

了解 AndroidManifest.xml 文件

如果你正在進入 Android 開發，你會遇到的第一個文件之一就是 `AndroidManifest.xml`。它就像你的應用程式的藍圖—它在應用程式運行之前告訴 Android 系統有關你應用程式的一切。今天，我們將分解一個名為「Flower」（套件名稱：`com.lzw.flower`）的應用程式的範例清單文件，並探索其關鍵組件、概念和模式。

什麼是 `AndroidManifest.xml`？ `AndroidManifest.xml` 文件是每個 Android 應用程式所需的配置文件。它位於項目的根目錄，並聲明了應用程式的基本信息，如應用程式的套件名稱、權限、組件（例如活動）以及它所需的硬件/軟件功能。可以將其視為應用程式的身份證，Android 操作系統會讀取它。

讓我們逐步走過這個範例。

清單的結構

這是我們要處理的清單（稍微簡化以便閱讀）：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.lzw.flower"
    android:versionCode="8"
    android:versionName="1.5.2">

    <uses-sdk android:minSdkVersion="14" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

    <application
        android:label="@string/app_name"
        android:icon="@drawable/icon128"
        android:name=".base.App"
        android:theme="@style/AppTheme">
```

```
<activity android:name=".deprecated.CameraActivity" android:screenOrientation="landscape" />
<activity android:name=".base.SplashActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".draw.DrawActivity" android:screenOrientation="landscape" />
<activity android:name=".result.ResultActivity" android:screenOrientation="landscape" />
<activity android:name=".material.MaterialActivity" android:screenOrientation="landscape" />
<activity android:name=".activity.PhotoActivity" android:screenOrientation="landscape" />
<activity android:name=".activity.LoginActivity" android:screenOrientation="portrait" />
</application>
</manifest>
```

現在，讓我們將其分解為其核心部分，並解釋其背後的概念。

1. 根 `<manifest>` 元素

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.lzw.flower"
    android:versionCode="8"
    android:versionName="1.5.2">
```

- `xmlns:android`：這定義了 Android 特定屬性的 XML 命名空間。這是你在每個清單中都會看到的標準佈局。
- `package`：這是你應用程式的唯一識別符（例如，`com.lzw.flower`）。它也是你的 Java/Kotlin 類的默認命名空間。
- `android:versionCode`：一個內部整數（這裡是 8），用於追蹤版本。它隨每次更新而增加。
- `android:versionName`：一個可讀的版本字符串（這裡是 1.5.2），顯示給用戶。

概念：`<manifest>` 標籤設置應用程式的身份和版本控制，確保系統知道它在處理哪個應用程式以及如何處理更新。

2. SDK 版本與 `<uses-sdk>`

```
<uses-sdk android:minSdkVersion="14" />
```

- `android:minSdkVersion`：指定應用程式支持的最低 Android API 級別。API 14 對應於 Android 4.0 (Ice Cream Sandwich)。

概念：這確保了兼容性。運行 Android 版本低於 4.0 的設備無法安裝此應用程式。這裡沒有 `targetSdkVersion` 或 `maxSdkVersion`，但可以添加它們以進一步微調兼容性。

3. 使用 `<uses-permission>` 的權限

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

這個應用程式請求了幾個權限：
- `CAMERA`：訪問設備的相機。
- `WRITE_EXTERNAL_STORAGE`：將文件（例如照片）保存到外部存儲。
- `INTERNET`：用於網絡訪問。
- `ACCESS_NETWORK_STATE`：檢查網絡連接。
- `READ_PHONE_STATE`：訪問設備信息（例如 IMEI）。
- `ACCESS_WIFI_STATE`：檢查 Wi-Fi 狀態。

概念：Android 使用權限系統來保護用戶隱私和安全。這些聲明告訴系統（和用戶）應用程式需要哪些敏感功能。在 Android 6.0 (API 23) 之後，危險權限（例如 `CAMERA`）也需要在應用程式代碼中進行運行時請求。

4. 使用 `<uses-feature>` 的功能

```
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
```

- `android.hardware.camera`：聲明應用程式需要相機。
- `android.hardware.camera.autofocus`：指定相機必須支持自動對焦。

概念：與權限不同，`<uses-feature>` 標籤在 Google Play 商店上過濾應用程式。如果設備沒有相機或自動對焦，應用程式將不會出現可安裝，除非這些標記為可選 (`android:required="false"`)。

5. <application> 元素

```
<application
    android:label="@string/app_name"
    android:icon="@drawable/icon128"
    android:name=".base.App"
    android:theme="@style/AppTheme">
```

- `android:label`：應用程式的名稱，從字符串資源中提取 (`@string/app_name`)。
- `android:icon`：應用程式的圖標，引用可繪製資源 (`@drawable/icon128`)。
- `android:name`：自定義應用程式類 (`.base.App`)，它擴展了 Android 的 `Application` 類以進行應用程式範圍的邏輯。
- `android:theme`：應用程式的默認視覺主題 (`@style/AppTheme`)。

概念：`<application>` 標籤定義應用程式範圍的設置。資源如 `@string` 和 `@drawable` 存儲在 `res/` 文件夾中，促進重用和本地化。

6. 使用 `<activity>` 的活動

清單列出了幾個活動，這些活動是應用程式的 UI 屏幕：

範例 1：啟動屏幕（啟動活動）

```
<activity
    android:name=".base.SplashActivity"
    android:theme="@android:style/Theme.Holo.Light.NoActionBar.Fullscreen">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

- `android:name`：類名 (`.base.SplashActivity`)。
- `intent-filter`：將其標記為應用程式的入口點 (MAIN 操作 +LAUNCHER 類別)，因此它會出現在設備的應用程式啟動器中。
- `android:theme`：一個全屏主題，沒有操作欄。

模式：啟動活動是一個常見的起點，通常是啟動屏幕或主屏幕。

範例 2：相機活動

```
<activity  
    android:name=".deprecated.CameraActivity"  
    android:screenOrientation="landscape">
```

- `android:screenOrientation`：強制橫向模式。
- `.deprecated`：暗示這個活動可能已過時，但仍然包含。

模式：活動通常為特定用例強制方向（例如，相機應用程式在橫向模式下工作更好）。

其他活動 清單中列出了更多活動，如 `DrawActivity`、`ResultActivity`、`PhotoActivity` 等，具有相似的模式：

- 大多數是橫向模式，暗示這是一個視覺或媒體專注的應用程式。- 一些覆蓋應用程式的默認主題（例如 `Theme.Holo.Light`）。

概念：活動是 Android 應用程式 UI 的構建塊。每個 `<activity>` 標籤都在系統中註冊了一個屏幕。

此清單中的關鍵模式

1. **媒體中心設計**：相機、存儲和自動對焦的權限和功能暗示這是一個照片或繪圖應用程式（可能是識別花朵，給定套件名稱 `com.lzw.flower`）。
 2. **方向控制**：大量使用 `android:screenOrientation="landscape"` 暗示專注於視覺任務。
 3. **模塊化活動**：多個活動（`CameraActivity`、`DrawActivity`、`ResultActivity`）表明多步驟的工作流程。
 4. **資源使用**：對 `@string`、`@drawable` 和 `@style` 的引用顯示了一個乾淨、可維護的結構。
-

結論

`AndroidManifest.xml` 不僅僅是一個配置文件—它是應用程式目的和行為的窗口。在這種情況下，「Flower」似乎是一個具有相機功能、繪圖功能和網絡功能的媒體應用程式，可能用於上傳或處理圖像。通過了解其組件—權限、功能和活動—你可以看到 Android 應用程式是如何構建的以及如何設計你自己的。

想要構建類似的東西嗎？從一個明確的目的（例如花朵識別）開始，定義你的權限和功能，並規劃你的活動。清單將把它們全部綁在一起！