

Лабораторная работа №8. Создание виджета

Задание: создать собственный виджет с настройками. Например, виджет, который открывает какой-то сайт, адрес которого можно поменять в настройках.

Создайте новый проект без Activity, то есть при создании необходимо выбрать шаблон No Activity вместо привычного Empty Views Activity. Теперь необходимо создать activity для виджета. Для этого нажмите правой кнопкой мыши по папке app, затем выберите New -> Widget -> App Widget. Здесь задайте необходимые параметры для виджета. Можно оставить все пункты по умолчанию, однако нужно поставить галочку в пункте Configuration Screen, чтобы в проект добавилась activity для настройки виджета.

Перейдите в файл res/xml/new_app_widget_info.xml и в параметре android:updatePeriodMillis смените значение «86400000» на «0», так как обновлять данный виджет не нужно. Затем перейдите в файл manifests/AndroidManifest.xml и перед тэгом application добавьте приложению разрешение пользоваться интернетом:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Теперь перейдите в файл NewAppWidget.java, в методе updateAppWidget удалите весь код и напишите следующий:

Java:

```
Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse("https://google.com"));
PendingIntent pending= PendingIntent.getActivity(context, 0, intent, PendingIntent.FLAG_IMMUTABLE);
RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.new_app_widget);
views.setOnClickListener(R.id.appwidget_text, pending);
appWidgetManager.updateAppWidget(appWidgetId, views);
```

Kotlin:

```
val intent = Intent(Intent.ACTION_VIEW, Uri.parse(R.id.appwidget_text.toString()))
val pending = PendingIntent.getActivity(context, 0, intent,
PendingIntent.FLAG_IMMUTABLE)
val views = RemoteViews(context.packageName, R.layout.new_app_widget)
views.setOnClickListener(R.id.appwidget_text, pending)
appWidgetManager.updateAppWidget(appWidgetId, views)
```

Здесь устанавливается, что при нажатии на виджет необходимо открыть браузер и перейти по ссылке google.com.

Теперь необходимо найти и удалить в этом классе метод onDelete.

Затем перейдите в файл NewAppWidgetConfigureActivity.java, удалите весь код в этом классе и добавьте следующие строчки:

Java:

```
private NewAppWidgetConfigureActivity context;  
private int widgetID;
```

Kotlin:

```
private val context: NewAppWidgetConfigureActivity? = null  
private val widgetID = 0
```

Строчкой ниже начните писать **protected** и в появившемся списке выберите **protected void** **onCreate** (**savedInstanceState**), метод **onCreate** создастся автоматически. Далее в методе **onCreate** необходимо написать:

Java:

```
setContentView(R.layout.new_app_widget_configure);  
setResult(RESULT_CANCELED);  
context = this;  
Bundle extras = getIntent().getExtras();
```

Kotlin (показано начало метода **onCreate**, они бывают разные, этот работает):

```
public override fun onCreate(icicle: Bundle?) {  
  
    super.onCreate(icicle)  
  
    setContentView(R.layout.new_app_widget_configure)  
    setResult(RESULT_CANCELED)  
    context = this  
    val extras = intent.extras
```

Затем добавляем следующее условие:

Java:

```
if (extras != null) {  
    widgetID = extras.getInt(AppWidgetManager.EXTRA_APPWIDGET_ID,  
    AppWidgetManager.INVALID_APPWIDGET_ID);  
    final AppWidgetManager widgetManager = AppWidgetManager.getInstance(context);  
    final RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.new_app_widget);  
    final EditText editText = (EditText) findViewById(R.id.appwidget_text);  
    Button button = (Button) findViewById(R.id.add_button);  
}
```

Kotlin:

```
if (extras != null) {  
    widgetID = extras.getInt(  
        AppWidgetManager.EXTRA_APPWIDGET_ID,  
        AppWidgetManager.INVALID_APPWIDGET_ID  
    )  
    val widgetManager = AppWidgetManager.getInstance(context)  
    val views = RemoteViews(context!!.packageName, R.layout.new_app_widget)  
    val editText = findViewById<View>(R.id.appwidget_text) as EditText  
    val button = findViewById<View>(R.id.add_button) as Button  
}
```

Далее в условии задайте listener и onClick для объявленной кнопки и в методе onClick напишите следующее:

Java:

```
Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse(editText.getText().toString()));
PendingIntent pending= PendingIntent.getActivity(context, 0, intent, PendingIntent.FLAG_IMMUTABLE);
views.setOnClickListener(PendingIntent.getBroadcast(context, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT));
widgetManager.updateAppWidget(widgetID, views);
Intent resultValue = new Intent();
resultValue.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, widgetID);
setResult(RESULT_OK, resultValue);
finish();
```

Kotlin:

```
val intent = Intent(Intent.ACTION_VIEW, Uri.parse(editText.text.toString()))
val pending =
    PendingIntent.getActivity(context, 0, intent, PendingIntent.FLAG_IMMUTABLE)
views.setOnClickListener(PendingIntent.getBroadcast(context, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT))
widgetManager.updateAppWidget(widgetID, views)
val resultValue = Intent()
resultValue.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, widgetID)
setResult(RESULT_OK, resultValue)
finish()
```

Когда виджет добавляется на экран смартфона, появляется страница с настройками, где необходимо прописать желаемый адрес, начиная с «<http://>», и после нажатия на кнопку виджет добавится на экран смартфона и будет работать только с заданной ссылкой.

Для того, чтобы изменить внешний вид виджета или его окна конфигурации/настроек, перейдите в файлы res -> layout -> new_app_widget.xml и new_app_widget_configure.xml соответственно. Например, в последнем файле можно добавить свои ФИО и поместить текстовое поле для адреса и кнопку по центру окна, изменить LinearLayout на RelativeLayout, в текстовом поле можно написать «<http://>», чтобы пользователю не нужно было набирать это вручную; а в первом – поменять слово EXAMPLE на другое, изменить цвет виджета, его размеры в файле res/xml/new_app_widget_info.xml и т.д.

Запустить проект не получится – в нём нет activity. Поэтому постройте проект (Build -> Build App Bundle(s)/APK(s) -> APK(s)) и скопируйте и установите app-debug.apk файл на ваше устройство (справа внизу нажмите locate, и откроется папка с только что построенным apk файлом виджета). Затем скопируйте и установите виджет в соответствии с правилами вашей версии ОС Android на ваше мобильное устройство – виджет будет отображаться среди других виджетов, его надо найти и перетащить на один из рабочих столов, при этом откроется сконфигурированное в коде окно настройки, где надо будет ввести адрес и нажать кнопку Add Widget.

Что касается эмулятора, то сначала его надо запустить. Затем, чтобы скопировать готовый apk файл на эмулятор в Android Studio, нужно выбрать пункт главного меню View -> Tool Windows -> Device Explorer. В открывшемся окне появится структура каталогов и файлов эмулятора. Откройте папку sdcard -> Download. Нажмите правой кнопкой мыши по этой папке, выберите Upload, найдите только что созданный apk файл виджета (обычно он находится в папке имя_проекта\app\build\outputs\apk\debug\) и нажмите кнопку OK, см. рисунок 32. После этого (для старых эмуляторов типа ОС Android 4.1.2) перейдите в окно эмулятора, нажмите на кнопку Все программы и перейдите на закладку WIDGETS, см. рисунки 33 и 34 (приведена инструкция и показан интерфейс для ОС Android 4.1.2) – вы должны увидеть виджет (внизу у него будет написано имя_проекта, например – 8th2try, рисунок 35; при этом

перезагружать эмулятор не надо). Для последних версий ОС Android на эмуляторе надо после копирования файла зайти в приложение Files - проведите по экрану снизу вверх, откроются приложения, найдите Files и запустите его – сразу будет показана папка Downloads с файлом app-debug.apk. Нужно нажать на этот файл и установить его на устройство (возможно, его надо будет отсканировать перед установкой). Затем надо вернуться на рабочий стол, и нажать и держать нажатой кнопку мыши на свободном от иконок месте стола, после чего появится окно добавления виджета, см. рисунок 38. Надо нажать на Widgets, после чего найти свой виджет и перетянуть его на свободное место рабочего стола.

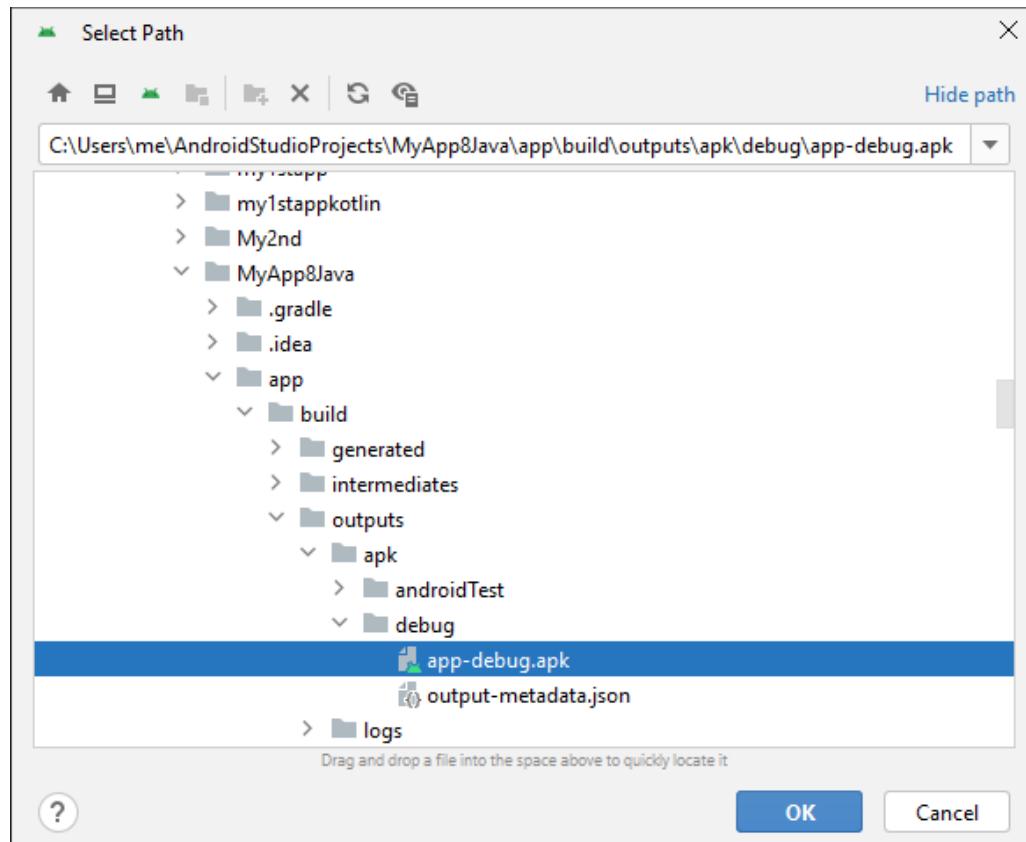


Рисунок 32 – Копирование файла виджета из папки проекта

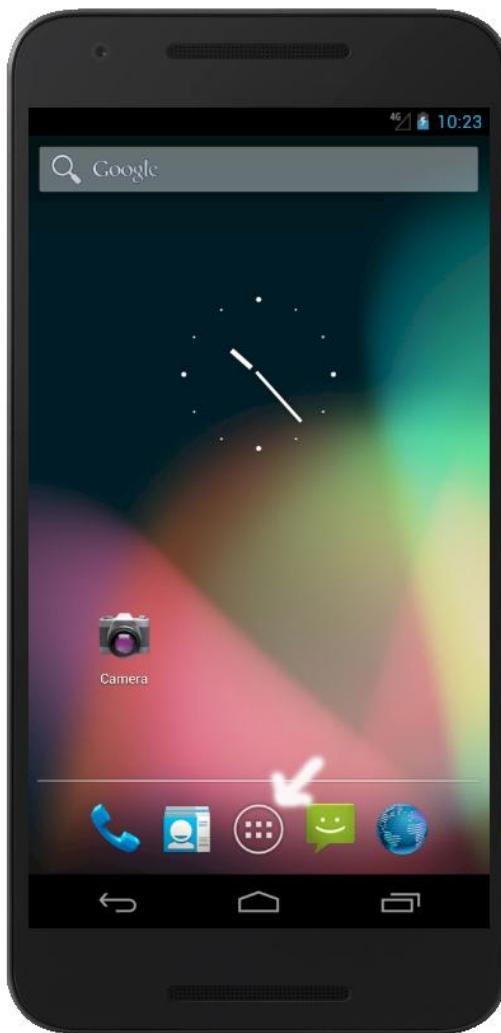


Рисунок 33 – Все программы



Рисунок 34 – Установленные приложения

Перетащите виджет на один из экранов и в настройках введите любой адрес, который обязательно должен начинаться с «<http://>! Затем нажмите кнопку Add Widget и кликните по виджету, после чего должен запуститься стандартный браузер с заданной в настройках ссылкой. Примеры страницы настройки и виджета представлены на рисунках 36 и 37 (старая версия Android), а также 39 и 40 (новая версия Android).

Иногда браузер google chrome (`com.android.chrome`) может глючить, завершая работу эмулятора после запуска себя, и мы не можем проверить, работает наш виджет, или нет. Это происходит из-за того, что выбирается неверный рендерер (`OpenGL` вместо `DirectX`), с которым `google chrome` работать не может. Чтобы избежать такого поведения браузера, нужно перейти в настройки эмулятора (три точки внизу), затем выбрать `Settings`, внутри них перейти на вкладку `Advanced` и поменять `OpenGL ES renderer` на `ANGLE (D3D11)` вместо значения по умолчанию, и затем перезагрузить эмулятор (`cold boot`). При этом `OpenGL ES API level` менять не надо, пусть остаётся `Autoselect`.

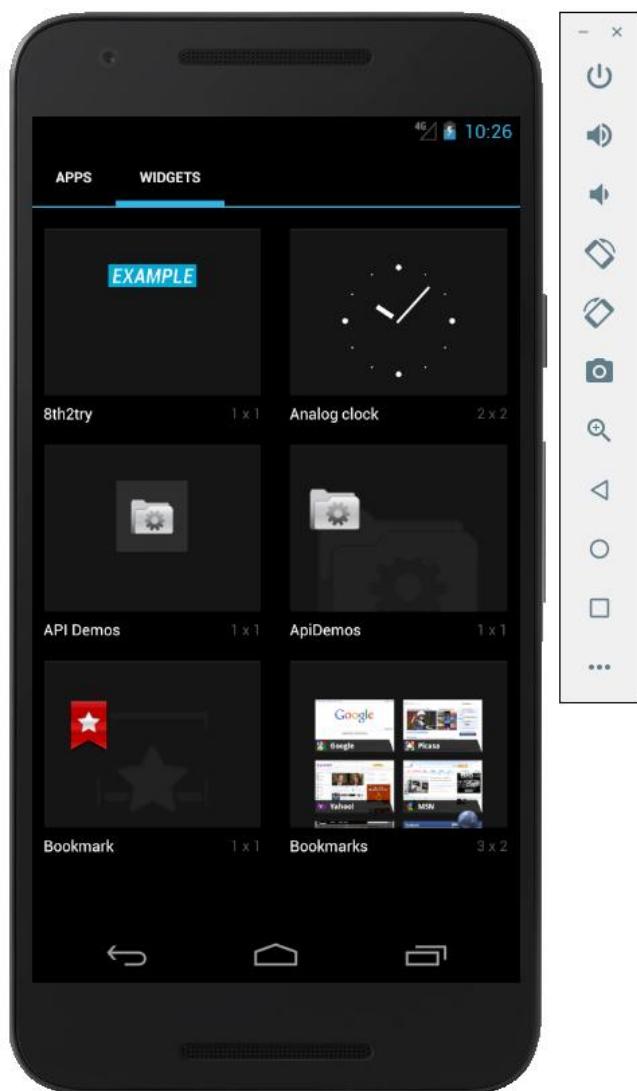


Рисунок 35 – Скопированный виджет 8th2try

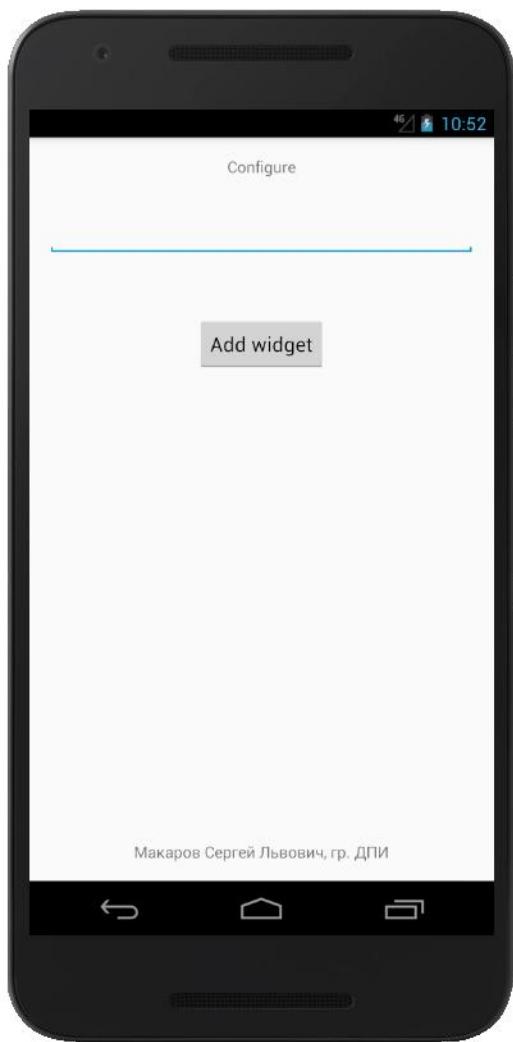


Рисунок 36 – Окно настройки виджета

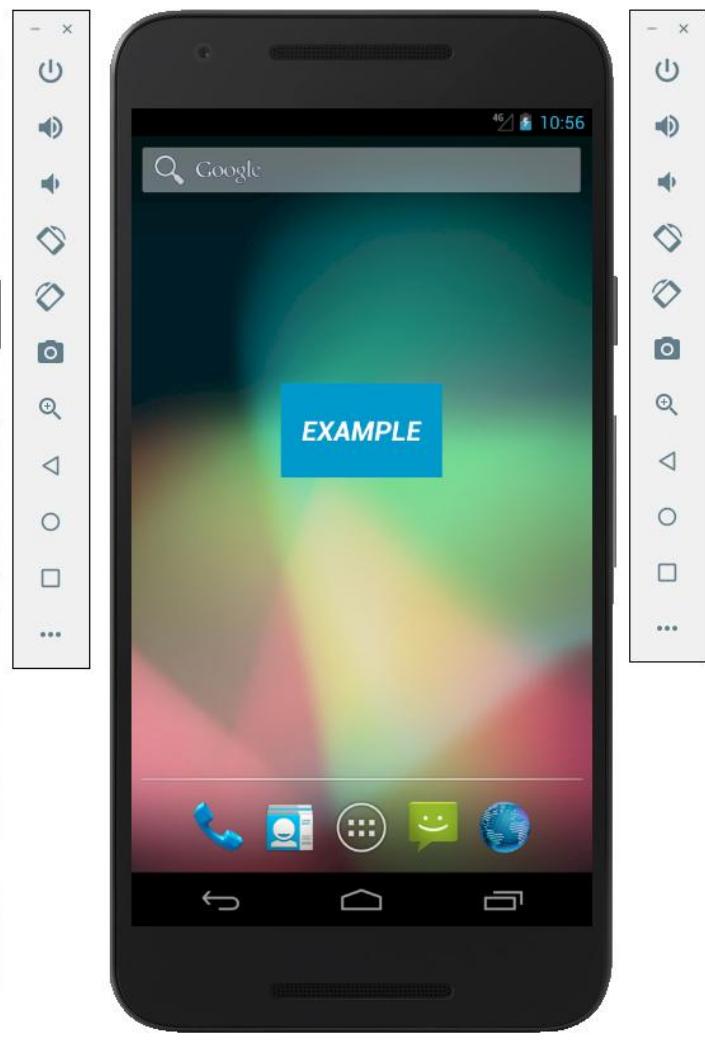


Рисунок 37 – Готовый виджет

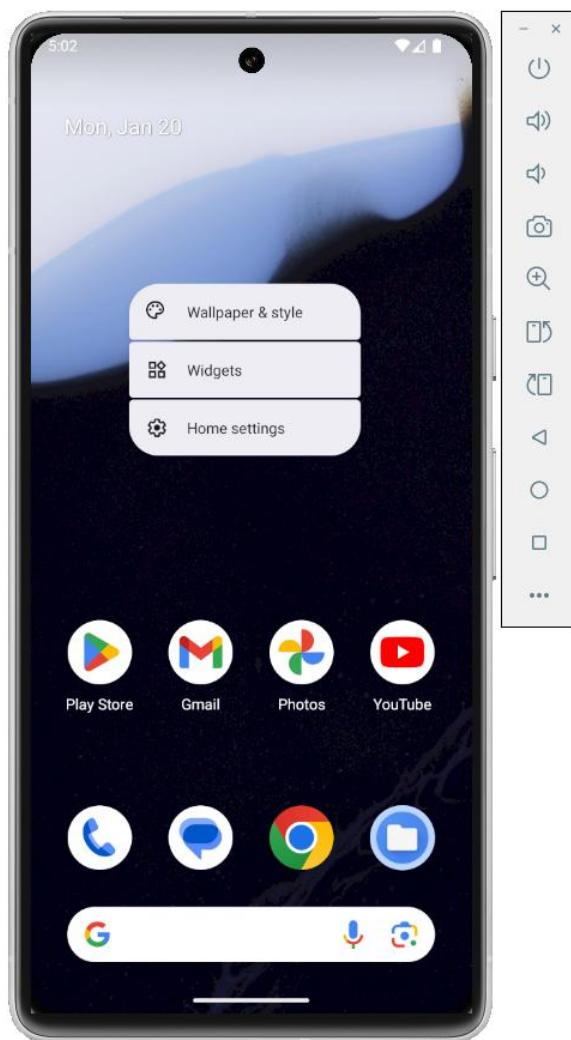


Рисунок 38 – Добавление виджета в API 35 (Android 15)

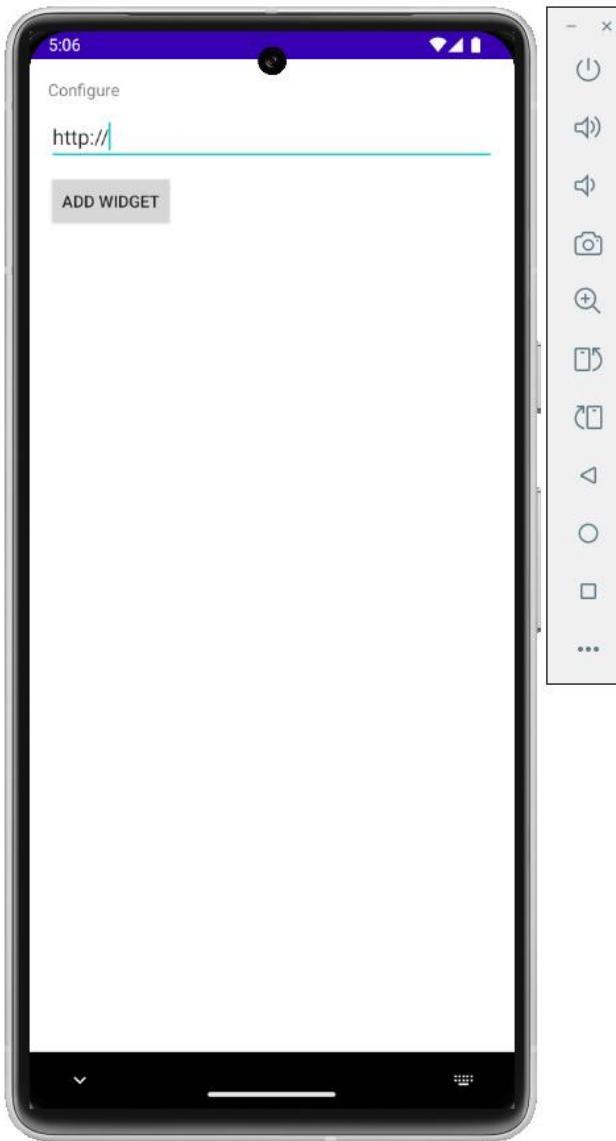


Рисунок 39 – Окно настройки виджета (API 35)

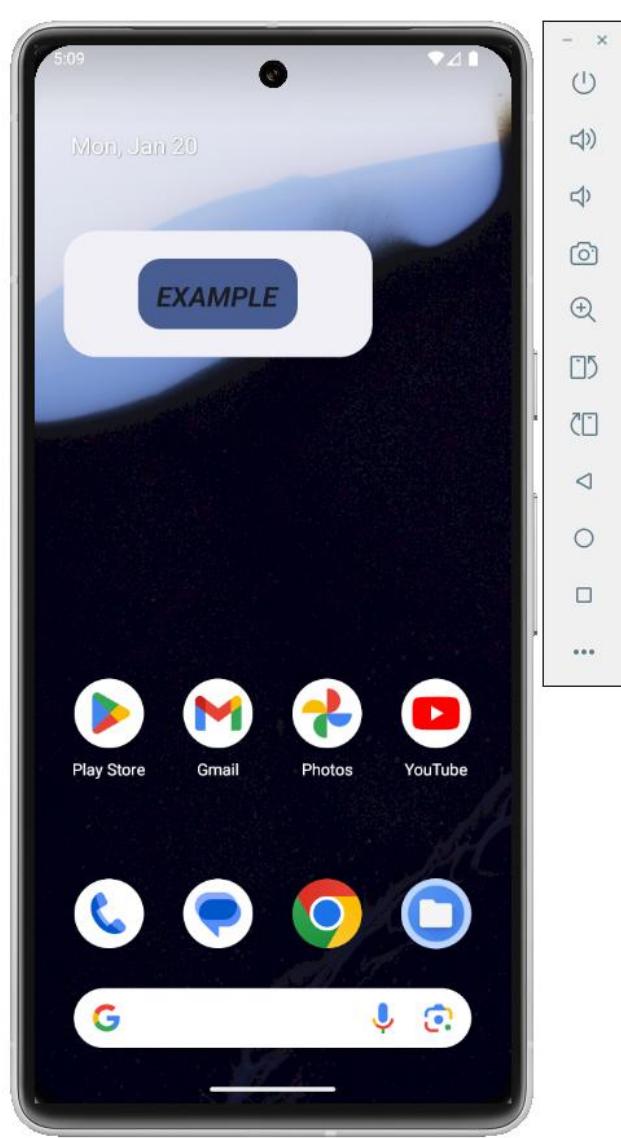


Рисунок 40 – Готовый виджет (API 35)