRoute` widget:

3. Return to the first route using Navigator.pop()

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


Complete example

TECMAN LIMITED 2019

```

import 'package:flutter/material.dart';

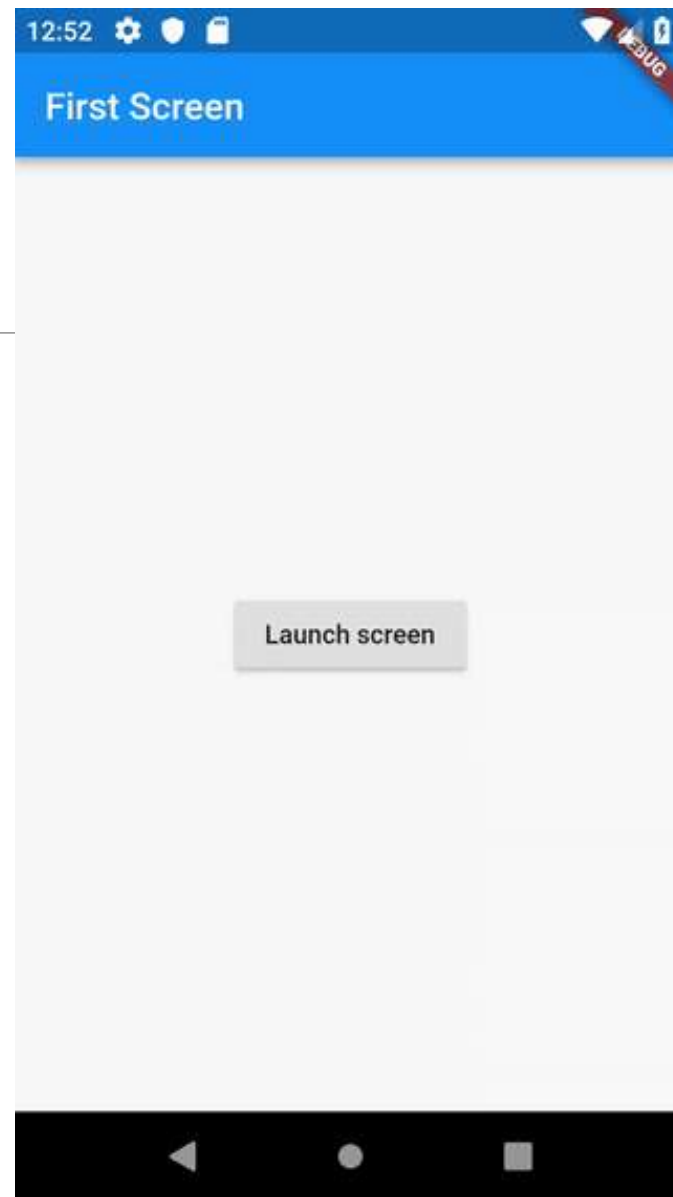
void main() {
  runApp(MaterialApp(
    title: 'Navigation Basics',
    home: FirstRoute(),
  ));
}

class FirstRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Route'),
      ),
      body: Center(
        child: RaisedButton(
          child: Text('Open route'),
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => SecondRoute()),
            );
          },
        ),
      ),
    );
  }
}

class SecondRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Second Route'),
      ),
      body: Center(
        child: RaisedButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Go back!'),
        ),
      ),
    );
  }
}

```

TECMAN LIMITED 2019



TECMAN LIMITED 2019

THANK YOU

TECMAN LIMITED 2019