使用Flutter的image_picker插件实现设备的相册的访问和拍照

江上清风山间明月

文章目录

需求描述

在应用开发时，我们有很多场景要使用到更换图片的功

能，即将原本的图像替换设置成其他的图像，从设备的相册或相机中选择图片或拍照的方式来更换图像。那么用 Flutter 要如何实现从设备的相册或相机中选择图片或拍照呢?
其实很简单一个插件就能解决，而且使用起来也很方便。

Flutter插件image_picker的介绍

image_picker 是 Flutter 中的一个插件，它提供了一个简单且易于使用的方法，用于从设备的相册或相机中选择图片或拍照。
使用 image_picker 插件，您可以轻松地实现以下功能：
从相册中选择图片：允许用户从设备的相册中选择一张图片。
拍照：允许用户使用设备的相机拍摄一张照片。

使用步骤

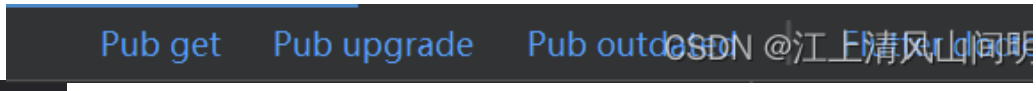
以下是使用 image_picker 插件的基本步骤：

1、添加依赖

在 pubspec.yaml 文件中添加 image_picker 依赖：

```
dependencies:
  flutter:
    sdk: flutter
  image_picker: ^0.8.7+5 # 请确保使用最新的版本
```

运行 flutter pub get 命令，以获取依赖的插件。如果你是使用的 Android Studio 可以直接在编辑 pubspec.yaml 文件后，选择 Pub upgrade 如图：



2、导入

在需要调用图片选择或拍照的地方导入

```
import 'package:image_picker/image_picker.dart';
```

例子

使用 ImagePicker 类的静态方法来选择图片或拍照。

以下是一个简单的示例，演示如何使用 image_picker 插件从相册中选择图片并显示在应用中：

```
  Container(
            padding: EdgeInsets.all(16),
            color: Colors.grey[200],
            child: Row(
              children: [
                // 使用 FutureBuilder 来等待异步操作完成，避免 LateInitializationError 错误
                FutureBuilder(
                  future: _loadPrefs(),
                  builder: (BuildContext context, AsyncSnapshot<File> snapshot) {
                    if (snapshot.connectionState == ConnectionState.done) {
                      return InkWell(
                        onTap: () {
                          showDialog(
```

```
                                context: context,
                                builder: (BuildContext
context) {
                                  return AlertDialog(
                                    title: Text('选择头
像'),
                                    actions: [
                                      TextButton(
                                        child: Text('从
相册选择'),
                                        onPressed: ()
async {

Navigator.of(context).pop();
                                          final
pickedImage = await ImagePicker().pickImage(source:
ImageSource.gallery);
                                          if
(pickedImage != null) {

_updateSelectedImage(File(pickedImage.path));

_saveImagePath(pickedImage.path);
                                          }
                                        },
                                      ),
```

```
                                    TextButton(
                                      child: Text('拍
照'),
                                      onPressed: ()
async {
Navigator.of(context).pop();
                                        final
pickedImage = await ImagePicker().pickImage(source:
ImageSource.camera);
                                        if
(pickedImage != null) {
_updateSelectedImage(File(pickedImage.path));
_saveImagePath(pickedImage.path);
                                        }
                                      },
                                    ),
                                  ],
                                );
                              },
                            );
                          },
                          // 使用条件运算符来检查
_selectedImage 是否为 null，并使用默认头像路径
```

```
                      child: CircleAvatar(
                        radius: 40,
                        backgroundImage:
snapshot.data != null ? FileImage(snapshot.data!)
as ImageProvider<Object>?: AssetImage('assets/
touxiang.jpg'),
                      ),
                    );
                  } else {
                    return
CircularProgressIndicator();
                  }
                },
              ),
              SizedBox(width: 16),
              Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  Text(
                    '江上清风山间明月',
                    style: TextStyle(fontSize:
18),
                  ),
                  Text(
                    '用户ID: 123456',
                    style: TextStyle(fontSize:
```

```
14, color: Colors.grey),
                    ),
                  ],
                ),
              ],
            ),
          ),
```

在上面的示例中，我们使用 ImagePicker 类中的 pickImage 方法来从相册中选择一张图片或者选择相机。如果用户选择了一张图片，我们将通过 pickedFile.path 获取到图片的文件路径，然后将其转换为 File 对象。

```
ImagePicker().pickImage(source:
ImageSource.gallery);
```

如果用户选择了从相机拍照，通过调用 pickImage 方法时指定 ImageSource.camera 来实现。

```
await ImagePicker().pickImage(source:
ImageSource.camera);
```

完整的代码如下：

```
import 'package:flutter/material.dart';
import 'dart:io';
import 'package:image_picker/image_picker.dart';
import 'package:shared_preferences/
shared_preferences.dart';

class SettingsPage extends StatefulWidget {
```

```
  const SettingsPage({Key? key}) : super(key: key);

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

class _SettingsPageState extends
State<SettingsPage> {
  late File _selectedImage;
  late SharedPreferences _prefs;

  @override
  void initState() {
    super.initState();
    // 调用 _loadPrefs 方法来初始化 _selectedImage 变
量
    _loadPrefs();
  }

  Future<File> _loadPrefs() async {
    _prefs = await SharedPreferences.getInstance();
    final imagePath =
_prefs.getString('imagePath');
    if (imagePath != null) {
      return File(imagePath);
    } else {
```

```
      return File('assets/touxiang.jpg');
    }
  }

  Future<void> _saveImagePath(String imagePath)
async {
    await _prefs.setString('imagePath', imagePath);
  }

  Future<void> _pickImage(ImageSource source) async
{
    final picker = ImagePicker();
    final pickedImage = await
picker.pickImage(source: source);
    if (pickedImage != null) {
      setState(() {
        _selectedImage = File(pickedImage.path);
      });
      _saveImagePath(pickedImage.path);
    }
  }


  void _updateSelectedImage(File image) {
    setState(() {
      _selectedImage = image;
    });
```

```
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: ListView(
        children: [
          Container(
            padding: EdgeInsets.all(16),
            color: Colors.grey[200],
            child: Row(
              children: [
                // 使用 FutureBuilder 来等待异步操作完成，避免 LateInitializationError 错误
                FutureBuilder(
                  future: _loadPrefs(),
                  builder: (BuildContext context, AsyncSnapshot<File> snapshot) {
                    if (snapshot.connectionState == ConnectionState.done) {
                      return InkWell(
                        onTap: () {
                          showDialog(
                            context: context,
                            builder: (BuildContext context) {
                              return AlertDialog(
```

```
                                    title: Text('选择头
像'),
                                    actions: [
                                      TextButton(
                                        child: Text('从
相册选择'),
                                        onPressed: ()
async {
Navigator.of(context).pop();
                                          final
pickedImage = await ImagePicker().pickImage(source:
ImageSource.gallery);
                                          if
(pickedImage != null) {
_updateSelectedImage(File(pickedImage.path));
_saveImagePath(pickedImage.path);
                                          }
                                        },
                                      ),
                                      TextButton(
                                        child: Text('拍
照'),
                                        onPressed: ()
```

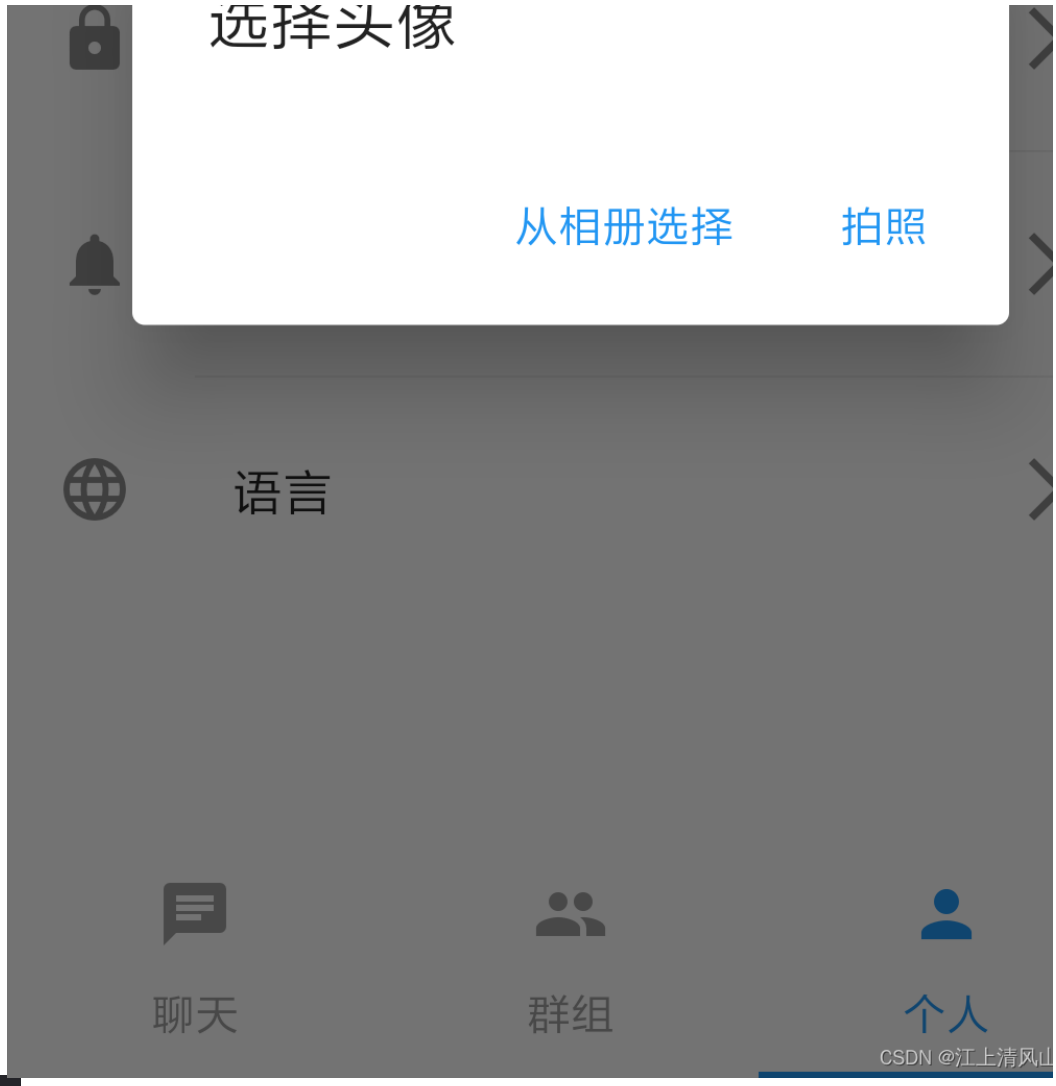
```
async {

Navigator.of(context).pop();
                                    final
pickedImage = await ImagePicker().pickImage(source:
ImageSource.camera);
                                    if
(pickedImage != null) {

_updateSelectedImage(File(pickedImage.path));

_saveImagePath(pickedImage.path);
                                    }
                                  },
                                ),
                              ],
                            );
                          },
                        );
                      },
                      // 使用条件运算符来检查
_selectedImage 是否为 null，并使用默认头像路径
                      child: CircleAvatar(
                        radius: 40,
                        backgroundImage:
snapshot.data != null ? FileImage(snapshot.data!)
as ImageProvider<Object>?: AssetImage('assets/
```

```
touxiang.jpg'),
                        ),
                    );
                  } else {
                    return
CircularProgressIndicator();
                  }
                },
              ),
              SizedBox(width: 16),
              Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  Text(
                    '江上清风山间明月',
                    style: TextStyle(fontSize:
18),
                  ),
                  Text(
                    '用户ID: 123456',
                    style: TextStyle(fontSize:
14, color: Colors.grey),
                  ),
                ],
              ),
            ],
```

```
            ),
          ),
          Divider(indent: 60,),
          SettingItem(icon: Icons.person, title:
'个人信息'),
          Divider(indent: 60,),
          SettingItem(icon: Icons.lock, title: '账
号与安全'),
          Divider(indent: 60,),
          SettingItem(icon: Icons.notifications,
title: '消息通知'),
          Divider(indent: 60,),
          SettingItem(icon: Icons.language, title:
'语言'),
          // 添加更多的设置项...
        ],
      ),
    );
  }
}

class SettingItem extends StatelessWidget {
  final IconData icon;
  final String title;

  const SettingItem({required this.icon, required
```

```
this.title});

  @override
  Widget build(BuildContext context) {
    return ListTile(
      leading: Icon(icon),
      title: Text(title),
      trailing: Icon(Icons.arrow_forward_ios),
      onTap: () => {},
    );
  }
}
```

效果如下图：



总结

总结一下，image_picker 插件是 Flutter 中一个方便的工具，用于在应用中从相册中选择图片或拍摄照片。使用这个插件，您可以轻松地实现图片选择和拍照功能，十分方便的实现替换图像的功能。

其他插件

Flutter 提供了许多插件和包，可以帮助你在应用程序中操作相机和相册。以下是一些常用的 Flutter 插件，用于处理相机

和相册功能：

camera

camera 插件是一个广泛使用的 Flutter 插件，它提供了用于访问和控制设备相机的 API。你可以使用 camera 插件来捕获照片和录制视频。

image_picker

image_picker 插件允许用户从设备的相册中选择图片，也可以使用相机拍摄新照片。这是一个非常方便的插件，用于选择和处理图片。

flutter_image_compress：

flutter_image_compress 这个插件可以帮助你压缩和处理图片，特别是在你从相机或相册获取图片后，你可能需要将其进行压缩以减小文件大小。

path_provider

path_provider 插件用于获取设备上特定目录的路径，这对于存储从相机或相册选择的图片文件以及其他数据非常有用。

image_cropper

image_cropper 插件用于裁剪图片。如果你需要让用户选择图片后进行裁剪，这个插件是一个很好的选择。

这些插件在 Flutter 社区中非常受欢迎，并提供了丰富的功能，以便于在你的 Flutter 应用程序中操作相机和相册。你可以通过 Flutter 官方的包管理工具 pub 来安装和使用这些插

件。要了解更多详情和示例，请查阅各个插件的文档和示例代码。请注意，插件的版本和功能可能会随时间而变化，因此请查看它们的GitHub页面或Flutter包管理器以获取最新信息。

掌握以上插件对Flutter操作相机和相册的操作基本就没有问题了。