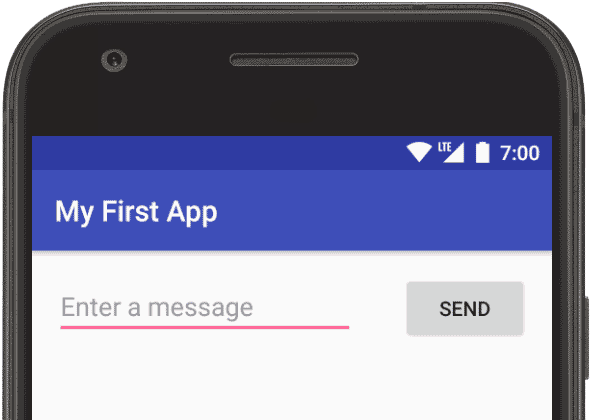
**Создание простого пользовательского интерфейса Andoid Studio**

В этом уроке рассматривается:

* как открыть Layout Editor (редактор макета)
* как добавить текстовое поле (Text Box)
* как добавить кнопку (Button)
* изменение строк в пользовательском интерфейсе
* как сделать размер текстового поля гибким.

В этом уроке вы будете использовать редактор макета Android Studio для создания макета На макете мы разместим  текстовое поле и кнопку. На следующем уроке сделаем так, что приложение будет реагировать на нажатие кнопки, посылая содержимое текстового поля другой активности (Activity).



Пользовательский интерфейс для Android-приложения построен с использованием иерархии макетов (объектов ViewGroup) и виджетов (View objects). Макеты — это невидимые контейнеры, которые определяют, как его дочерние представления (Views — вьюшки) располагаются на экране. Виджеты — это компоненты пользовательского интерфейса, такие как кнопки и текстовые поля.



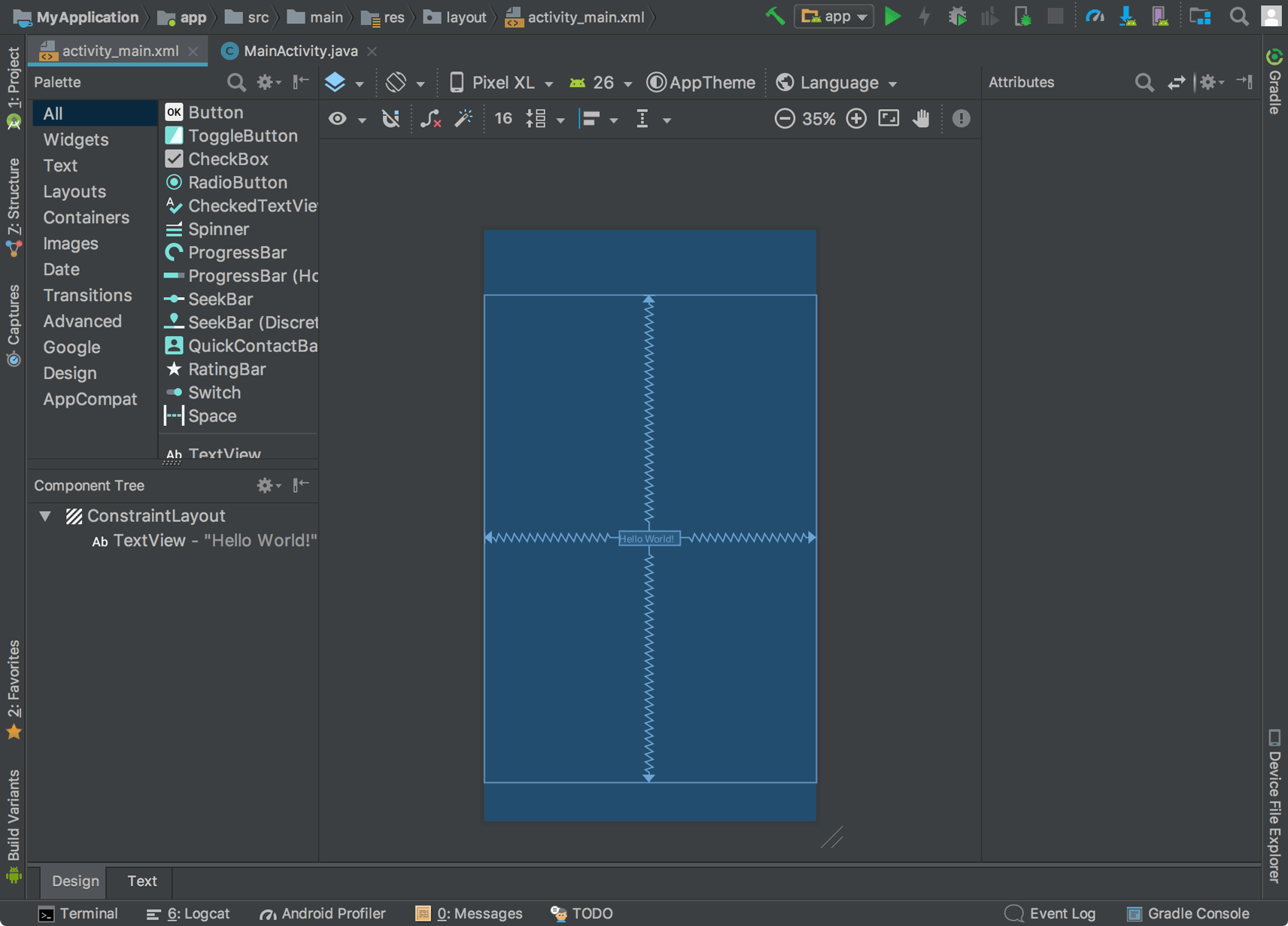
Android предоставляет словарь XML для классов ViewGroup и View, поэтому большая часть вашего пользовательского интерфейса определена в XML-файлах. Однако вместо того, чтобы научить вас писать какой-либо XML, этот урок показывает вам, как создать макет с помощью редактора макетов Android Studio, что упрощает создание макета с помощью перетаскиваемых представлений (Вьюшек).

Откройте Layout Editor (редактор макета)

*Примечание. Этот урок предполагает, что вы используете Android Studio 3.0, и вы выполнили предыдущий урок, чтобы создать свой Android-проект.*

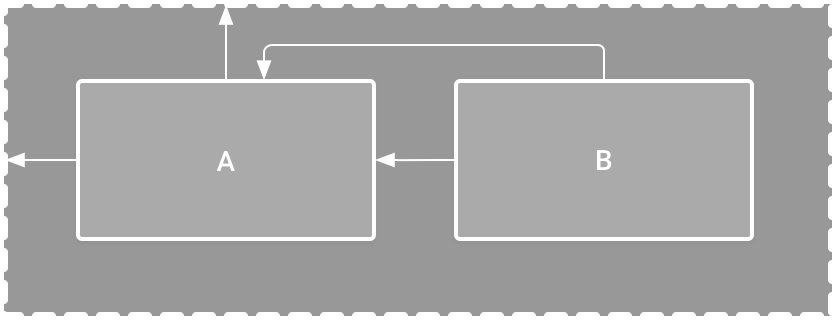
Чтобы начать работу, настройте рабочее пространство следующим образом:

1. В окне проекта Android Studio откройте **app** (приложение) **-> res -> layout -> activity\_main.xml.**
2. Чтобы редактор макета занимал больше места, скройте окно «Проект», выбрав **View > Tool Windows > Project (**«Вид» -> «Инструменты»-> «Проект») (или щелкните **Project** https://developer.android.com/studio/images/buttons/window-project.png?hl=ru («Проект») в левой части Android Studio).
3. Если ваш редактор показывает источник XML, откройте вкладку **Design (**«Дизайн») в нижней части окна.
4. Нажмите **Select Design Surface** https://developer.android.com/studio/images/buttons/layout-editor-design.png?hl=ru («Выбрать проектную поверхность») и выберите **Blueprint**.
5. Нажмите **Show** https://developer.android.com/studio/images/buttons/layout-editor-show-constraints.png?hl=ru («Показать») на панели инструментов и убедитесь, что **Show Constraints**(«Показать ограничения») отмечен.
6. Убедитесь, что Autoconnect выключен. На всплывающей подсказке на панели инструментов следует включить **Turn On Autoconnect** https://developer.android.com/studio/images/buttons/layout-editor-autoconnect-on.png?hl=ru (это значит, что он выключен).
7. Нажмите **Default Margins** https://developer.android.com/studio/images/buttons/layout-editor-margin.png?hl=ru («Поля по умолчанию») на панели инструментов и выберите**16**(вы все равно можете настроить отступы для каждого вида позже).
8. Нажмите   **Device in Editor** https://developer.android.com/studio/images/buttons/layout-editor-device.png?hl=ru («Устройство в редакторе») на панели инструментов и выберите «**Pixel XL**».  
   Теперь ваш редактор должен выглядеть так, как показано на рисунке ниже.



Окно **Component Tree** в нижней левой части показывает иерархию представлений (Вьюшек) макета. В этом случае корневой вид представляет собой ConstraintLayout, содержащий только один объект TextView.

**ConstraintLayout** — это макет, который определяет положение для каждого представления, основанное на ограничениях дочерних представлений и родительского макета. Таким образом, вы можете создавать как простые, так и сложные макеты с помощью двумерной иерархии. Это позволяет избежать излишней вложенности макетов (макет внутри макета, как показано на рисунке выше), что может увеличить время, необходимое для рисования пользовательского интерфейса.

Например, вы можете объявить следующий макет:

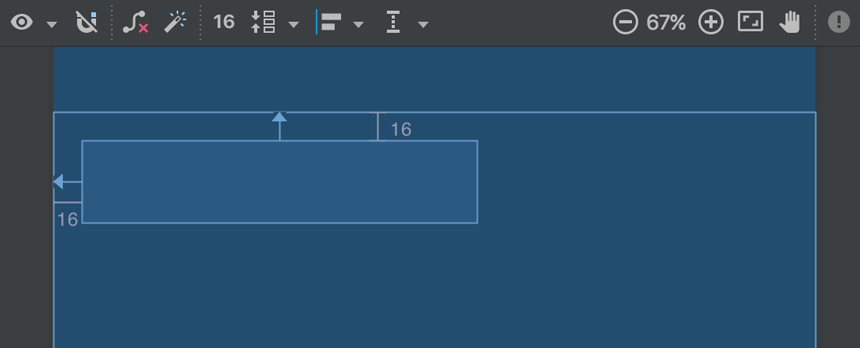
— View A располагается на 16dp ниже от верхней части родительского макета.  
— View A располагается на 16dp слева от родительского макета.  
— View В располагается на 16dp справа от View A.  
— View В выровнен по верхней границе View A.

В следующих разделах вы создадите макет, подобный этому.

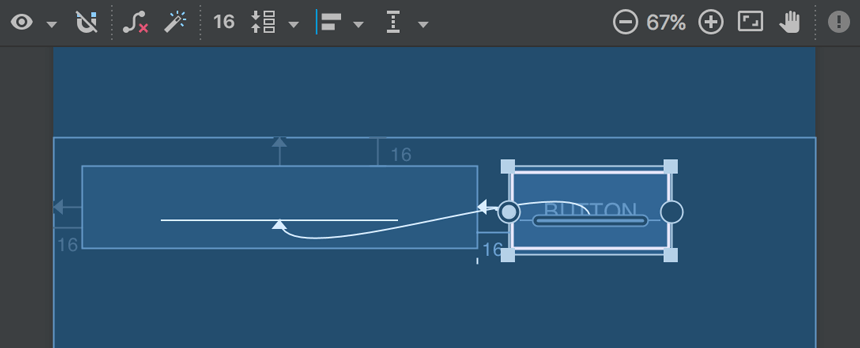
Добавляем текстовое поле (Text Box)

1. Сначала удалим то, что уже находится в макете. Поэтому нажмите «**TextView**» в окне **Component Tree**(«Дерево компонентов»), а затем нажмите **Delete** («Удалить»).
2. В окне **Palette** («Палитра») слева нажмите **Text** («Текст») в левой панели, а затем перетащите **Plain Text**(«Обычный текст») в редактор проекта и поместите его в верхней части макета. Это виджет **[EditText](https://developer.android.com/reference/android/widget/EditText.html?hl=ru" \t "_blank)**, который принимает обычный ввод текста.
3. Щелкните по представлению в редакторе проекта. Теперь вы можете увидеть границы изменения размера на каждом углу (квадраты) и привязки ограничений на каждой стороне (круги).  Для лучшего управления вам может потребоваться увеличить масштаб редактора, используя кнопки на панели инструментов.
4. Нажмите и удерживайте якорь на верхней стороне, а затем перетащите его вверх, пока он не защелкнется до верхней части макета и не отпустит. Это ограничение — оно указывает, что представление должно быть 16dp из верхней части макета (поскольку вы устанавливаете значения по умолчанию 16dp).
5. Аналогичным образом создайте ограничение с левой стороны вида в левой части макета.

Результат должен выглядеть как скриншот на рисунке.



Добавляем кнопку

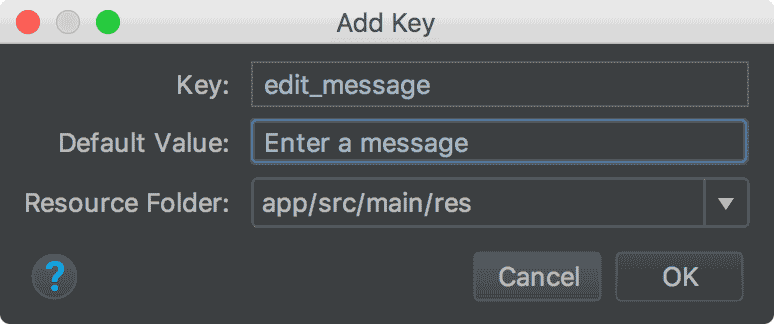
1. В окне **Palette**(«Палитра») нажмите **Widgets**(«Виджеты») на левой панели, а затем перетащите **Button**(«Кнопка») в редактор проекта и поместите ее рядом с правой границей.
2. Создайте ограничение с левой стороны кнопки в правой части текстового поля.  
   Чтобы ограничить представления в горизонтальном направлении, вам необходимо выровнять две базовые линии (линии под текстами) вьюшек. Итак, нажмите кнопку, а затем нажмите  **Edit Baseline** https://developer.android.com/studio/images/buttons/layout-editor-action-baseline.png?hl=ru («Изменить базовую линию»), которая появляется в редакторе дизайна непосредственно под выбранной вьюшкой. Внутри кнопки появляется базовый якорь. Нажмите и удерживайте этот якорь, а затем перетащите его в базовый якорь, который появляется в текстовом поле.  
   Результат должен выглядеть как скриншот на рисунке .

Примечание. Вы также можете создать горизонтальное выравнивание с использованием верхнего или нижнего края, но это не совсем верное выравнивание.

Измените строки пользовательского интерфейса

Чтобы посмотреть как же будет выглядеть пользовательский интерфейс, нажмите  **Select Design Surface** https://developer.android.com/studio/images/buttons/layout-editor-design.png?hl=ru  («Выбрать отображение поверхности») на панели инструментов и выберите  **Design**(«Дизайн»). Обратите внимание, что ввод текста предварительно заполнен словом«Name», а кнопка обозначается как «Button». Итак, теперь узнаем как изменить эти строки.

1. Откройте окно **Project** и откройте **app -> res -> values -> strings.xml**.  
   Это [файл строковых ресурсов](https://developer.android.com/guide/topics/resources/string-resource.html?hl=ru), где вы должны указать все свои строки пользовательского интерфейса. Этот файл позволяет управлять всеми строками UI в одном месте, что упрощает поиск, обновление и локализацию (по сравнению с строками жесткого кодирования в макете или коде приложения).
2. Нажмите **Open editor**(«Открыть редактор») в верхней части окна редактора. Это открывает [Translations Editor](https://developer.android.com/studio/write/translations-editor.html?hl=ru) (редактор переводов), который предоставляет простой интерфейс для добавления и редактирования строк по умолчанию и помогает сохранить все переведенные строки.
3. Нажмите **Add Key** https://developer.android.com/studio/images/buttons/add-sign-green-icon.png?hl=ru («Добавить ключ»), чтобы создать новую строку в качестве «текста подсказки» («hint text») для текстового поля.
   1. Введите «edit\_message» для имени ключа.
   2. Введите фразу «Введите сообщение» для значения.
   3. Нажмите «ОК».



4. Добавьте еще один ключ с именем «button\_send» со значением «Отправить».

Теперь вы можете установить строки для соответствующих вьюшек. Поэтому вернитесь к файлу макета, нажав на Activity\_main.xml на панели вкладок и добавьте строки следующим образом:

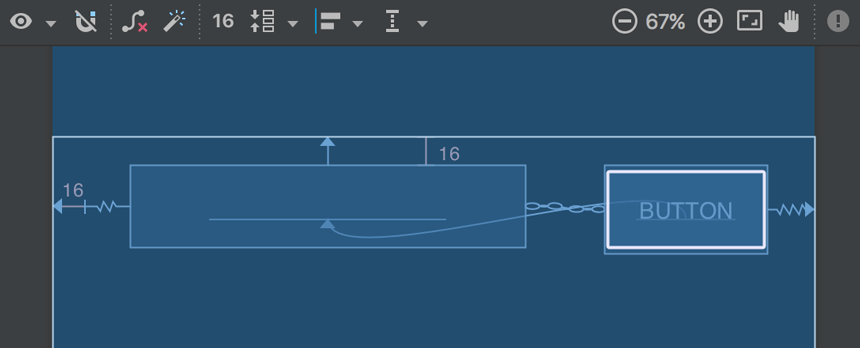
1. Щелкните текстовое поле в макете и, если окно атрибутов уже не видно справа, нажмите **Attributes**(«Атрибуты») на правой боковой панели.
2. Найдите свойство **text** текста (в настоящее время установлено значение «Имя») и удалите значение.
3. Найдите свойство **hint** подсказки и нажмите **Pick a Resource** https://developer.android.com/studio/images/buttons/pick-resource.png?hl=ru («Выбрать ресурс») справа от текстового поля. В появившемся диалоговом окне дважды щелкните по элементу edit\_message из списка.
4. Теперь нажмите кнопку в макете, найдите свойство **text** текста, нажмите  **Pick a Resource** https://developer.android.com/studio/images/buttons/pick-resource.png?hl=ru («Выбрать ресурс»), а затем выберите button\_send.

Измените строки пользовательского интерфейса

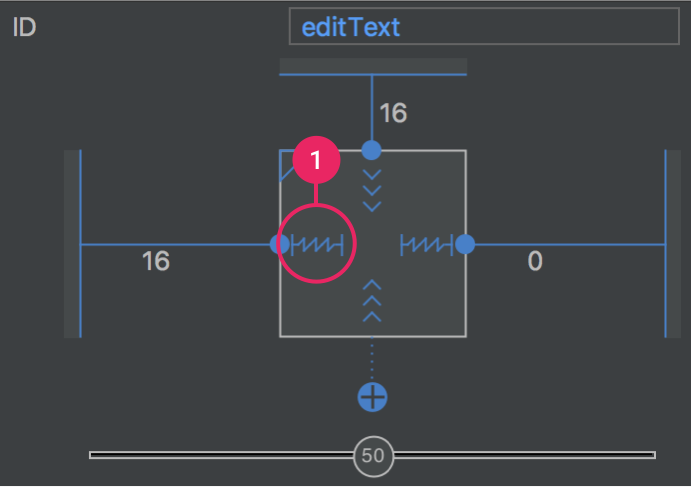
Чтобы создать макет, отвечающий различным размерам экрана, теперь необходимо растянуть текстовое поле, чтобы заполнить все оставшееся горизонтальное пространство (после учета кнопки и полей).

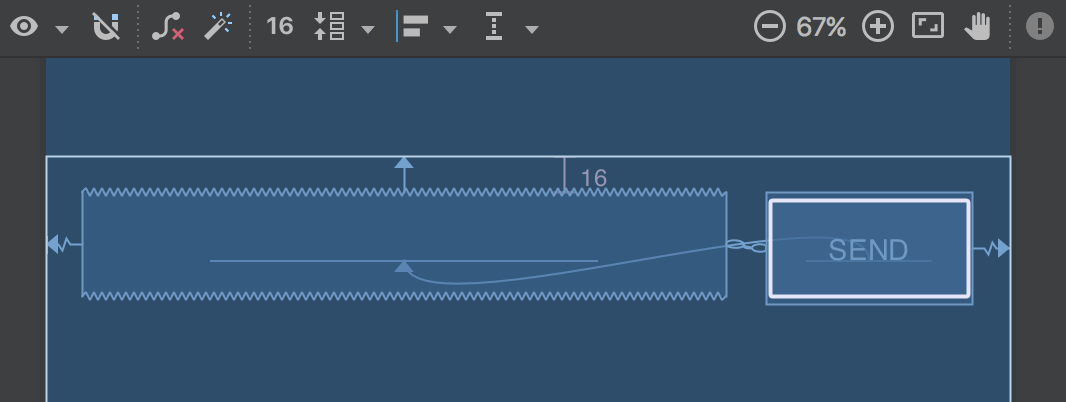
Прежде чем продолжить, нажмите **Show** https://developer.android.com/studio/images/buttons/layout-editor-design.png?hl=ru («Показать») на панели инструментов и выберите «**Blueprint**».

1. Выберите обе вьюшки (нажмите на одну и, удерживая Shift, выберите вторую), а затем щелкните правой кнопкой мыши или выберите **Chain -> Create Horizontal Chain**(«Цепочка»-> «Создать горизонтальную цепочку»). Цепь представляет собой двунаправленное ограничение между двумя или более вьюшками, которое позволяет вам выложить цепочку в один ряд.



2. Выберите кнопку (button) и откройте окно **Attributes** («Атрибуты»). Используя привязку в верхней части окна «Атрибуты», установите для правого поля значение 16.

3. Теперь щелкните текстовое поле, чтобы просмотреть его атрибуты. Дважды щелкните индикатор ширины, чтобы установить значение **Match Constraints**(«Сопоставление ограничений»), как указано выводом 1 на рисунке.  
**«Match constraints»** означают, что ширина расширяется, чтобы соответствовать определению горизонтальных ограничений и полей. Поэтому текстовое поле растягивается, чтобы заполнить горизонтальное пространство (после учета кнопки и всех полей).  
Теперь макет сделан и должен выглядеть так:



Если кажется, что ваш макет не получился так, как ожидалось, посмотрите ниже, чтобы увидеть, как должен выглядеть ваш XML и сравнить его с тем, что вы видите на вкладке «Текст». (Если ваши атрибуты отображаются в другом порядке, все в порядке.)

См. Окончательный формат XML

xml version="1.0" encoding="utf-8"?>

<android.support.constraint.constraintlayout< span="">

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context="com.example.myfirstapp.MainActivity">

    <edittext< span="">

        android:id="@+id/editText"

        android:layout\_width="0dp"

        android:layout\_height="wrap\_content"

        android:layout\_marginStart="16dp"

        android:layout\_marginTop="16dp"

        android:ems="10"

        android:hint="@string/edit\_message"

        android:inputType="textPersonName"

        app:layout\_constraintEnd\_toStartOf="@+id/button"

        app:layout\_constraintHorizontal\_bias="0.5"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent" />

    <button< span="">

        android:id="@+id/button"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_marginEnd="16dp"

        android:layout\_marginStart="16dp"

        android:text="@string/button\_send"

      app:layout\_constraintBaseline\_toBaselineOf="@+id/editText"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.5"

        app:layout\_constraintStart\_toEndOf="@+id/editText" />

</button<></edittext<></android.support.constraint.constraintlayout<>

Для получения дополнительной информации о цепочках и всех других вещах, которые вы можете сделать с помощью ConstraintLayout, прочитайте «Постройте адаптивный пользовательский интерфейс с помощью ConstraintLayout».

Запустить приложение

Если ваше приложение уже установлено на устройстве из предыдущего урока, просто нажмите **Apply Changes** https://developer.android.com/studio/images/buttons/toolbar-apply-changes.png?hl=ru («Применить изменения») на панели инструментов, чтобы обновить приложение с помощью нового макета. Или нажмите  **Run** https://developer.android.com/studio/images/buttons/toolbar-run.png?hl=ru («Запустить»), чтобы установить и запустить приложение.

Кнопка все еще ничего не делает. Переходите к следующему уроку.