

Flutter 检测应用版本号、服务器下载文件 以及实现 App 自动升级、安装

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

目录

1、 Android App 升级执行流程	1
2、 升级 app 之前的准备工作配置权限	2
3、 Android 升级 app 涉及的 API 库	2
4、 获取版本信息	2
5、 获取文件存储路径	3
6、 下载文件	3
7、 打开文件	4
8、 检测权限	4
9、 注意事项	4

1、 Android App 执行自动升级流程

- 1、获取本地版本号
- 2、请求服务器的 api 接口获取服务器保存版本号
- 3、本地版本和服务器版本不一致提示升级，弹窗提示用户是否更新
- 4、用户确定升级，调用文件传输方法下载 apk 文件
- 5、监听下载进度
- 6、下载完成打开 Apk 进行安装

注意：在 ios 中没法直接下载安装，如果版本不一致直接跳转到 ios 应用对应的应用市场就可以了。

2、升级 app 之前的准备工作配置权限

配置 AndroidManifest.xml 文件

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />
```

3、Android 升级 app 涉及的 API 库

插件名称	描述	插件地址
package_info	检测版本号	https://pub.dev/packages/package_info_plus
path_provider	获取文件存储路径	https://pub.dev/packages/path_provider
Dio	下载文件	https://pub.dev/packages/dio
open_file	打开文件插件	https://pub.dev/packages/open_file https://github.com/phonegap100/open_file.git
permission_handler	检测 Apk 是否拥有权限	https://pub.dev/packages/permission_handler

4、获取版本信息

https://pub.dev/packages/package_info

```
PackageInfo packageInfo = await PackageInfo.fromPlatform();
String appName = packageInfo.appName;
String packageName = packageInfo.packageName;
String version = packageInfo.version;
String buildNumber = packageInfo.buildNumber;

print("appName:${appName}");
print("packageName:${packageName}");
print("version:${version}");
print("buildNumber:${buildNumber}");
```

5、获取文件存储路径

https://pub.dev/packages/path_provider

```
Directory tempDir = await getTemporaryDirectory();
String tempPath = tempDir.path;

Directory appDocDir = await getApplicationDocumentsDirectory();
String appDocPath = appDocDir.path;

var directory = await getExternalStorageDirectory();

String storageDirectory=directory.path;

print("tempPath:${tempPath}");

print("appDocDir:${appDocPath}");

print("StorageDirectory:${storageDirectory}");
```

6、下载文件

<https://pub.dev/packages/dio>

```
final directory = await getExternalStorageDirectory();
String _localPath = directory.path;
String _appName = "aaa.apk";
String _savePath="$_localPath/$_appName";

String apkUrl = "https://jd.itying.com/jdshop.apk";

///参数一 文件的网络储存 URL
///参数二 下载的本地目录文件
///参数三 下载监听
Dio dio = new Dio();
await dio.download(apkUrl,_savePath,
    onReceiveProgress: (received, total) {
    if (total != -1) {
        ///当前下载的百分比例
```

```
print((received / total * 100).toStringAsFixed(0) + "%");  
}  
});  
print(_savePath);  
await OpenFile.open(_savePath,type: "application/vnd.android.package-archive");
```

7、打开文件

https://pub.dev/packages/open_file

```
OpenFile.open("${_localPath}/jdshop.apk");  
  
await OpenFile.open(_savePath,type: "application/vnd.android.package-archive");
```

8、检测权限

https://pub.dev/packages/permission_handler

```
Future<bool> _checkPermission() async {  
  if (Theme.of(context).platform == TargetPlatform.android) {  
    final status = await Permission.storage.status;  
    if (status != PermissionStatus.granted) {  
      final result = await Permission.storage.request();  
      if (result == PermissionStatus.granted) {  
        return true;  
      }  
    } else {  
      return true;  
    }  
  }  
  return false;  
}
```

9、注意事项

- 1、服务器的 App 版本必须大于本地 App 版本
- 2、本地 App 和服务器 App 的包名称 签名必须一致，这样的话服务器的包才可以替换本地的包。