



支持库

本文内容

- [概览](#)
- [向后兼容性](#)
- [支持一般布局模式](#)
- [支持不同的设备类型](#)
- [一般实用程序](#)

另请参阅

- [支持库功能](#)
- [支持库设置](#)
- [支持库修订历史记录](#)

Android 支持库提供了诸多未内置于框架的功能。这些库提供向后兼容版本的新功能、框架中未包含的实用 UI 元素，以及应用可以利用的一系列实用程序。

概览

许多情况下，某项功能可能对应用开发者很有用，但是添加到 Android 框架却并不合适。例如，某个应用可能仅需要用于特定用例的某项功能，如在不同版本的 Android 系统之间顺畅切换。

为了解决这一问题，Android SDK 添加了多个库，这些库统称为 *Android 支持库*。如果应用开发者想要在教育应用中集成库功能，他们可以添加其中任意一个库。

支持库提供一系列不同的功能：

- [向后兼容](#) 版本的框架组件。
- 用于实现建议的 Android [布局模式](#) 的 UI 元素。
- 支持不同的 [设备类型](#)。
- 其他 [实用程序](#) 功能。

向后兼容性



图 1. 由于此应用使用支持库 UI 元素，即使是在对 Material Design 不提供原生支持的 Android 4.4 上运行，其界面仍符合 Material Design 原则。

支持库可以让在旧版本 Android 平台上运行的应用支持为新版本平台推出的功能。例如，应用在依赖于框架类的 5.0（API 级别 21）版本以下的 Android 系统上运行时，将无法显示 Material Design 元素，因为该版本的 Android 框架不支持 Material Design。但是，如果此应用添加了支持库的 [appcompat 库](#)，则可以访问 API 级别 21 中具有的所有功能，其中包括对 Material Design 的支持。因此，您的应用可以在多个平台版本中提供更为一致的体验。

某些情况下，类的支持库版本很大程度上取决于框架提供的功能。因此，如果应用调用其中一个支持类的方法，则支持库的行为将取决于运行应用的 Android 版本。如果框架提供必要的功能，则支持库将通过调用框架执行任务。如果应用在旧版本的 Android 上运行，且框架未显示所需的功能，则支持库自身可能会尝试提供功能或什么都不做。无论是哪一种情形，应用通常都不需要检查其在哪一版本的 Android 上运行，而是通过支持库执行检查并选择适当的行为。通常情况下，名称以 `...Compat`（如 [ActivityCompat](#)）结束的类即是如此。

而另外一些情况下，支持库类提供一个不依赖于任何框架 API 可用性的完整、独立版框架类。这些方法可以在支持的所有平台中提供一致的行为。

无论是哪一种情形，应用均无需在运行期间检查系统版本。应用可通过支持库类执行适当的系统检查，并在必要时修改其行为。

支持一般布局模式

支持库提供 Android 框架中未提供的用户界面元素。例如，Android 支持库提供其他布局类，如 [DrawerLayout](#)。这些类遵循建议的 Android 设计做法；例如，设计库以一种适合多个 Android 版本的方式遵循 Material Design 的原则。

通过使用这些支持库类，您可以避免做一些重复性工作；如果应用有特殊的用户界面要求，您可以利用现有代码，这些代码将提供用户已经熟悉的用户界面。这些元素还可以帮助您开发看起来像 Android 生态系统一部分的应用。例如，许多应用需要显示任意长的元素列表，且需要能够在列表发生变化时快速有效地重复使用这些元素；这可以是电子邮件列表、联系人列表以及音乐专辑列表，等等。这些应用可以使用支持库 [RecyclerView](#) 小部件显示列表。这既可以让应用开发者不必从头开始开发列表，又能确保用户看到一个外观和行为与其他应用中的列表类似的列表。

支持不同的设备类型

Android SDK 为 TV 和穿戴式设备等多种不同的设备类型提供库。应用可以通过相应的支持库为各种平台版本提供功能，且可以在外部屏幕、扬声器和其他目标设备上提供内容。

一般实用程序

Android 支持库提供后向兼容的实用程序功能。应用可以使用这些实用程序功能为各种 Android 系统版本提供相应的用户体验。例如，支持库的权限处理方式取决于运行应用的平台版本。如果平台支持运行时权限模式，这些方法会向用户请求相应的权限；在不支持运行时权限模式的平台版本中，这些方法将在安装时检查是否已获得相应的权限。