1.6 App resources

resource ID

A resource ID is composed of a resource type, a period, and a resource name.

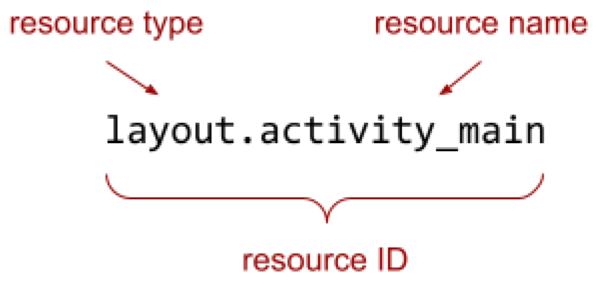
resource type

A resource type is a generic "type" by which resources are grouped, like `layout`, `drawable`, and `string`.

resource name

A resource name is a file name (without the extension) or a value specified in XML with the `name` attribute.

Figure 1.6.1: Parts of a resource ID.



Feedback?

R class

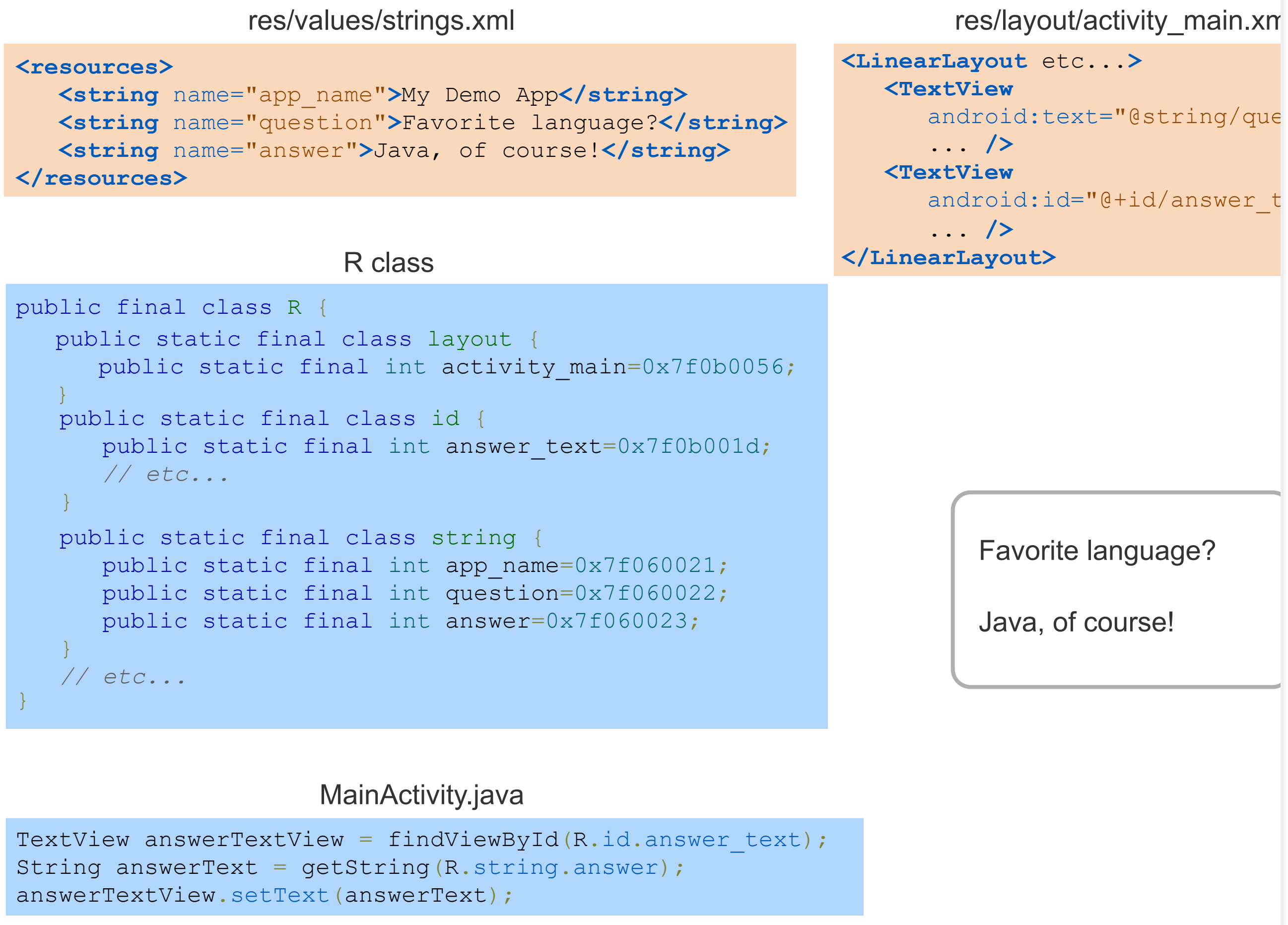
Android Studio generates a hidden R class containing subclasses named after each resource type and a static integer variable for each resource name.

android:id

Views in a layout file use the android:id attribute and **`@+id`** attribute value to specify the view's resource ID.

PARTICIPATION ACTIVITY 1.6.2: App resources and the R class.

Start 2x speed



res/values/strings.xml

```
<resources>
   <string name="app_name">My Demo App</string>
   <string name="question">Favorite language?</string>
   <string name="answer">Java, of course!</string>
</resources>
```

res/layout/activity_main.xm

```
<LinearLayout etc...>
   <TextView
      android:text="@string/que
      ... />
   <TextView
      android:id="@+id/answer_t
      ... />
</LinearLayout>
```

R class

```
public final class R {
   public static final class layout {
      public static final int activity_main=0x7f0b0056;
   }
   public static final class id {
      public static final int answer_text=0x7f0b001d;
      // etc...
   }
   public static final class string {
      public static final int app_name=0x7f060021;
      public static final int question=0x7f060022;
      public static final int answer=0x7f060023;
   }
   // etc...
}
```

MainActivity.java

```
TextView answerTextView = findViewById(R.id.answer_text);
String answerText = getString(R.string.answer);
answerTextView.setText(answerText);
```

Captions

1. strings.xml contains string resources. activity_main.xml is a layout resource.
2. Android Studio automatically creates the R class. The equivalent Java code is shown. The resource ID layout.activity_main is assigned a unique hexadecimal number
3. A TextView uses "@+id/" to specify a resource ID that is added to the subclass R.id.
4. String resources in strings.xml are added to the R.string subclass.
5. The first TextView references the string resource named "question", which automatically displays in the UI.

6. MainActivity uses findViewById() with the resource name defined in the R.id subclass to get a reference to the second TextView.
7. The Activity method getString() returns the string resource named "answer". The View method setText() displays the string in the UI.

Feedback?

This section does not contain presentation elements.

How was this section? Provide feedback