

[Report errors to a service](#)[Navigate to a new screen and back](#)

Animate a widget across screens

[Docs](#) > [Cookbook](#) > [Navigation](#) > Animate a widget across screens

It's often helpful to guide users through an app as they navigate from screen to screen. A common technique to lead users through an app is to animate a widget from one screen to the next. This creates a visual anchor connecting the two 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

In 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.

Note: This example builds upon the [Navigate to a new screen and back](#) and [Handle taps](#) recipes.

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

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

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

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

  @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.

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

3. Add a [Hero](#) widget to the second screen

To complete the connection with the first screen, wrap the [Image](#) on the second screen with a [Hero](#) widget that has the same [tag](#) as the [Hero](#) in the first screen.

After applying the [Hero](#) widget to the second screen, the animation between screens just works.

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

Contents

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
- [Interactive example](#)

Note: This code is identical to what you have on the first screen. As a best practice, create a reusable widget instead of repeating code. This example uses identical code for both widgets, for simplicity.

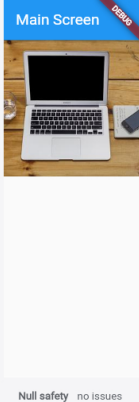
Interactive example

Dart

Install SDKFormatResetRun

```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(const HeroApp());
4
5 class HeroApp extends StatelessWidget {
6   const HeroApp({Key? key}) : super(key: key);
7
8   @override
9   Widget build(BuildContext context) {
10    return const MaterialApp(
11      title: 'Transition Demo',
12      home: MainScreen(),
13    );
14  }
15 }
16
17 class MainScreen extends StatelessWidget {
18   const MainScreen({Key? key}) : super(key: key);
19
20   @override
21   Widget build(BuildContext context) {
22     return Scaffold(
23       appBar: AppBar(
24         title: const Text('Main Screen'),
25       ),
26       body: GestureDetector(
```

Main Screen



Console

Null safety no issues

[Report errors to a service](#)

[Navigate to a new screen and back](#)



flutter-dev@ • terms • brand usage • security • privacy • español • 社区中文资源 • 한국어 • We stand in solidarity with the Black community. Black Lives Matter.

Except as otherwise noted, this work is licensed under a Creative Commons Attribution 4.0 International License, and code samples are licensed under the BSD License.