

flutter 图片加载

何景根

在 Flutter 中，提供了 Image 组件来描述图片组件，Image 的数据源可以是静态文件、内存以及网络。

在 Image Widget 中，有一个必须的属性 image，它对应的是一个 ImageProvider。ImageProvider 是一个抽象类，主要定义了图片数据获取的接口 load()，从不同的数据源获取图片需要实现不同的 ImageProvider。

加载本地图片

在 Flutter 中，加载本地图片比加载网络图片还要复杂。。。因为加载本地图片不仅仅是引用本地文件的资源路径，还要进行相关的配置。

步骤一：在工程的根目录创建一个 images 目录，并将图片拷贝至该目录中。

步骤二：在 pubspec.yaml 中的 flutter 部分添加配置：

```
assets:  
  - images/space.jpg
```

需要注意的是 yaml 的语法，空格以及缩进不能错，修改配置后热重载不会生效，需要重新启动项目。

步骤三：加载图片

```
Image(  
  image: AssetImage("images/space.jpg"),  
  width: 100,  
)
```

加载网络图片

转自 (<https://flutterchina.club/assets-and-images/>)

显示图片是大多数移动应用程序的基础。Flutter 提供了 Image Widget 来显示不同类型的图片。

为了处理来自 URL 的图像，请使用 Image.network 构造函数。

```
new Image.network(  
  'https://raw.githubusercontent.com/flutter/website/master/  
_includes/code/layout/lakes/images/lake.jpg',  
)
```

Bonus: GIF 动画

Image Widget 的一个惊艳的功能是：支持 GIF 动画！

```
new Image.network(  
  'https://github.com/flutter/plugins/raw/master/packages/  
video_player/doc/demo_ipod.gif?raw=true',  
)
```

```
);
```

完整的例子


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

void main() => runApp(new MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var title = 'Web Images';

    return new MaterialApp(
      title: title,
      home: new Scaffold(
        appBar: new AppBar(
          title: new Text(title),
        ),
        body: new Image.network(
          'https://github.com/flutter/website/blob/master/_includes/code/layout/lakes/images/lake.jpg?raw=true',
        ),
      ),
    );
  }
}
```

占位图过渡

当使用默认  widget 显示图片时，您可能会注意到它们在加载完成后会直接显示到屏幕上。这可能会让用户产生视

觉突兀。

相反，如果你最初显示一个占位符，然后在图像加载完显示时淡入，那么它会不会更好？我们可以使用[FadeInImage]来达到这个目的！

FadeInImage适用于任何类型的图片：内存、本地Asset或来自网上的图片。

在这个例子中，我们将使用 transparent_image 包作为一个简单的透明占位图。您也可以考虑按照 Assets 和图片指南使用本地资源来做占位图。

1.在 pubspec.yaml 里 dependencies 添加

```
dependencies:  
  flutter:  
    sdk: flutter  
  flutter_localizations:  
    sdk: flutter  
  cached_network_image: ^3.2.0
```

2.cmd 下进入到应用目录，运行 flutter packages get

```
new FadeInImage.memoryNetwork(  
  placeholder: kTransparentImage,  
  image: 'https://github.com/flutter/website/blob/master/_includes/code/layout/lakes/images/lake.jpg?raw=true',  
);
```

完整例子

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

```

import 'package:transparent_image/transparent_image.dart';

void main() {
  runApp(new MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final title = 'Fade in images';

    return new MaterialApp(
      title: title,
      home: new Scaffold(
        appBar: new AppBar(
          title: new Text(title),
        ),
        body: new Stack(
          children: <Widget>[
            new Center(child: new
CircularProgressIndicator()),
            new Center(
              child: new FadeInImage.memoryNetwork(
                placeholder: kTransparentImage,
                image:
                  'https://github.com/flutter/website/
blob/master/_includes/code/layout/lakes/images/lake.jpg?
raw=true',
            ),
          ],
        ),
      ),
    );
  }
}

```

```
}  
}
```

使用缓存图片

在某些情况下，在从网上下载图片后缓存图片可能会很方便，以便它们可以脱机使用。为此，我们可以使用 `cached_network_image` 包来达到目的。

除了缓存之外，`cached_image_network` 包在加载时还支持占位符和淡入淡出图片！

1.在 pubspec.yaml 里 dependencies 添加

```
dependencies:  
  flutter:  
    sdk: flutter  
  flutter_localizations:  
    sdk: flutter  
  cached_network_image: ^3.2.0
```

2.cmd 下进入到应用目录，运行 flutter packages get

```
new CachedNetworkImage(  
  imageUrl: 'https://github.com/flutter/website/blob/  
master/_includes/code/layout/lakes/images/lake.jpg?  
raw=true',  
);
```

添加一个占位符

`cached_network_image` 包允许我们使用任何 Widget 作为占位符！在这个例子中，我们将在图片加载时显示一个进度

卷。

```
new CachedNetworkImage(  
  placeholder: new CircularProgressIndicator(),  
  imageUrl: 'https://github.com/flutter/website/blob/  
master/_includes/code/layout/lakes/images/lake.jpg?  
raw=true',  
);
```

完整的例子

```
import 'package:flutter/material.dart';  
import 'package:cached_network_image/  
cached_network_image.dart';  
  
void main() {  
  runApp(new MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    final title = 'Cached Images';  
  
    return new MaterialApp(  
      title: title,  
      home: new Scaffold(  
        appBar: new AppBar(  
          title: new Text(title),  
        ),  
        body: new Center(  
          child: new CachedNetworkImage(  
            placeholder: new CircularProgressIndicator(),
```

```
        imageUrl:
            'https://github.com/flutter/website/blob/
master/_includes/code/layout/lakes/images/lake.jpg?
raw=true',
    ),
),
);
}
```