

# XAMARIN

Марьюшкин Антон

# Содержание презентации

01 Что такое Xamarin?

02 Xamarin.Android

03 Android Design

04 Material Theme

# ЧТО ТАКОЕ XAMARIN?

# Xamarin/Mono



**Xamarin / Mono** -это реализация .NET для запуска приложений во всех основных мобильных операционных системах.

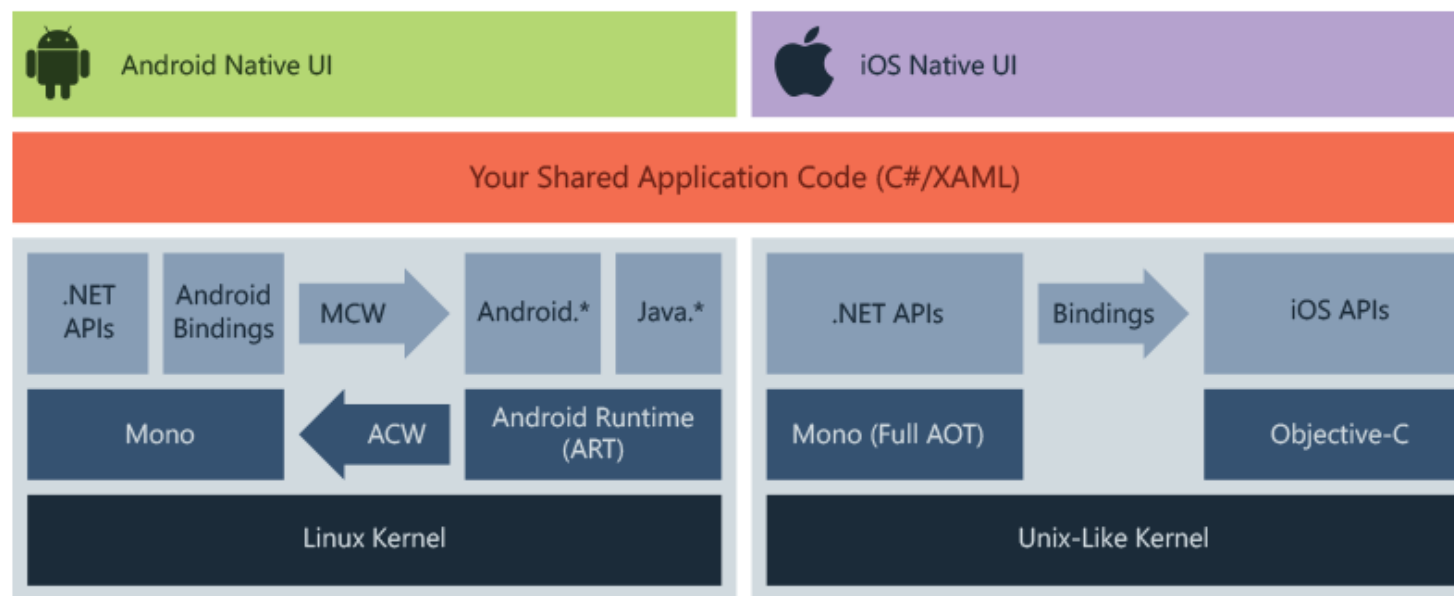
# Что такое Xamarin



[Xamarin](#) - это платформа с открытым исходным кодом, предназначенная для построения современных производительных приложений для iOS, Android и Windows с [.NET](#).

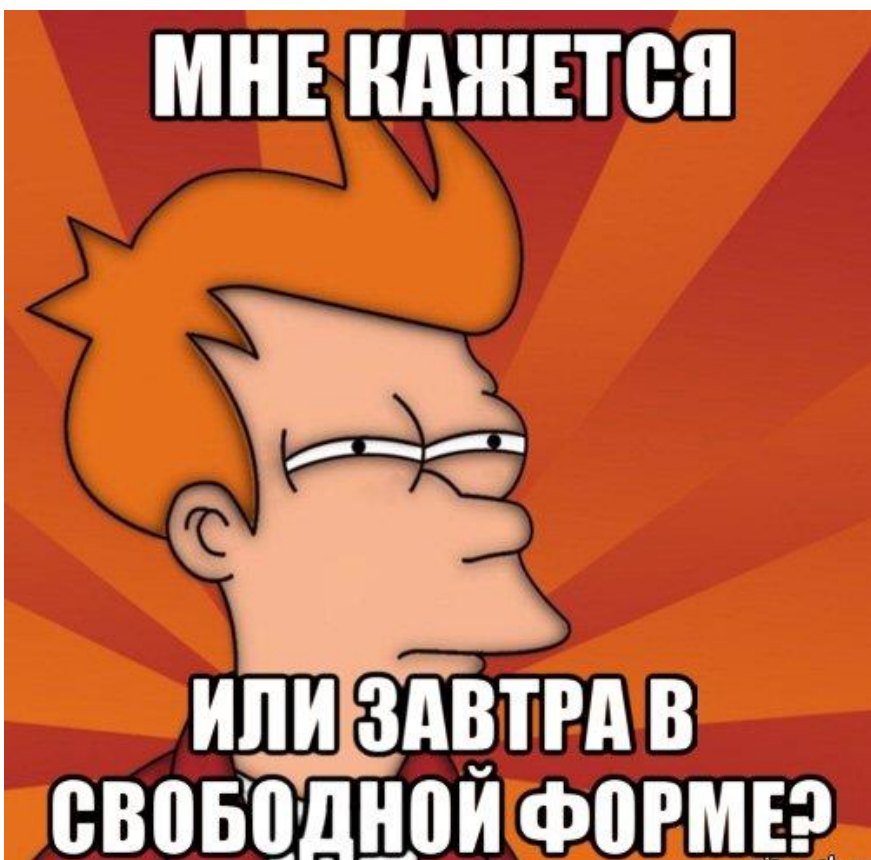
# Как работает Xamarin

С помощью [Xamarin](#) вы можете создавать собственный пользовательский интерфейс для каждой платформы и писать на языке `C#` общую бизнес-логику, которая будет использоваться на различных платформах.



В основе [Xamarin](#) лежит среда [.NET](#), которая автоматически обрабатывает такие задачи, как выделение памяти, сборка мусора и обеспечение взаимодействия с базовыми платформами.

# Xamarin.Forms



[Xamarin.Forms](#) - платформа пользовательского интерфейса с открытым кодом. С помощью Xamarin.Forms разработчики могут создавать приложения для Xamarin.iOS, Xamarin.Android и Windows на основе общей базы кода. Xamarin.Forms позволяет разработчикам создавать пользовательские интерфейсы в XAML с помощью кода программной части в C#.

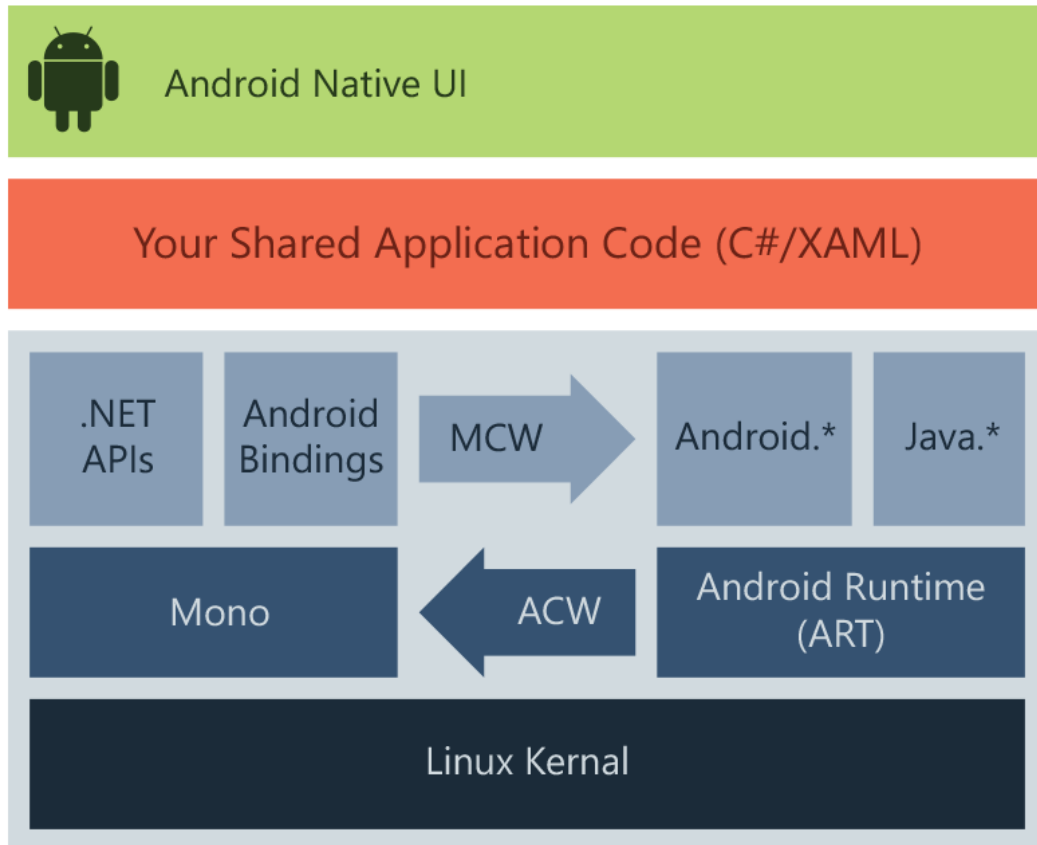
Примеры функций, предоставляемых Xamarin.Forms:

- Язык пользовательского интерфейса XAML
- Привязка данных
- Жесты
- Произведенный эффект
- Задание стиля

# Xamarin.Android



# Xamarin.Android



Компилируются из языка C# в **промежуточный язык (IL)**, который при запуске приложения претерпевает **Just-in-Time-компиляцию (JIT)** в машинную сборку.

Приложения Xamarin.Android работают в среде выполнения Mono параллельно с виртуальной машиной среды выполнения Android (ART).

Xamarin предоставляет привязки .NET к пространствам имен Android.\* и Java.\*. Среда выполнения Mono обращается к этим пространствам имен с использованием **управляемых вызываемых оболочек (MCW)** и предоставляет среде выполнения ART **вызываемые программы-оболочки Android (ACW)**.

# Параметры проекта



[Xamarin.Android](#) предоставляет три параметра проекта уровня API Android:

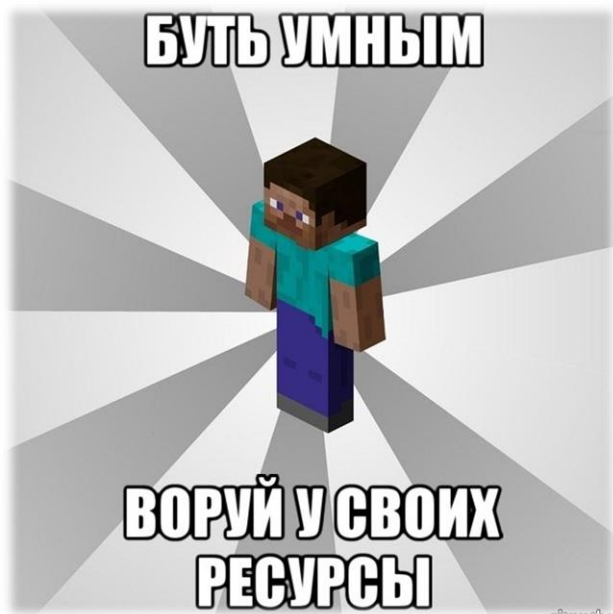
- Целевая платформа — указывает, какую платформу следует использовать при создании приложения. Этот уровень API используется во время *компиляции* [Xamarin.Android](#).
- Минимальная версия Android — указывает старую версию Android, которую вы хотите поддерживать в приложении. Этот уровень API используется во время *выполнения* Android.
- Целевая версия Android — указывает версию Android, в которую приложение предназначено для запуска. Этот уровень API используется во время *выполнения* Android.

# Службы Android

Существует четыре разных типа служб **Android**:

- **Привязанная служба** — это служба, к ней привязана какая-то другая служба (обычно действие). Привязанная служба предоставляет интерфейс, который позволяет связанному компоненту и службе взаимодействовать друг с другом. После отсутствия клиентов, привязанных к службе, Android завершит работу службы.
- **IntentService** — это специализированный подкласс Service класса, который упрощает создание и использование служб. Предназначен IntentService для обработки отдельных автономных вызовов.
- **Запущенная служба** — это служба, которая была запущена некоторыми другими компонентами Android и выполняется непрерывно в фоновом режиме, пока не будет явно указано, что служба должна остановиться. В отличие от привязанной службы, запущенная служба не привязана к ней напрямую.
- **Гибридная служба** — это служба, которая имеет характеристики *запущенной службы* и *привязанной службы*. Гибридную службу можно запустить, когда компонент привязывается к ней или может быть запущен некоторым событием. Клиентский компонент может быть привязан к гибридной службе. Гибридная служба будет работать, пока она не будет явно остановлена или пока к ней не будет привязано больше клиентов.

# Ресурсы Android



Приложение [Android](#) редко является исходным кодом. Часто существует множество других файлов, составляющих приложение: видео, изображения, шрифты и аудиофайлы только для того, чтобы назвать несколько. В совокупности эти файлы, не являющиеся исходным кодом, называются ресурсами и компилируются (вместе с исходным кодом) во время процесса сборки и упаковываются в виде APK для распространения и установки на устройства.

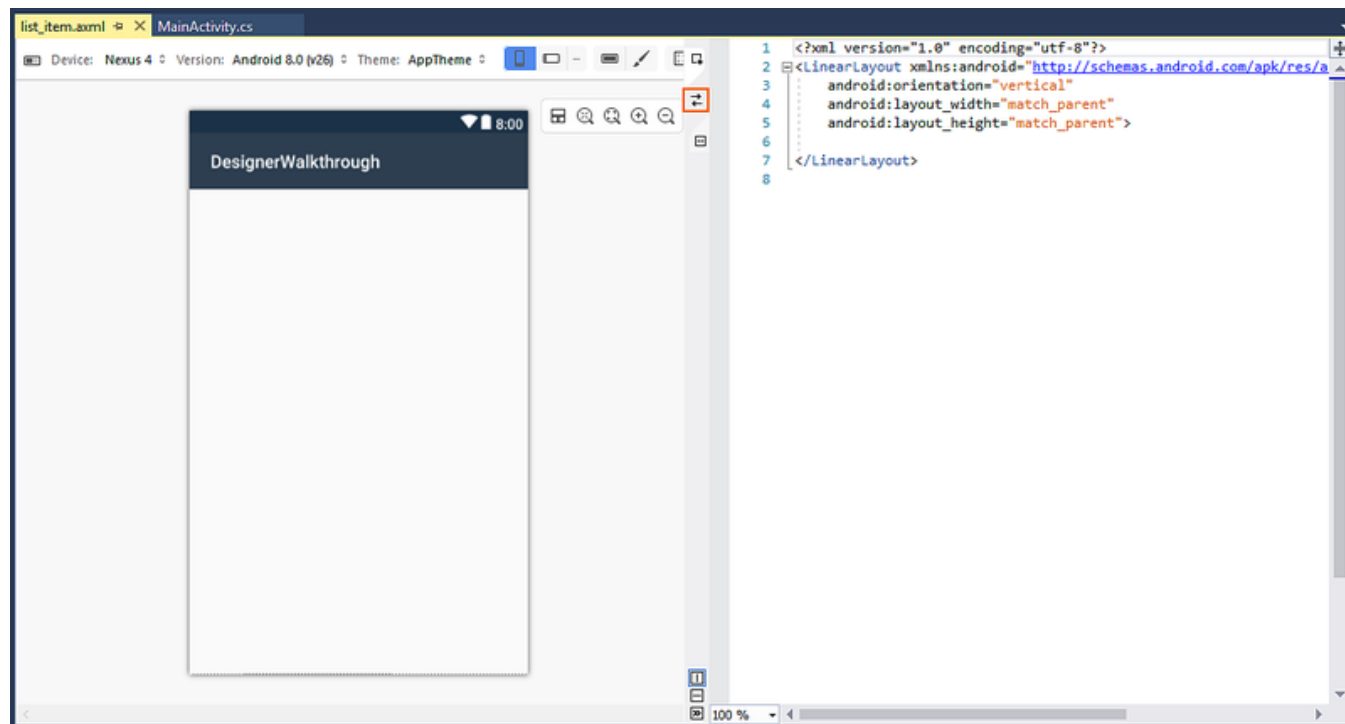
Ресурсы предлагают несколько преимуществ для приложения [Android](#):

- [Разделение кода](#) — отделяет исходный код от изображений, строк, меню, анимаций, цветов и т. д. Так как такие ресурсы могут значительно помочь при локализации.
- [Назначение нескольких устройств](#) — обеспечивает более простую поддержку различных конфигураций устройств без изменений кода.
- [Проверка во время компиляции](#) — ресурсы являются статическими и скомпилированы в приложение. Это позволяет проверять использование ресурсов во время компиляции, когда будет легко перехватывать и исправлять ошибки, в отличие от времени выполнения, когда труднее найти и дорогостоящие исправления.

# Android Design

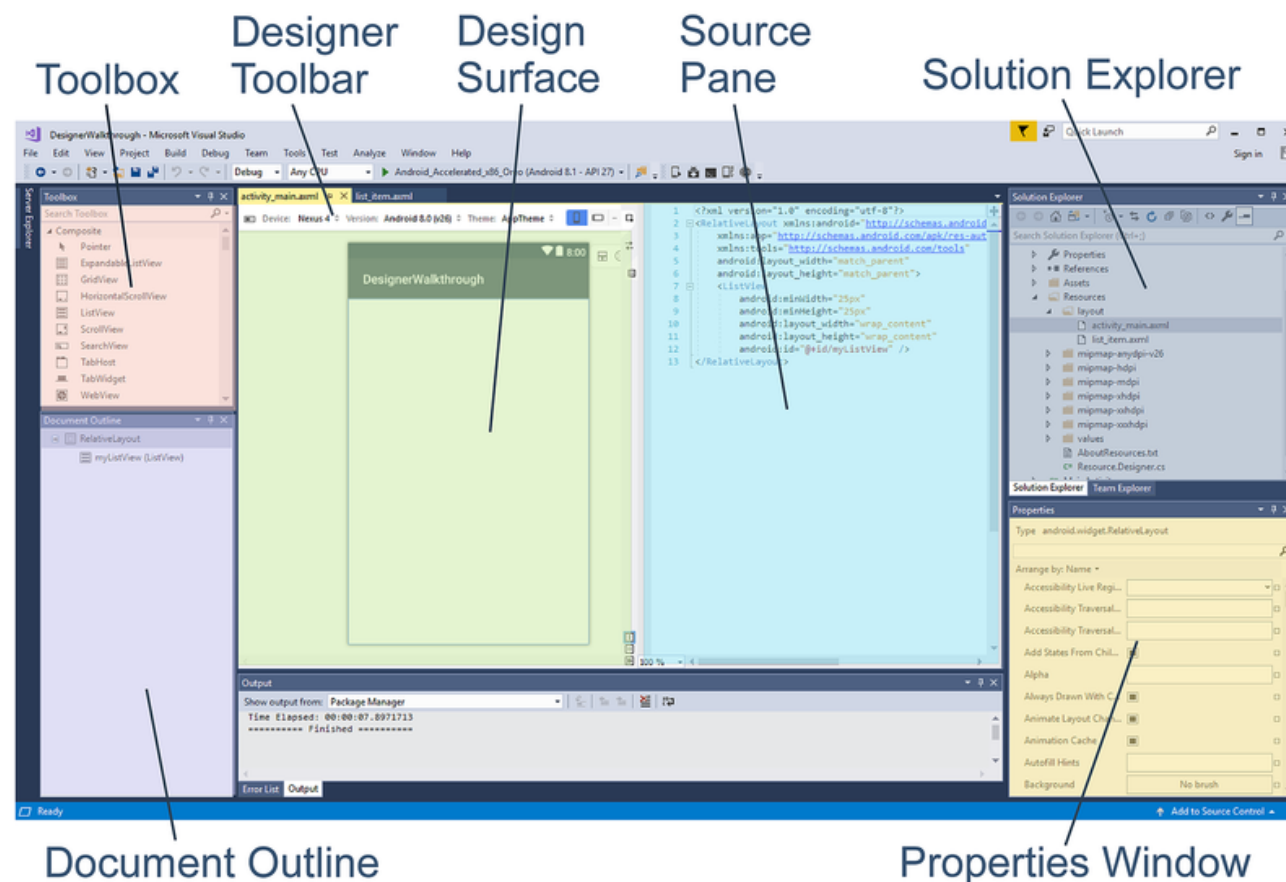
# Что такое Android Design

Конструктор [Xamarin.Android](#) позволяет разработчикам визуально создавать и изменять декларативные макеты, не требуя ручного редактирования XML-файлов. Конструктор также предоставляет откывы в режиме реального времени, которые позволяют разработчику оценивать изменения пользовательского интерфейса без необходимости повторного развертывания приложения на устройстве или в эмуляторе.



# Возможности конструктора

- **Design Surface** — упрощает визуальное построение пользовательского интерфейса, предоставляя редактируемое представление о том, как макет будет отображаться на устройстве.
- **Область источника** — предоставляет представление базового источника XML, соответствующего дизайну, представленному в области конструктора.
- **Панель инструментов конструктора** — отображает список селекторов: устройство, версия, тема, конфигурация макета и параметры панели действий. **Панель элементов** — предоставляет список мини-приложений и макетов, которые можно перетаскивать в область конструктора.
- **Окно "Свойства"** — выводит список свойств выбранного мини-приложения для просмотра и редактирования.
- **Структура документа** — отображает дерево мини-приложений, составляющих макет.



# Панель инструментов

**Альтернативный селектор макета** — позволяет выбирать из разных версий макета.

**Выбор устройства** — определяет набор квалификаторов, связанных с определенным устройством.

**Селектор версий Android** — версия Android, предназначенная для макета.

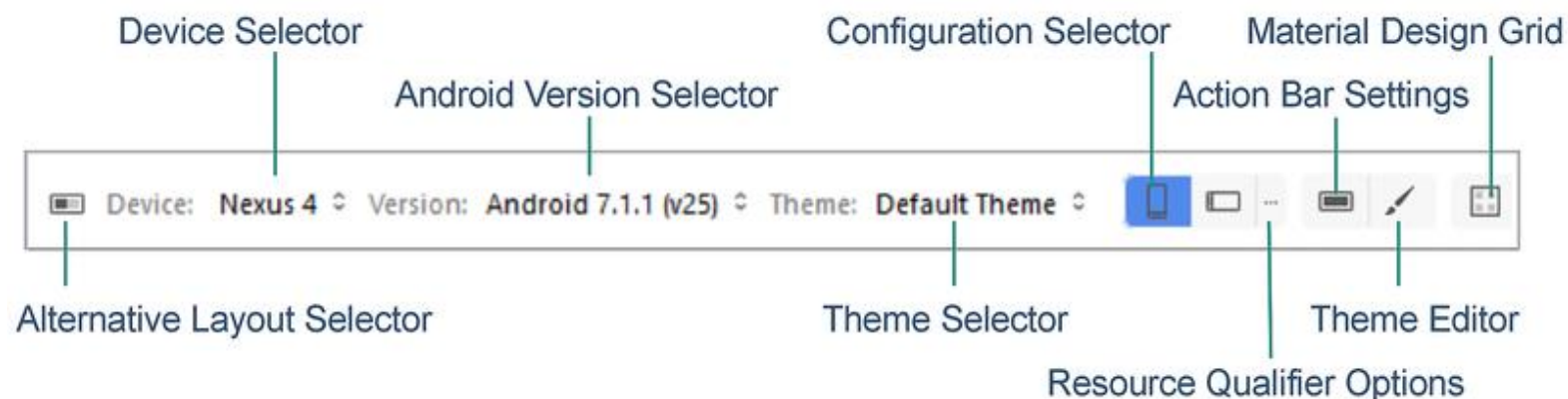
**Выбор темы** — выбирает тему пользовательского интерфейса для макета.

**Селектор конфигурации** — выбирает конфигурацию устройства.

**Параметры квалификатора ресурсов** — открывает диалоговое окно с раскрывающимися меню для выбора параметров *языка, режима пользовательского интерфейса, ночного режима и круглого экрана*.

**Редактор тем** — открывает *редактор тем*, что позволяет настраивать элементы выбранной темы.

**Сетка конструктора материалов** — включает или отключает *сетку конструктора материалов*.

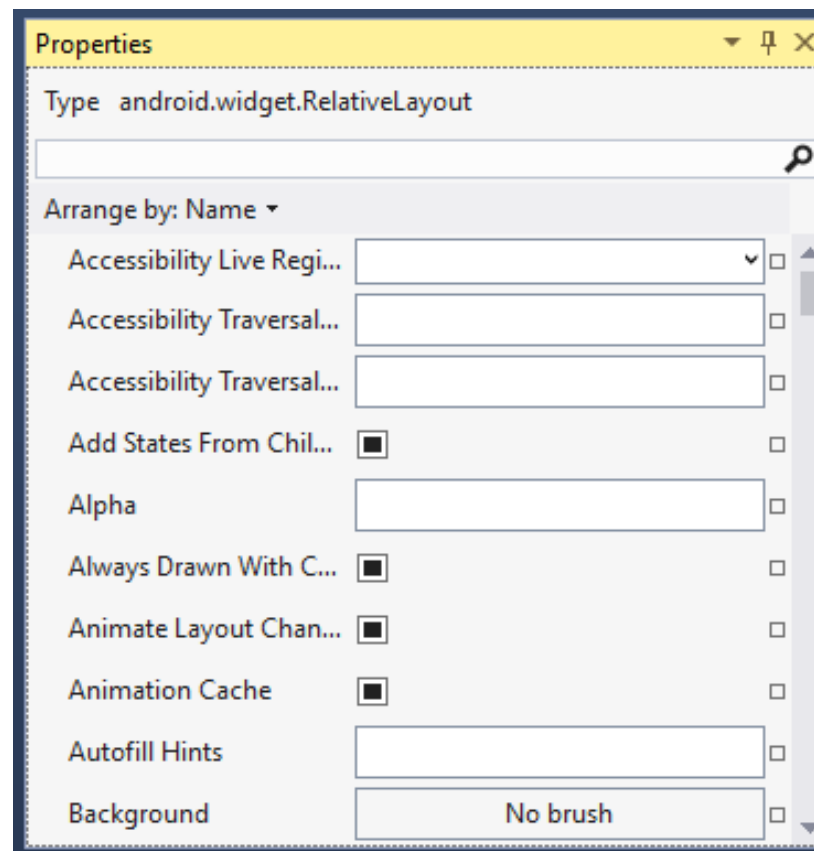




# Панель свойств

Свойства, перечисленные в области **СВОЙСТВ**, изменяются в зависимости от того, какое мини-приложение выбрано в **области конструктора**.

Свойства большинства мини-приложений будут пустыми в окне **"Свойства"**, так как их значения наследуются от выбранной темы Android. В окне **"Свойства"** будут отображаться только значения, явно заданные для выбранного мини-приложения;



# Material Theme

# Что такое Material Theme?

**Тема материала** - это стиль пользовательского интерфейса, который определяет внешний вид и вид представлений и действий, начиная с Android 5.0 (Lollipop). Материальная тема встроена в Android 5.0, поэтому она используется системным пользовательским интерфейсом, а также приложениями.



**Вы открыли тёмную тему!**

Долгожданная тёмная тема теперь в приложении ВКонтакте.

Вы сможете в любой момент сменить оформление в настройках.

# Варианты тем

- `Theme.Material` — темная версия темы материала; это разновидность по умолчанию в Android 5.0.

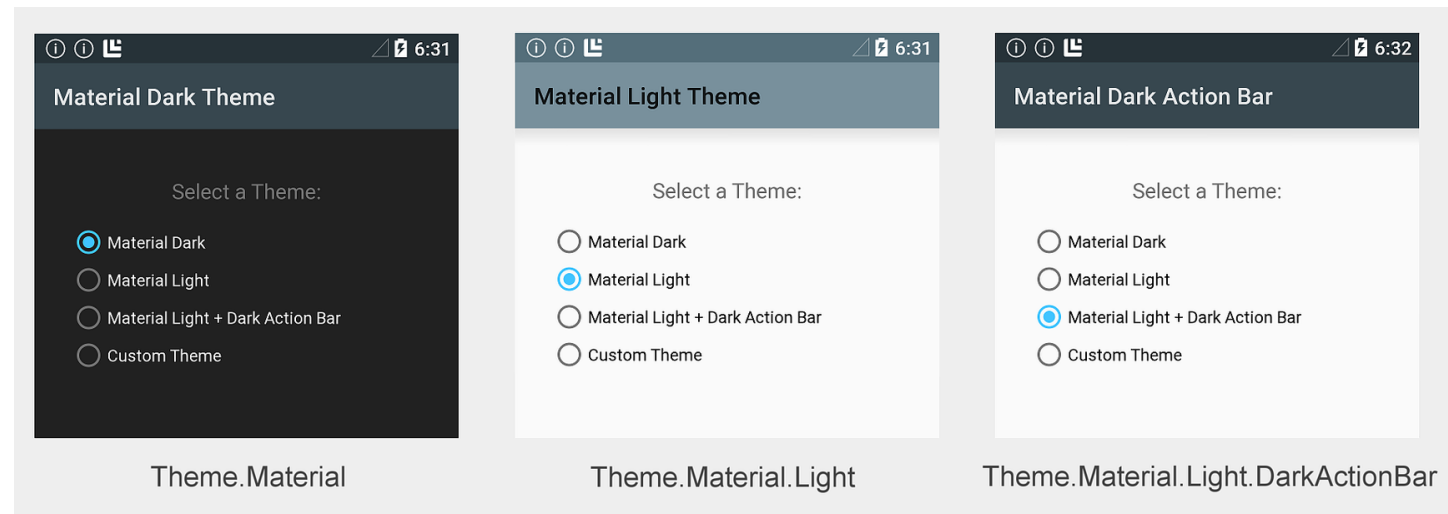
`@android:style/Theme.Material`

- `Theme.Material.Light` — облегчаемая версия темы материала.

`@android:style/Theme.Material.Light`

- `Theme.Material.Light.DarkActionBar` — светлая версия темы материала, но с темной панелью действий.

`@android:style/Theme.Material.Light.DarkActionBar`



# Требования

[Xamarin.Android](#) — Xamarin.Android 4.20 или более поздней версии необходимо установить и настроить с помощью Visual Studio или Visual Studio для Mac.

[Пакет SDK для Android](#) — Android 5.0 (API 21) или более поздней версии необходимо установить с помощью диспетчера пакетов SDK для Android.

[Java JDK 1.8](#) — JDK 1.7 можно использовать, если вы специально ориентированы на API уровня 23 и более ранних версий. JDK 1.8 доступен в Oracle.

