

APPLink 接入

1.前言

在 `Android M` 之前，点击一个链接会产生一个对话框，询问用户打开哪个应用 - 包括浏览器应用。但是谷歌在 `Android M` 实现了一个自动认证（`auto-verify`）机制，让开发者可以避开这个弹出框，使用户不必去选择一个列表，直接跳转到他们的app中。

`Android App Links` 有以下几点好处：

- 安全性/特殊性：由于 `Android App Links` 使用了 `HTTP/HTTPS URL` 的方式向开发者的服务器进行连接认证，所以其他应用无法使用我们的链接
- 无缝的用户体验：当用户未安装我们的应用时，由于使用的是 `HTTP/HTTPS URL`，会直接打开一个网页，我们可以在这个网页中展示应用介绍等，而不是显示404或者是其他错误页面
- 支持 `Instant Apps`：可以使用 `App Links` 直接打开一个未安装的 `Instant App`
- 支持 `Google Search` 或其他浏览器：用户可以直接在 `Google Search` / `Google Assistant` / `手机浏览器` / `屏幕搜索` 中直接通过点击一个URL来打开我们的指定页面

2.个推平台域名注册

参见：[个推文档 > 开发者中心 > AppLink配置说明书，APPLink配置>4. 在【域名配置】对域名进行配置](#)

3.创建intent filter

我们在此处先假设用户是通过 `https://link.applk.cn/getui` 或 `http://link.applk.cn/getui` 来打开我们的应用的。

`Android Studio 2.3` 以后提供了 `App Links Assistant` 来帮助开发者快速在 `AndroidManifest.xml` 中创建需要配置的 `intent filter`，使用 `App Links Assistant` 有以下几个步骤：

- 点击 `Android Studio` 的菜单栏中的 `Tools > App Links Assistant`
- 点击 `Open URL Mapping Editor`，然后在对话框底部点击 `+` 去添加一个新的 `URL mapping`
- 在弹出的 `Add URL Mapping` 对话框中输入对应的内容，包括 `Host`、`Path`、`Activity`，输入完成后点击 `OK`

Add URL Mapping

Basic URL Mapping
You can add or edit your URL mapping here

Host

Path

[How it works](#)

Activity

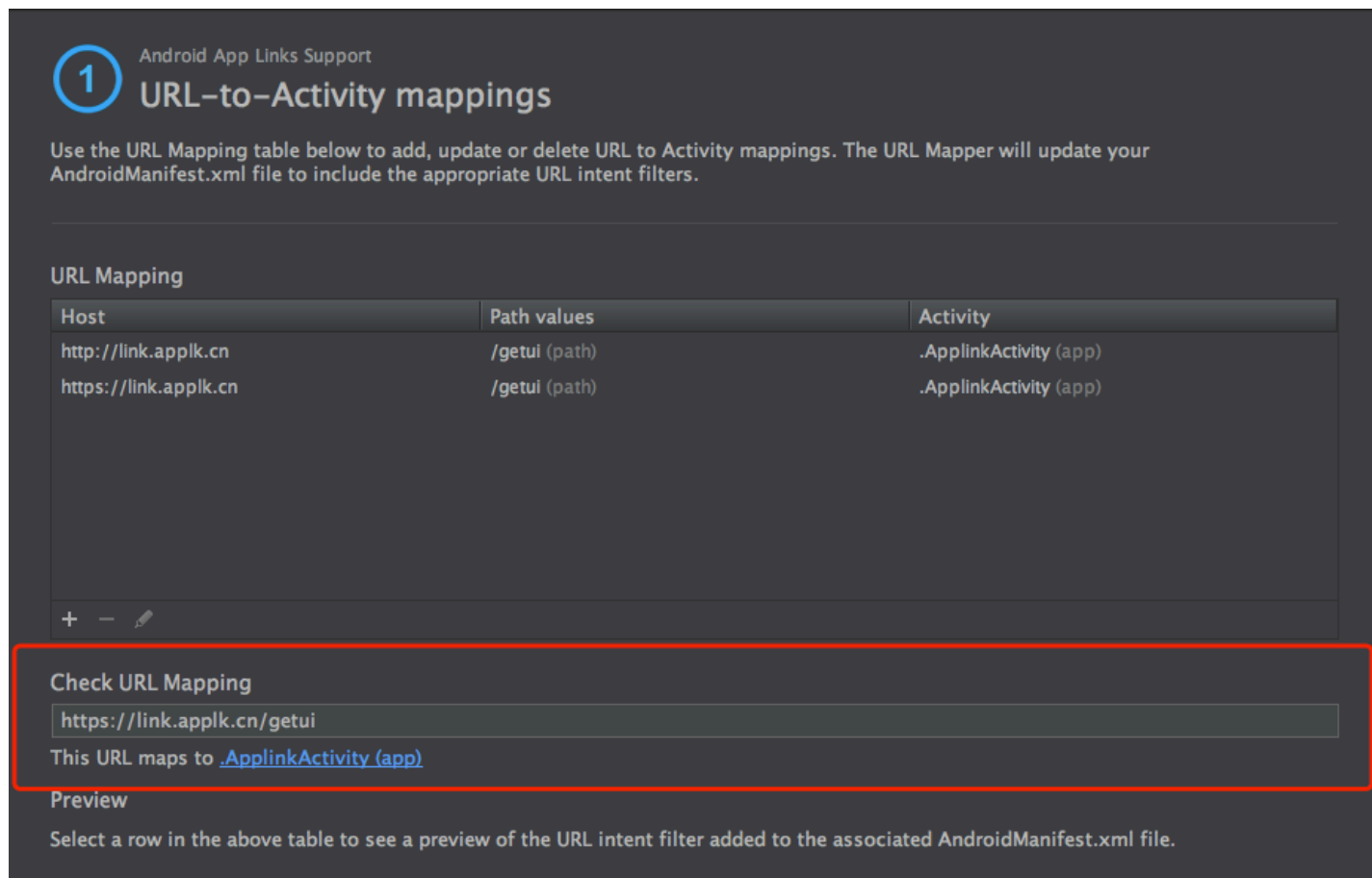
使用 App Links Assistant 在 manifest 文件中自动生成的内容如下：

```
<activity android:name=".ApplinkActivity">
    <intent-filter android:autoVerify="true">
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data
            android:scheme="http"
            android:host="link.applk.cn"
            android:path="/getui" />
    </intent-filter>
    <intent-filter android:autoVerify="true">
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data
            android:scheme="https"
            android:host="link.applk.cn"
            android:path="/getui" />
    </intent-filter>
</activity>
```

4.检查URL Mapping是否配置正确

App Links Assistant 提供了检查 URL Mapping 是否配置正确的快捷方式，操作如下：

- 点击 `Open URL Mapping Editor` , 然后在 `Check URL Mapping` 对话框中输入 `URL` ,当输入一个能够成功匹配到 `Acitivity` 的 `URL` 后, 输入框下方会显示 `This URL maps to xxxxx(app)`



5.处理App Links进入应用的场景

通过 `App Links Assistant` -> `Select Activity` 选择之前配置的 `URL` 对应的 `Activity` , 点击 `Insert Code` 即可在 `onCreate` 方法中插入获取从 `App Links` 跳转而来的 `URL` 的代码, 生成代码如下

```
// ATTENTION: This was auto-generated to handle app links.
Intent appLinkIntent = getIntent();
String appLinkAction = appLinkIntent.getAction();
Uri appLinkData = appLinkIntent.getData();
```

6.调用个推SDK发送App Links点击回执

如果需要统计用户通过个推短信补量的方式下发后的点击数量, 需要在获取到 `appLinkData` 后, 调用个推sdk的接口进行回执

```
if (appLinkData != null) {  
    PushManager.getInstance().sendApplinkFeedback(getApplicationContext(), appLink  
Data.toString());  
}
```

7.检查assetlinks.json是否上传成功

- 为了在应用安装成功后，系统能自动验证该应用是否有权使用对应的域名，系统会向 `https://link.applk.cn/.well-known/assetlinks.json` 请求数据，根据获取到的文件内容，验证应用域名的合法性。
- 如果您是通过个推的App Links短信补量接入的，只需要完成第二步中的域名配置，审核成功后，个推会在对应的域名下放置正确的文件。
- 如果您是自己集成 `App Links`，则需要通过 `App Links Assistant` -> `Open Digital Asset Links File Generator` -> `Generate Digital Asset Links file` 的方式生成 `assetlinks.json` 文件，然后将该文件放置到正确的域名地址中。
- 通过 `App Links Assistant` -> `Open Digital Asset Links File Generator` -> `Generate Digital Asset Links file` -> `Link and Verify` 可以检测是否正确的在服务器中放置配置文件，检测成功的话，显示如下图：



Declare Website Association

To associate your website with your app, enter the information below to generate a Digital Asset Links file and upload to your website.

Site domain

Application ID

SHA256 Fingerprint of signing certificate

Specify either the signing config or the keystore file used to sign your app to obtain the SHA256 fingerprint.

- ☒ Signing config ☐ Select keystore file



Reminder: if you generate the DAL file with a debug keystore, it won't work with your release build.

Generate Digital Asset Links file

Preview:

```
{
  "relation": ["delegate_permission/common.handle_all_urls"],
  "target": {
    "namespace": "android_app",
    "package_name": "com.getui.applink",
    "sha256_cert_fingerprints":
      ["F0:DA:F8:6D:09:BC:DE:6F:B7:14:BD:2B:CA:FD:00:36:04:B5:A1:32:42:A1:26:3D:9D:A7:29:D6:EE:52:DA:24"]
  }
}
```

To complete associating your app with your website, save the above file to

<https://link.applk.cn/.well-known/assetlinks.json>

Save file

Complete the association

Link your Digital Asset Links file with your app and verify that it has been uploaded to the current location.

Link and Verify

- ✓ Added asset statements to [strings.xml](#)
- ✓ Linked strings to [AndroidManifest.xml](#)
- ✓ Added autoVerify to intent filter elements.
- ✓ Success! Your app is associated with the selected [domain](#)