

# 支持不同的平台版本

尽管最新版本的 Android 经常会为您的应用提供强大的 API，但您还是应继续支持旧版 Android，直至更多设备更新到新版本。本课将介绍如何利用最新 API，同时继续为旧版 API 提供支持。

**平台版本** (<https://developer.android.google.cn/about/dashboards/index.html?hl=zh-cn>)信息中心会定期更新，以根据访问 Google Play 商店的设备数量显示搭载 Android 各版本的活跃设备分布情况。一般而言，在针对最新版本开发应用的同时，最好使**应用能够支持大约 90% 的活跃设备**。

**提示：**如需为多个 Android 版本提供最佳特性和功能，您应在应用内使用 **Android 支持库** (<https://developer.android.google.cn/tools/support-library?hl=zh-cn>)，从而**能够在旧版系统上使用一些最新的平台 API**。

## 指定最低和目标 API 级别

**AndroidManifest.xml** (<https://developer.android.google.cn/guide/topics/manifest/manifest-intro?hl=zh-cn>) 文件介绍了关于您应用的详细信息，并列出了其支持的 Android 版本。具体而言，**<uses-sdk>** (<https://developer.android.google.cn/guide/topics/manifest/uses-sdk-element?hl=zh-cn>) 元素的 minSdkVersion 和 targetSdkVersion 属性分别表示您的**应用兼容的最低 API 级别**以及**您设计和测试应用所使用的最高 API 级别**。

例如：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" ... >
    <uses-sdk android:minSdkVersion="4" android:targetSdkVersion="15" />
    ...
</manifest>
```

随着 Android 新版本的发布，某些样式和行为可能会发生变化。为了让您的应用能够充分利用这些变化并确保其适应各类用户设备的样式，您**应将 targetSdkVersion** (<https://developer.android.google.cn/guide/topics/manifest/uses-sdk-element?hl=zh-cn#target>) **值设置为与市场上最新的 Android 版本一致**。

## 在运行时检查系统版本

Android 会在 **Build** (<https://developer.android.google.cn/reference/android/os/Build?hl=zh-cn>) 常量类中为每个平台版本提供一个唯一的代码。请在您的应用内使用这些代码构建条件，以确保那些依赖更高 API 级别的代码仅在系统上提供这些 API 时才会执行。

KOTLINJAVA (#JAVA)

```
private fun setUpActionBar() {
    // Make sure we're running on Honeycomb or higher to use ActionBar APIs
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
        actionBar.setDisplayHomeAsUpEnabled(true)
    }
}
```

**注意：****解析 XML 资源时，Android 会忽略当前设备不支持的 XML 属性**。因此您可以放心地使用只有新版系统才支持的 XML 属性，无需担心旧版系统遇到该代码时会中断执行。例如，如果您设置了 targetSdkVersion="11"，默认情况下您的应用在 Android 3.0 及更高版本的系统上才会包含 **ActionBar** (<https://developer.android.google.cn/reference/android/app/ActionBar?hl=zh-cn>)。之后如需为操作栏添加菜单项，则需在您的菜单资源 XML 中设置 android:showAsAction="ifRoom"。您可以放心地在跨版本 XML 文件中执行此操作，因为旧版 Android 会直接忽略 showAsAction 属性（也就是说，您不需要在 res/menu-v11/ 中单独提供一个版本）。

# 使用平台样式和主题背景

Android 提供的用户体验主题背景可为应用赋予底层操作系统的外观和风格。这些主题背景可在清单文件中应用于您的应用。通过利用这些 **内置样式和主题背景**，您的应用可在每个**新版 Android** 系统发布时**自动沿用其最新外观和风格**。

如需为您的 Activity 赋予类似对话框的外观：

```
<activity android:theme="@android:style/Theme.Dialog">
```

如需为您的 Activity 赋予透明背景：

```
<activity android:theme="@android:style/Theme.Translucent">
```

如需应用您在 /res/values/styles.xml 中定义的自定义主题背景：

```
<activity android:theme="@style/CustomTheme">
```

如需将某个主题背景应用于您的整个应用（所有 Activity），请为 `<application>` (<https://developer.android.google.cn/guide/topics/manifest/application-element?hl=zh-cn>) 元素添加 `android:theme` 属性：

```
<application android:theme="@style/CustomTheme">
```

如需详细了解如何创建和使用主题背景，请参阅[样式和主题背景](https://developer.android.google.cn/guide/topics/ui/themes?hl=zh-cn) (<https://developer.android.google.cn/guide/topics/ui/themes?hl=zh-cn>)指南。