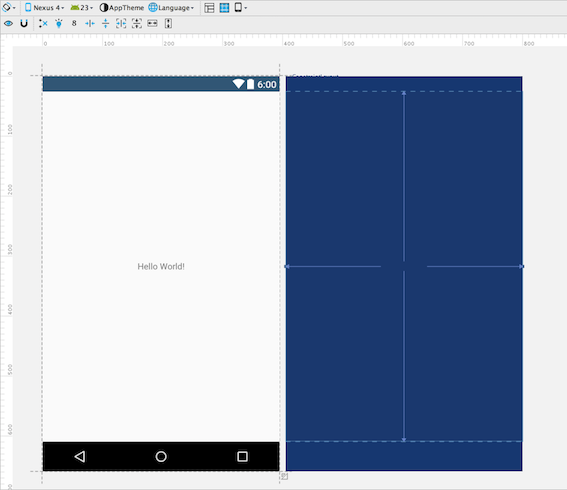
**Интро к новому ConstraintLayout в Android**

30 мая 2016 года



На прошлой неделе, также в [Гугле ввода/вывода](https://events.google.com/io2016/) конференции Google представил новый набор инструментов для разработчиков Android. Среди них-новый редактор и новый макет называется ConstraintLayout. В ConstraintLayout было сказано, что новый инструмент для разработчиков расширить возможности создания очень плоский интерфейс-иерархий для сложных макетов.

Я играл вокруг и исследовать новые вещи и я буду суммировать мои выводы в этом посте.

Чтобы быть в состоянии получить доступ к этим новым функциям, вам нужно будет под управлением Android Студия 2.2 предварительного просмотра и Java 8. Вы можете отправиться на [Канарские канал](http://tools.android.com/download/studio) , чтобы скачать ее. Будьте уверены, чтобы запустить его параллельно с стабильной версией. **Это еще в превью!**

**Редактор Макет И Ограничения Обзора.**

Новый редактор макета в Android Студия 2.2 предварительного просмотра специально построен для ConstraintLayout. Вы можете задать ограничения вручную, или автоматически ссылка в редакторе макета.

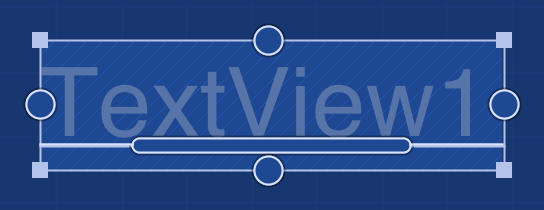
**Обзор ограничения?**

Ограничение является описание того, как должны располагаться по отношению к другим элементам в макете. Ограничение, как правило, определяется для одной или нескольких сторон, подключив целях:

* Точки привязки, или другой вид,
* Края макета
* Или невидимую линию руководства.

Поскольку каждый вид в макете определяется ассоциаций с другими представлениями в макете, его легче добиться плоского иерархии для сложных макетов.

В принципе, ConstraintLayout работает очень похоже на уровень мельче, но использует различные ручки (или якоря) для ограничения.



[(Источник)](https://codelabs.developers.google.com/codelabs/constraint-layout/index.html#3)

* Маркер изменения размера. Этот кружок-это Alt-текст видно в углах рисунке выше, и он используется для изменения его размера.
* Боковая рукоятка. Боковую ручку на Alt-текст на рисунке выше, и он используется для указания местоположения виджета. Например, используя в левой стороне ручки, чтобы всегда быть выровнен по правому другому мнению, или слева от себя ConstraintLayout.
* Базовые ручки. Базовый ручка на Alt-текст на рисунке выше. Оно использован для того чтобы выровнять текст смотреть по базовой линии текста на другой вид.

**Приступая к работе с ConstraintLayout**

**Установки**

Убедитесь, что вы используете в качестве предварительной версии 2.2, и Android поддержки репозитория версии 32 или выше, это необходимо, прежде чем вы можете использовать ConstraintLayout. Давайте начнем.

* Во-первых, вы должны добавить ограничения макета библиотека для приложения зависимостей в build.gradle файл:

**dependencies {**

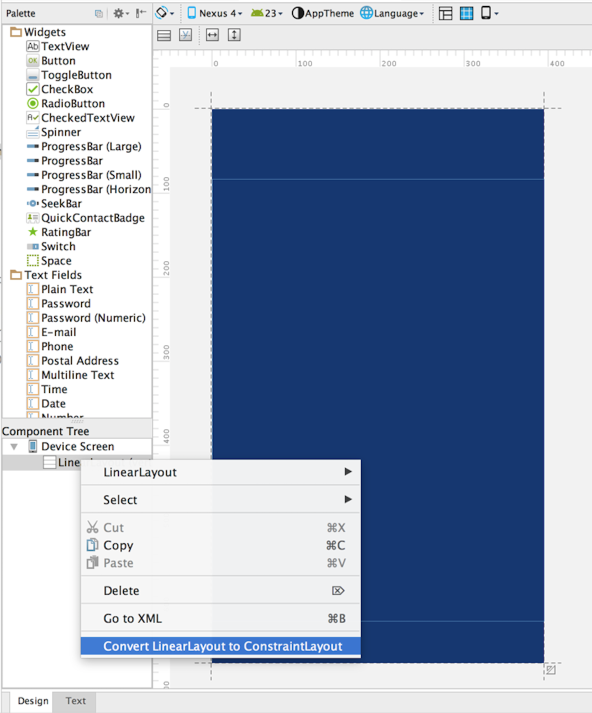
**compile 'com.android.support.constraint:constraint-layout:1.0.0-alpha1'**

**}**

* Синхронизируйте проект.

**Добавить Ограничения**

Как правило, существует два способа создания ConstraintLayout в качестве. Вы можете создать новый XML-макета и выберите корневой элемент должен быть ConstraintLayout или преобразовать существующий макет в ConstraintLayout, как показано на рисунке ниже:



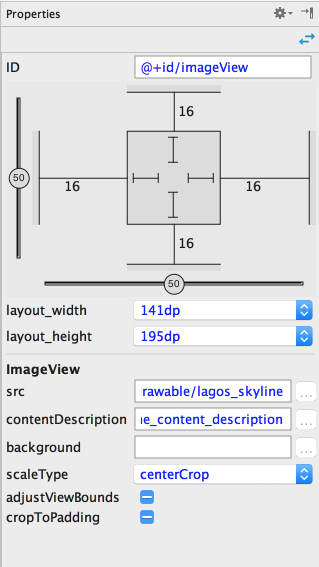
После установки ConstraintLayout, что далее следует добавить ограничения на Просмотры в рамках этого макета.

В качестве примера, перетащите ImageView на макет. Новый конструктор макета будет немедленно попросить добавить мешочки или ресурса, выберите один из предложенных вариантов и нажмите OK. Также перетащить TextView к макету.

Чтобы создать ограничение, перетащите верхнюю сторону ручку ImageView в верхней части ConstraintLayout. Вы можете также перетащить из верхней части ручки TextView до нижней рукоятке ImageView

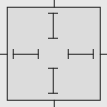
**С помощью панели инспектора**

Теперь, мы можем добавить ограничения, мы должны использовать инспектор. Это на правой стороне застройщика макет и в нем перечислены различные свойства выбранного виджета. Как правило, это выглядит так, как показано ниже:



Вы можете использовать ползунки, чтобы переместить вид в процентах по осям X и Y. Вы можете также контролировать размеры вид с инспектором области, путем изменения значения, соответствующие layout\_width и layout\_height поля.

Принимая пристальный взгляд на квадратный виджет на панели инспектора. Она содержит более определенный контроль над размерами мнения.



Есть другие способы контролировать Размер представления. Нажав на внутренних линиях на картинке выше помогает вам цикл через другие режимы.

* Фиксированный режим: Alt-текст этот режим позволяет задать ширину и высоту смотреть.
* Любого размера: Alt-текст в этом режиме изображение, чтобы заполнить все пространство, необходимое для выполнения этого ограничения. Вы можете смотреть на это как *“матч ограничения”*
* Обтекание Контента: Alt-текст это просто расширяется, чтобы заполнить содержимое представления. Например, текст или изображения

**С помощью автоматического подключения к добавить ограничения.**

Автосоединение как следует из названия, автоматически создает соединения между вид/виджеты. Она пытается создать подключение к соседнему вид/виджеты.  
Чтобы включить автосоединение, обратите внимание на Alt-текст значок на верхней панели редактора.

**Последние слова**

Этот пост не исчерпывающий перечень способов создать и использовать ConstraintLayout , но это предназначается, чтобы просто служить в качестве грунтовки. Я буду делиться как использовать вывод для создания ограничения на более поздний срок. Я надеюсь, вам проверить ссылки и дополнительная литература для более.

По состоянию на сегодняшний день, нет никакой документации ConstraintLayout , но я думаю, это в работах. Пожалуйста, обратитесь к разделу ниже для получения дополнительной информации о том, как использовать его.

**Ссылки и дополнительная литература.**

* <http://tools.android.com/tech-docs/layout-editor>
* <https://codelabs.developers.google.com/codelabs/constraint-layout/#0>
* <http://android-developers.blogspot.com.ng/2016/05/android-studio-22-preview-new-ui.html>

Если у вас есть замечания/предложения или исправления, пожалуйста, не стесняйтесь оставить комментарий ниже :)

Пожалуйста, поделитесь, если это помогло вам понять ConstraintLayout.

Ура.