Mobile Application Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)
Dr. Siobhán Drohan

Department of Computing, Maths & Physics
Waterford Institute of Technology

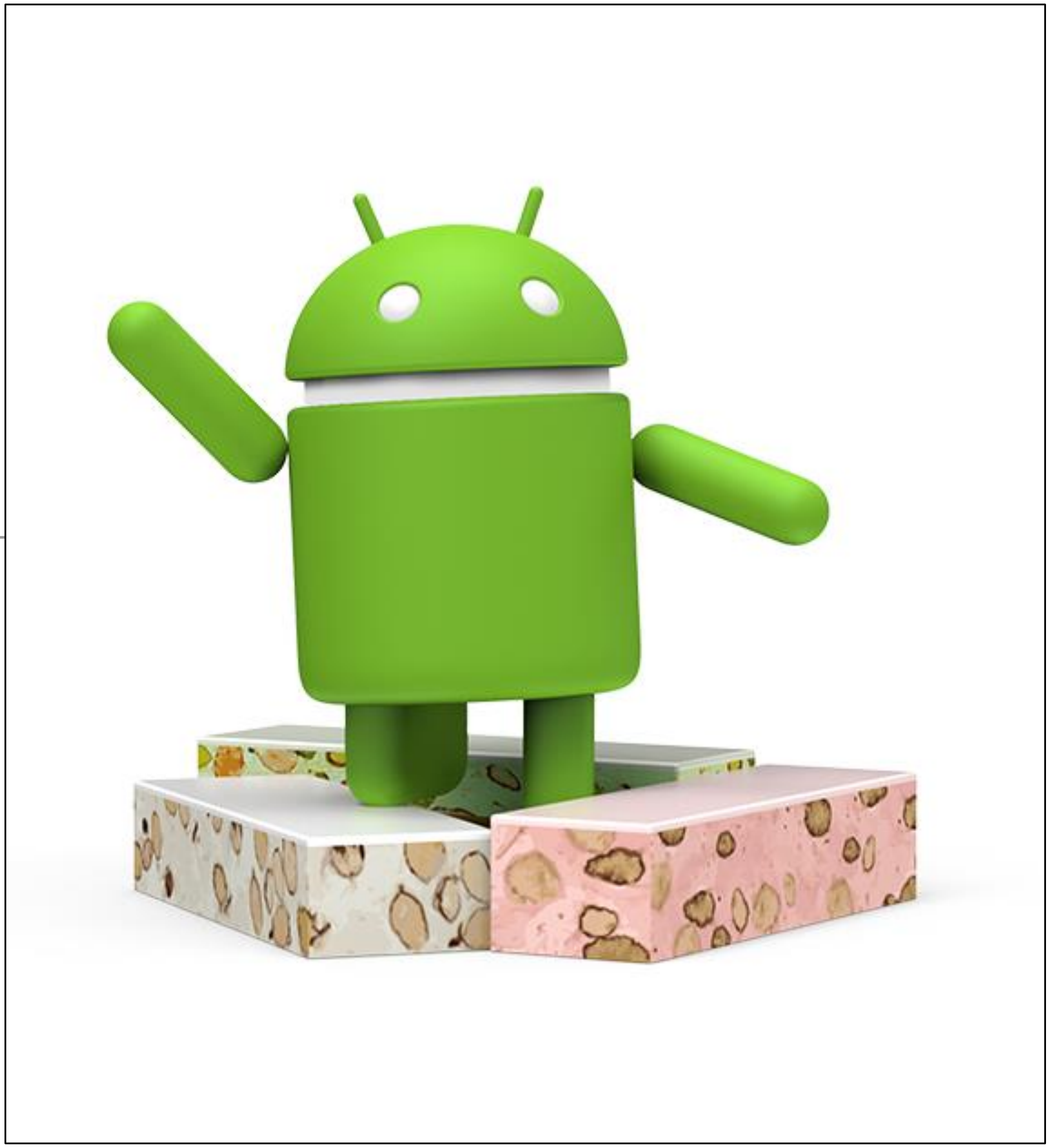
http://www.wit.ie

http://elearning.wit.ie

Second Android Application

MyRent

Starting with MyRent V00



MyRent

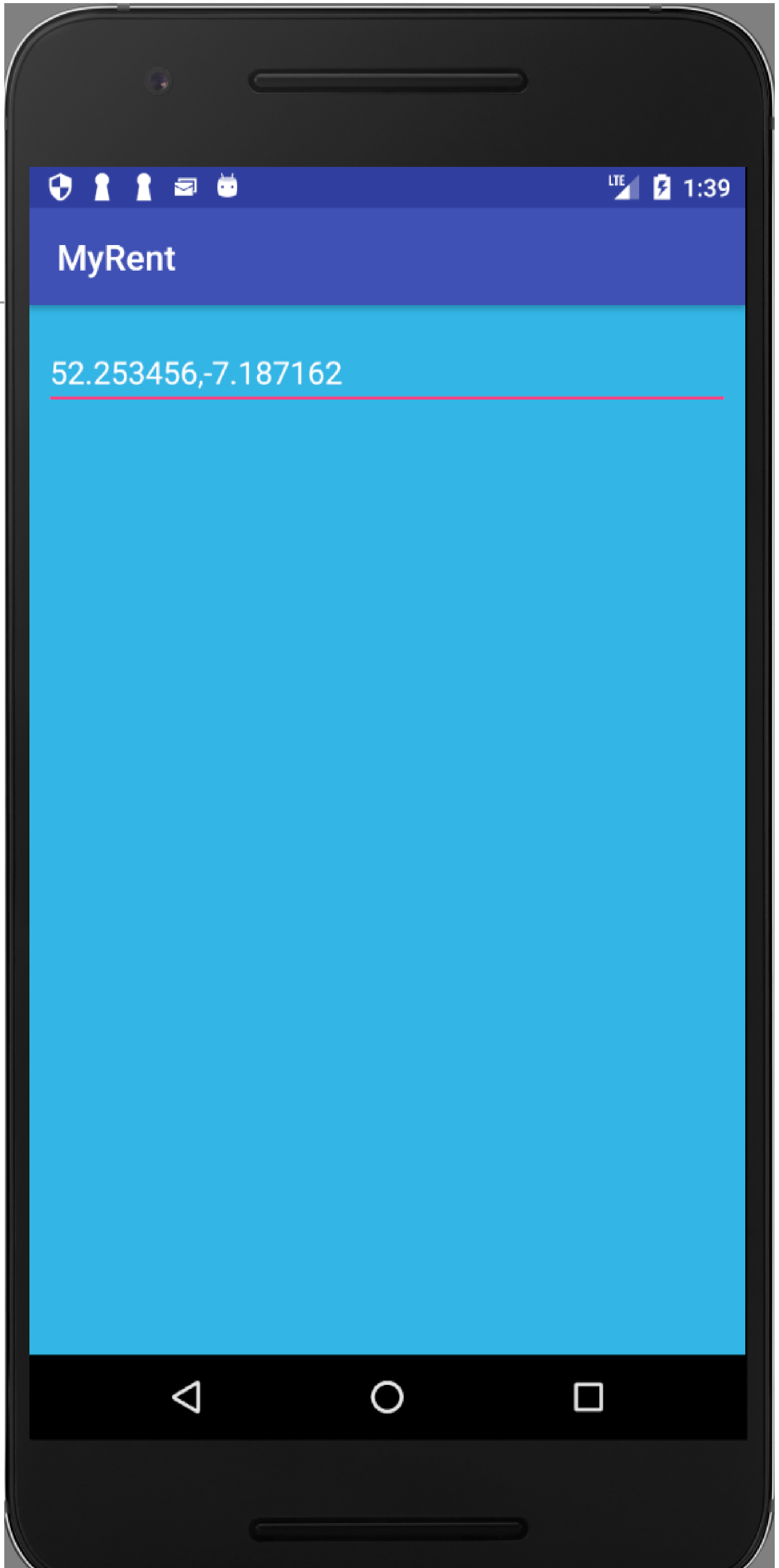
- An application to track location, rental status and other information about properties.
- Develop in several versions to include:
  - multiple properties
  - location map
  - images
  - contact information
  - etc…

MyRent Versions (week 3 – 5 inclusive)

MyRent V00	One Activity with a simple TextField where user enters two coordinates, separated by comma.
MyRent V01	One Activity with multiple widgets to display Location, Status and Rented?
MyRent V02	Two Activities utilising a '*Master-Detail*' pattern. *Master* holds a list of rented residences whereas the *Detail* is the Activity from V01.
MyRent V03	Significant update to include an Action Bar, allowing new residence creation and navigation to existing ones. Also includes a Date Picker Dialog that can add a Registration Date for the residence.
MyRent V04	Allows Residences to save to and load from a file. Contents are loaded on launch and saved automatically as Residences are added / updated.
MyRent V05	Evolution of the App Navigation to provide navigation from the Activity back to the List of Activities.
MyRent V06	Enable app to select a contact from the phone's contact list and send an email to the selected user. Requesting permissions is included here.
MyRent V07	Use of Fragments to create a multi-pane screen, which can later support landscape orientations of our app.

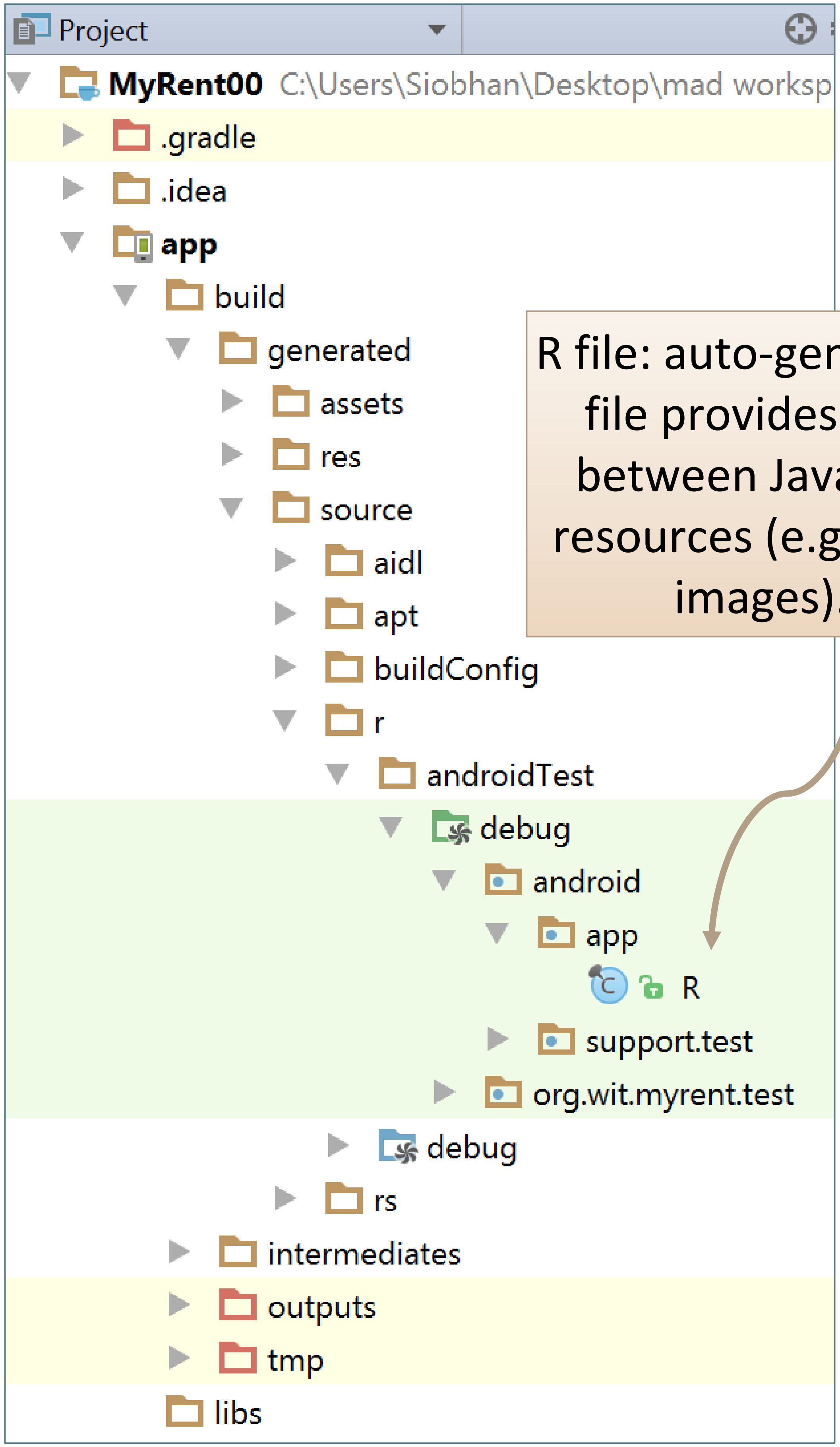
MyRent V00

- Simple TextField
- User enters two coordinates, separated by comma



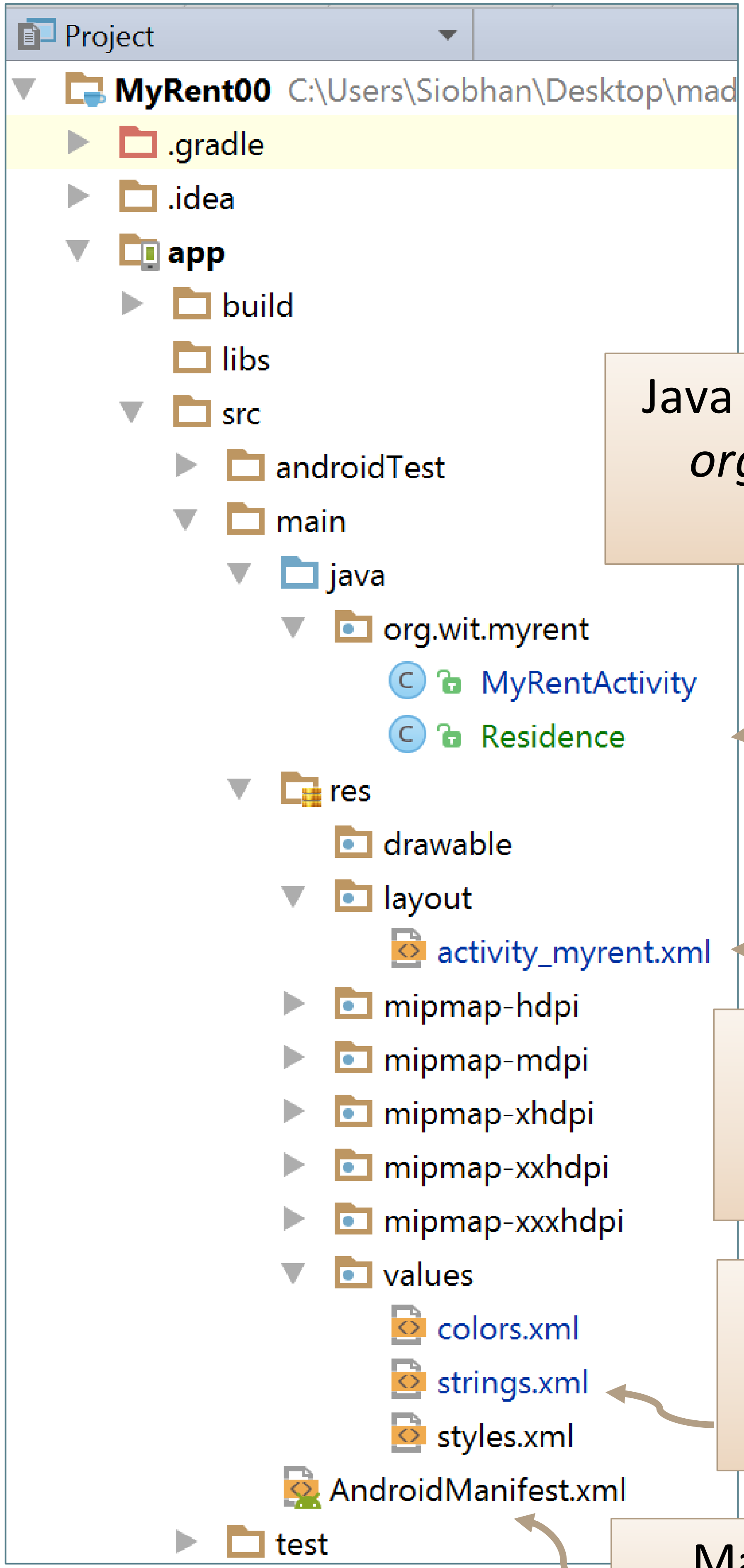
52.253456,-7.187162

Project Structure



```
Project
MyRent00  C:\Users\Siobhan\Desktop\mad worksp
    .gradle
    .idea
    app
        build
            generated
                assets
                res
                source
                    aidl
                    apt
                    buildConfig
                    r
                        androidTest
                            debug
                                android
                                    app
                                        R
                                    support.test
                                org.wit.myrent.test
                        debug
                    rs
            intermediates
            outputs
            tmp
        libs
```

R file: auto-generated file provides glue between Java and resources (e.g. XML, images).



```
Project
MyRent00  C:\Users\Siobhan\Desktop\mad
    .gradle
    .idea
    app
        build
        libs
        src
            androidTest
            main
                java
                    org.wit.myrent
                        MyRentActivity
                        Residence
                res
                    drawable
                    layout
                        activity_myrent.xml
                    mipmap-hdpi
                    mipmap-mdpi
                    mipmap-xhdpi
                    mipmap-xxhdpi
                    mipmap-xxxhdpi
                    values
                        colors.xml
                        strings.xml
                        styles.xml
                AndroidManifest.xml
            test
```

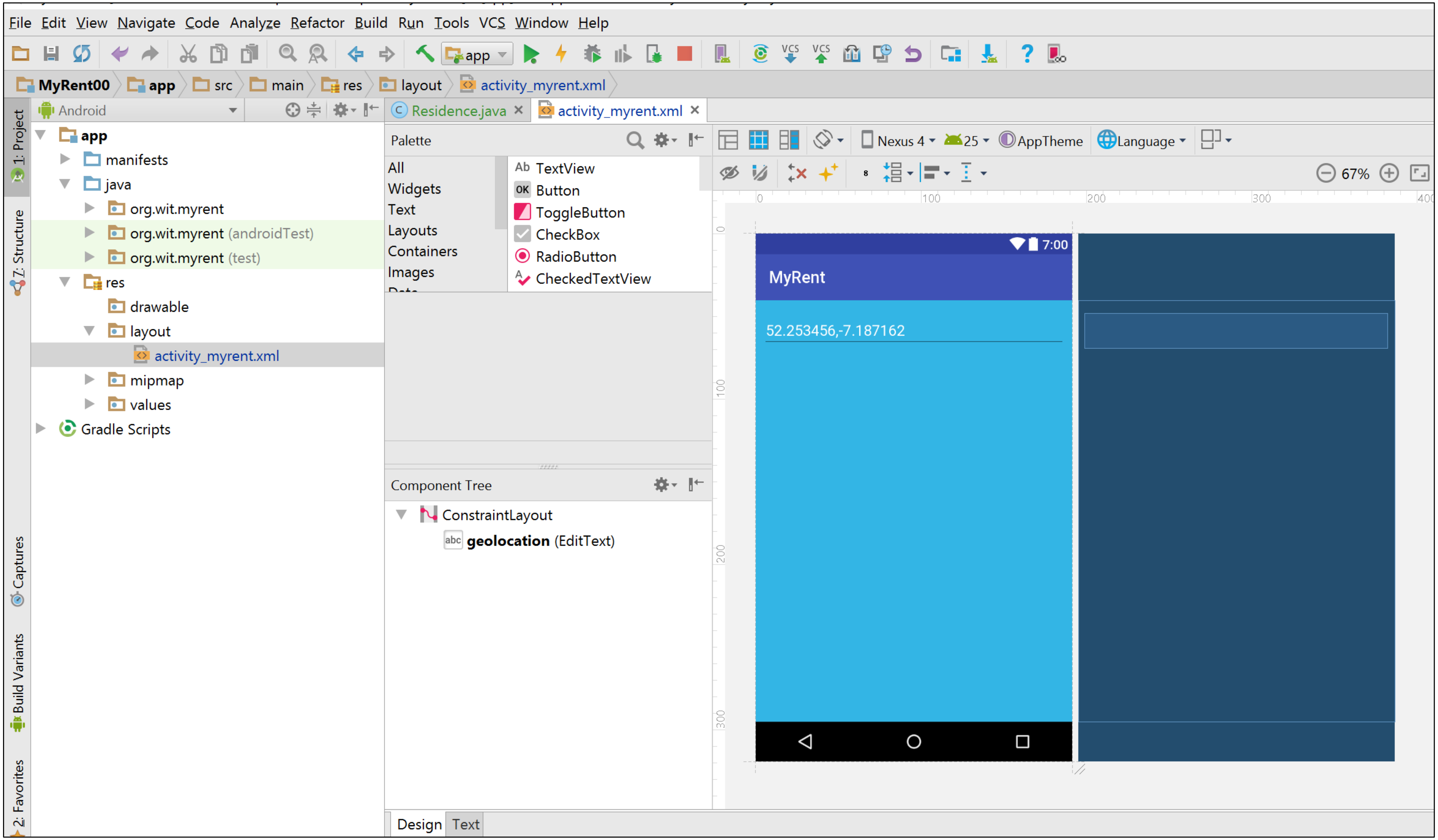
Java files located in *org.wit.myrent* package

Layout XML file: specifies screen layout

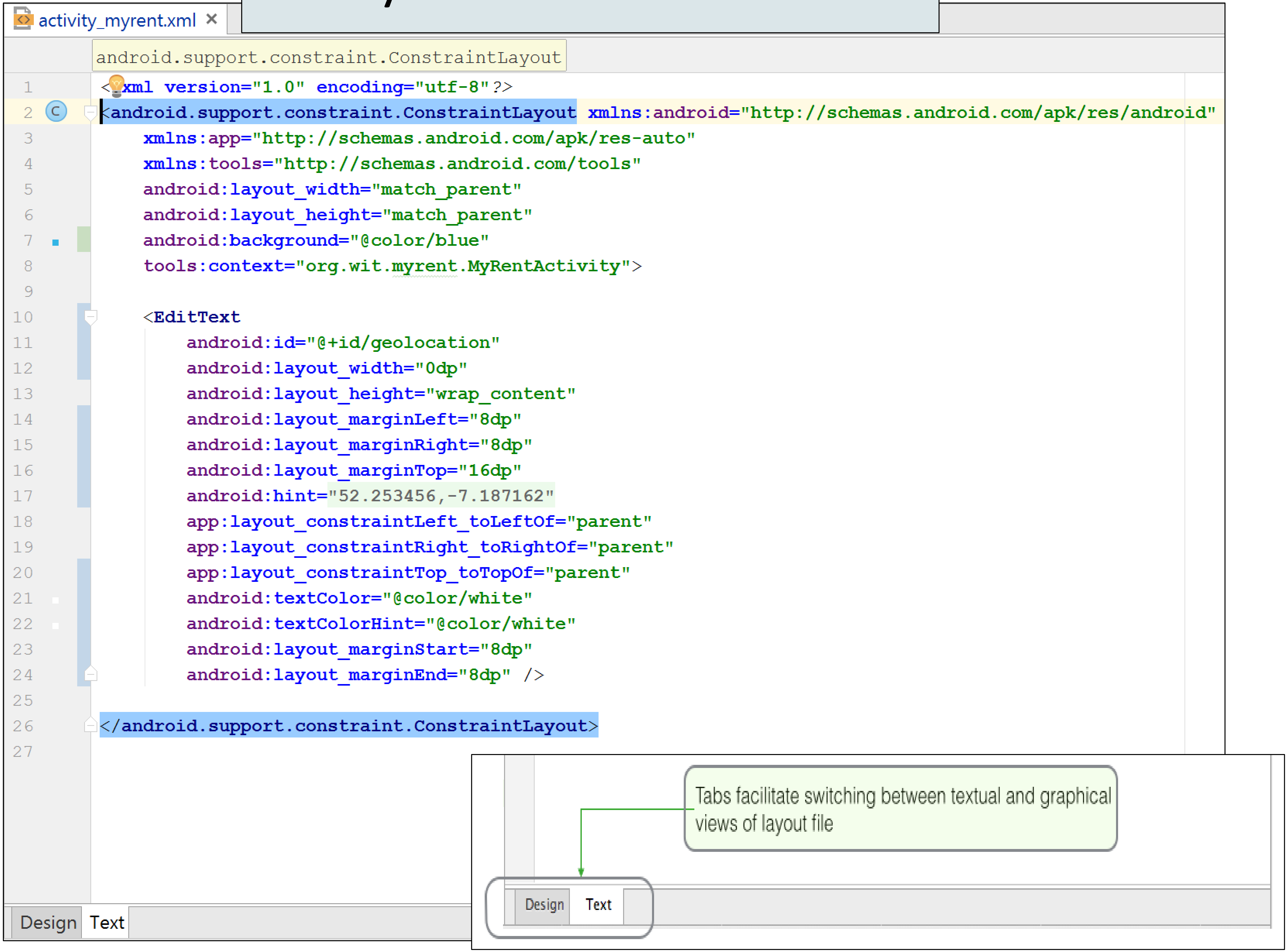
Strings.xml: contains text used in app

Manifest – holds everything together

Layout Editor - Visual



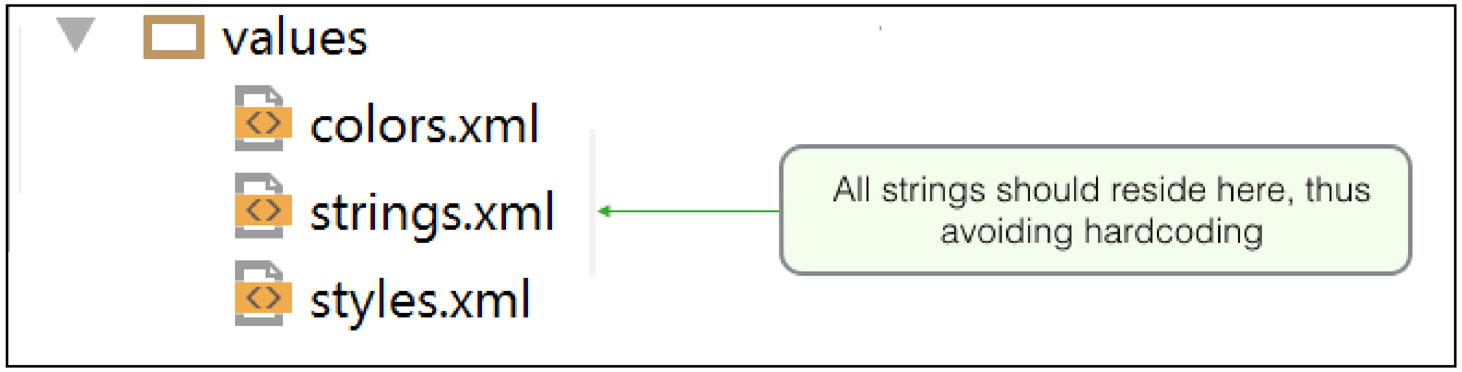
Layout Editor - Textual



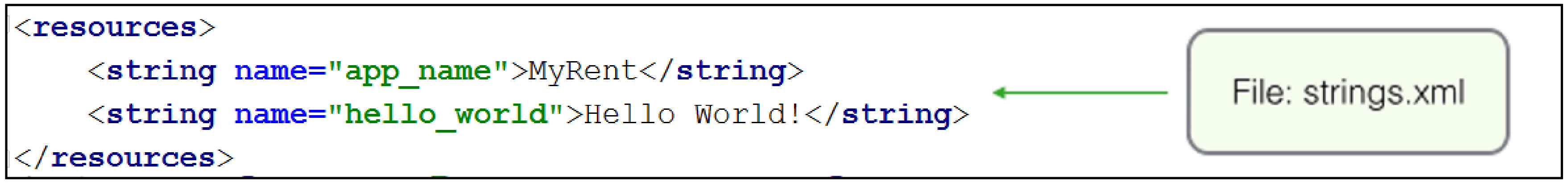
```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/blue"
    tools:context="org.wit.myrent.MyRentActivity">

    <EditText
        android:id="@+id/geolocation"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="16dp"
        android:hint="52.253456,-7.187162"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:textColor="@color/white"
        android:textColorHint="@color/white"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp" />

</android.support.constraint.ConstraintLayout>
```

Tabs facilitate switching between textual and graphical views of layout file

strings.xml

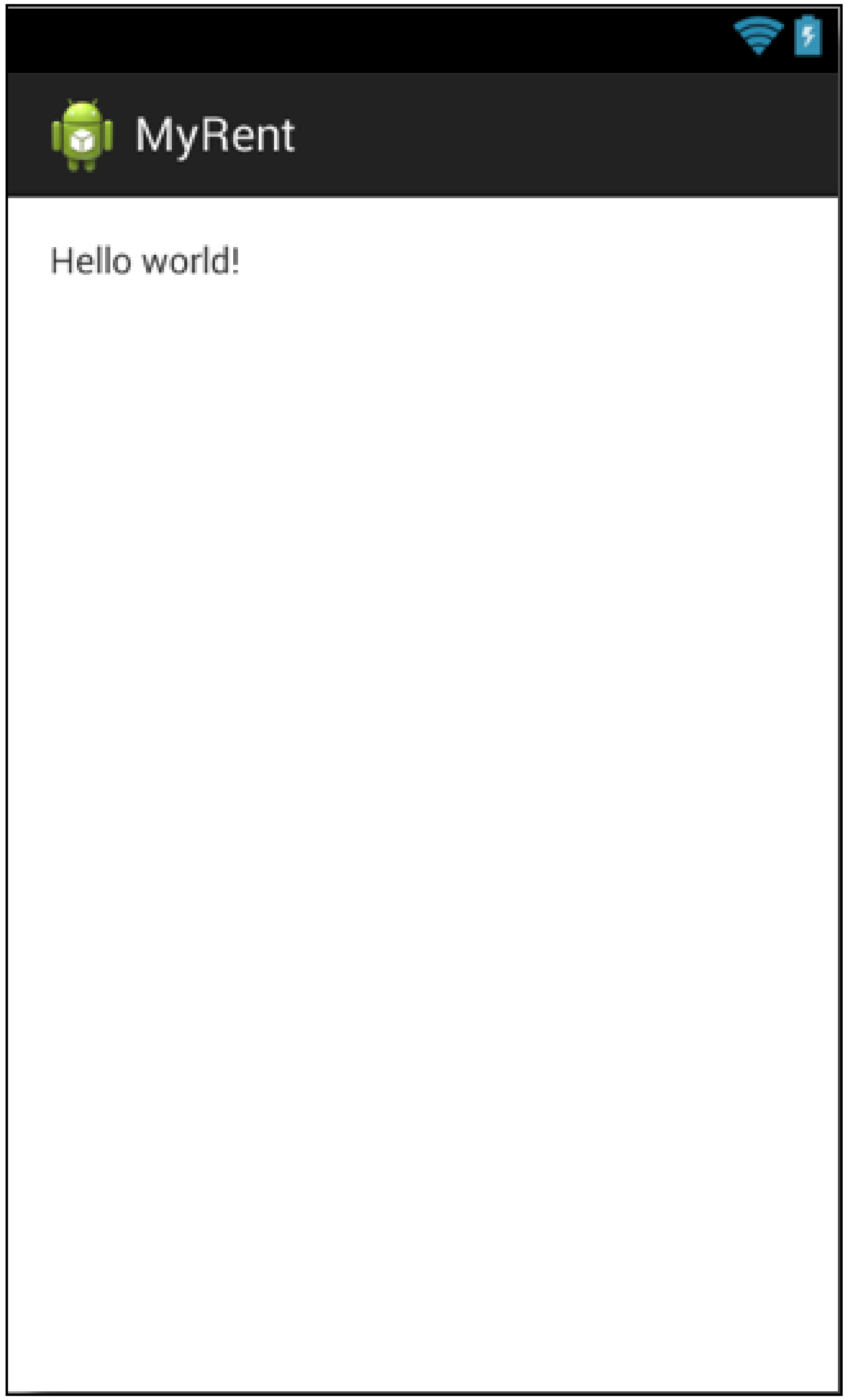


All strings should reside here, thus avoiding hardcoding



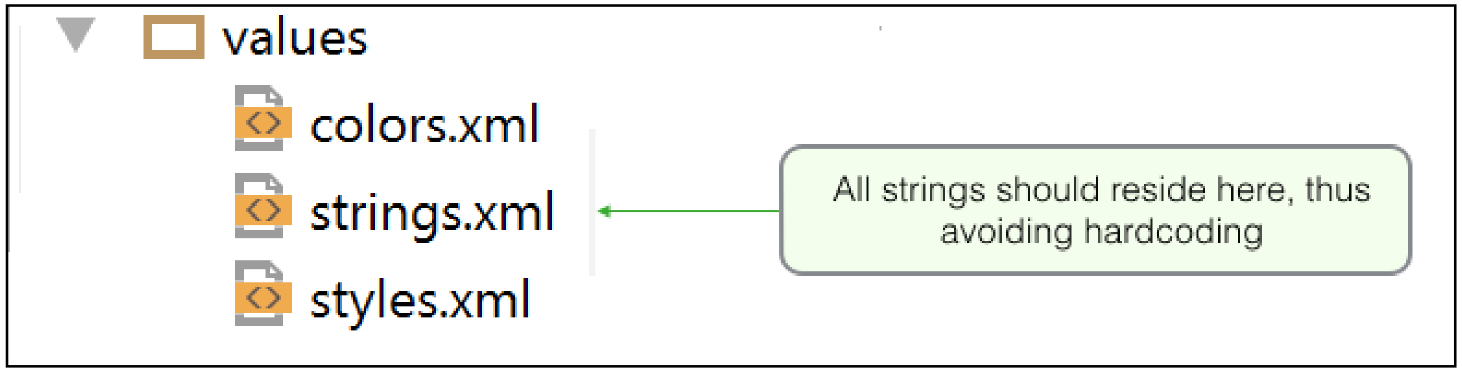
```
<resources>
    <string name="app_name">MyRent</string>
    <string name="hello_world">Hello World!</string>
</resources>
```

File: strings.xml

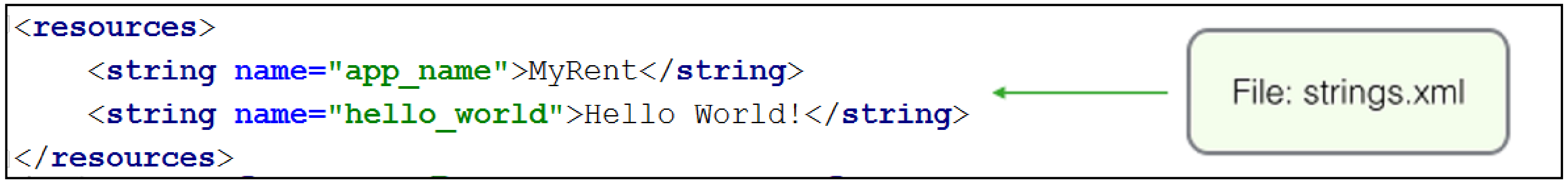


Hello world!

strings.xml - TextView Example

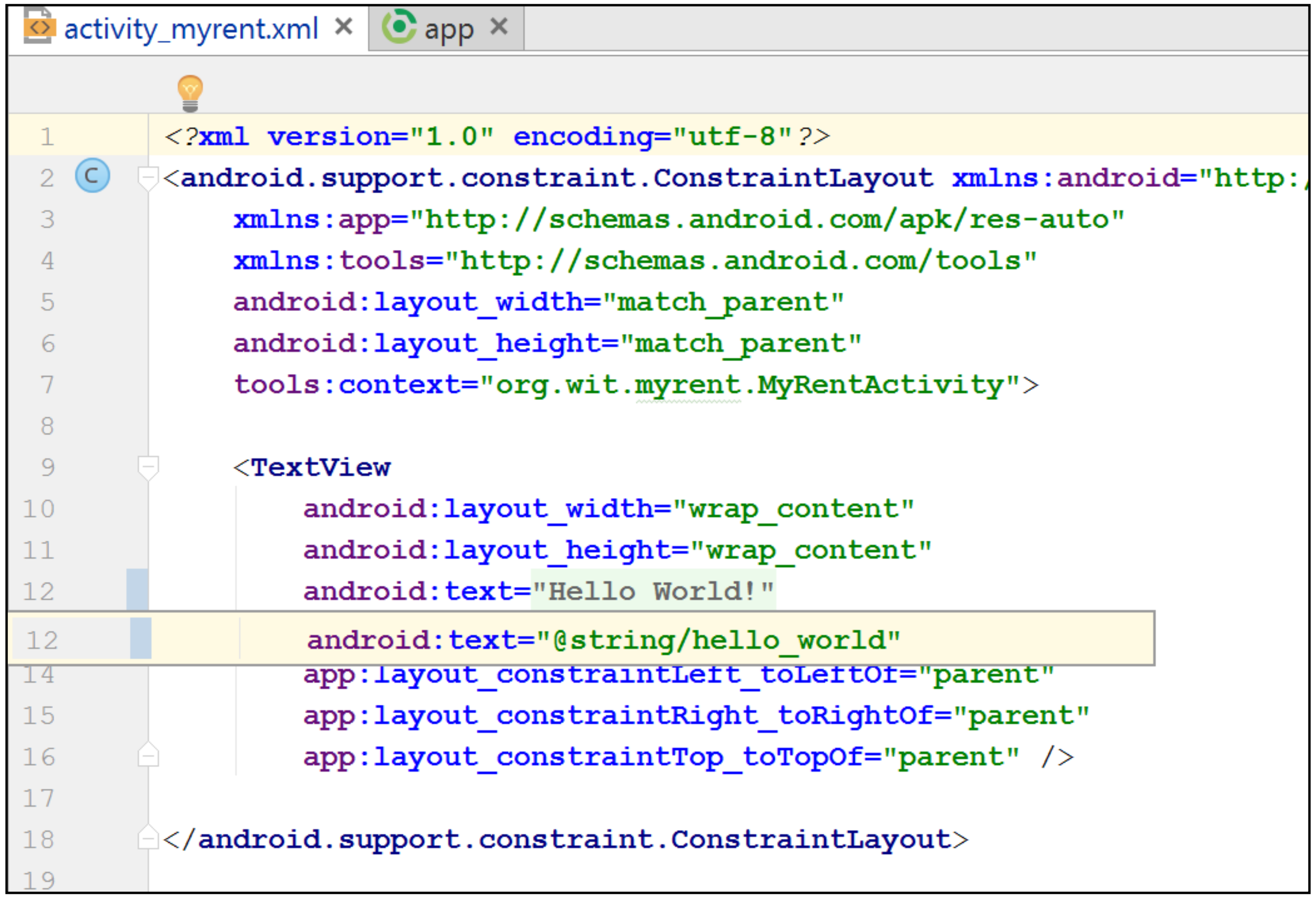


All strings should reside here, thus avoiding hardcoding

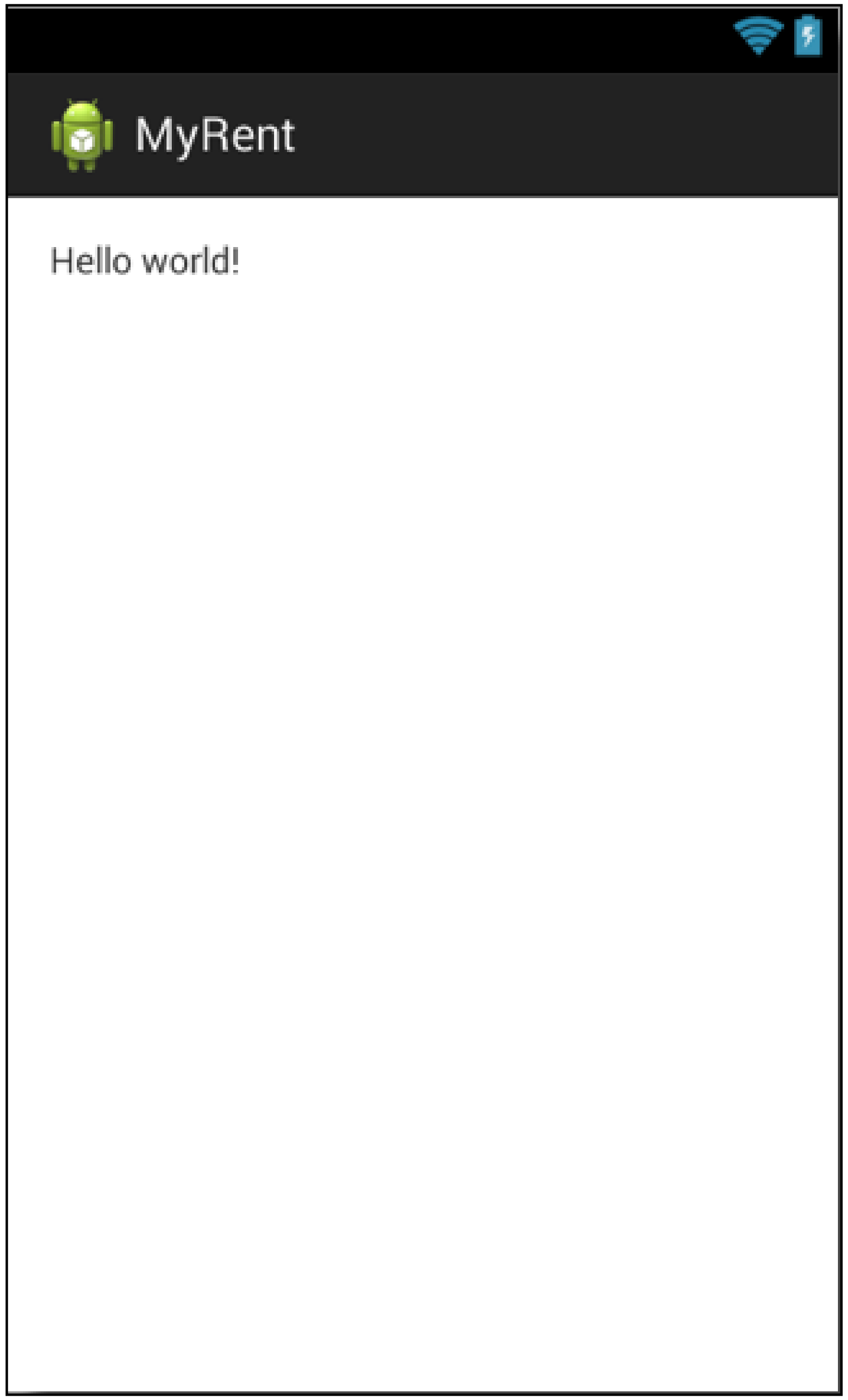


```
<resources>
    <string name="app_name">MyRent</string>
    <string name="hello_world">Hello World!</string>
</resources>
```

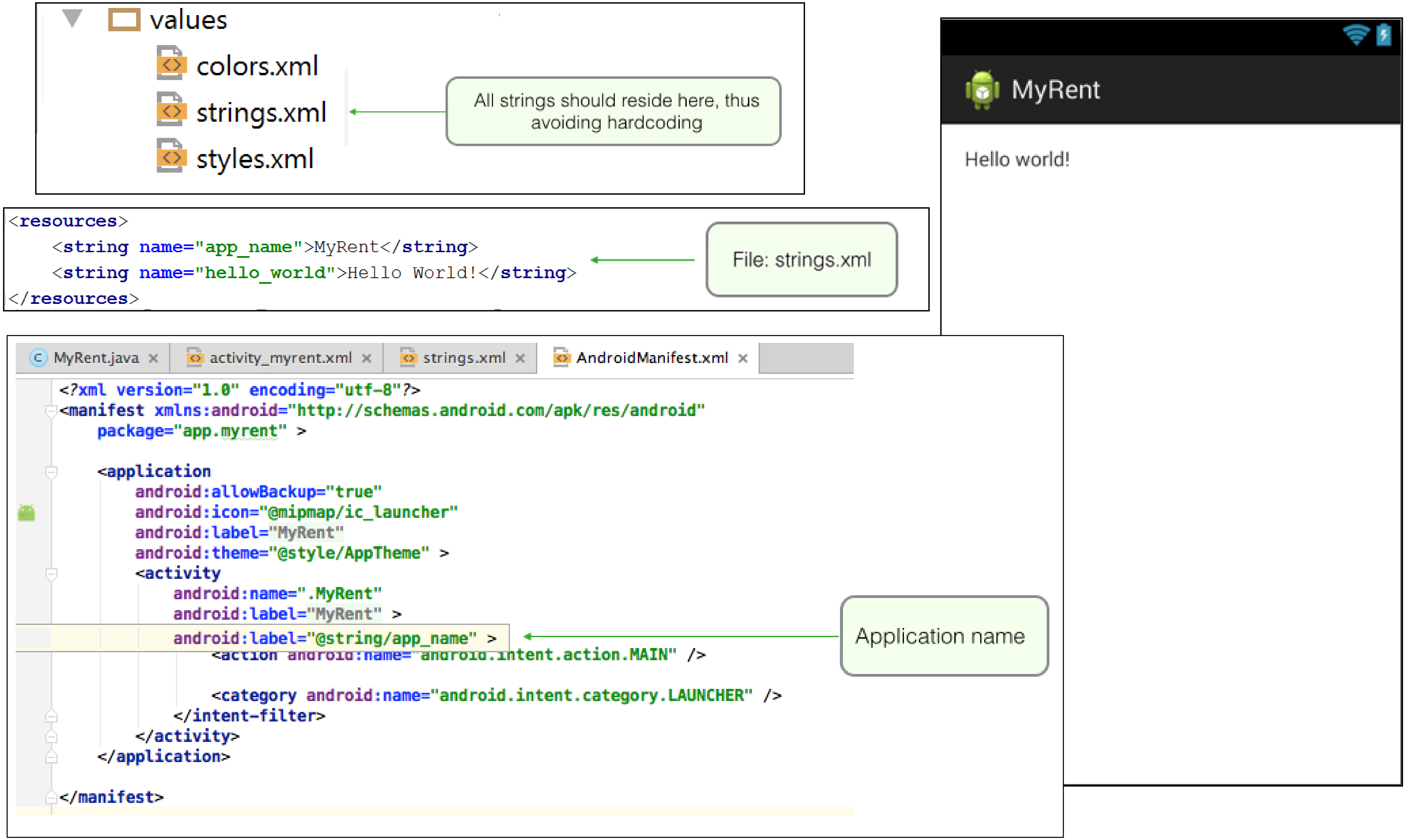
File: strings.xml



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http:/
    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="org.wit.myrent.MyRentActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:text="@string/hello_world"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```



strings.xml – Application Name Example



All strings should reside here, thus avoiding hardcoding

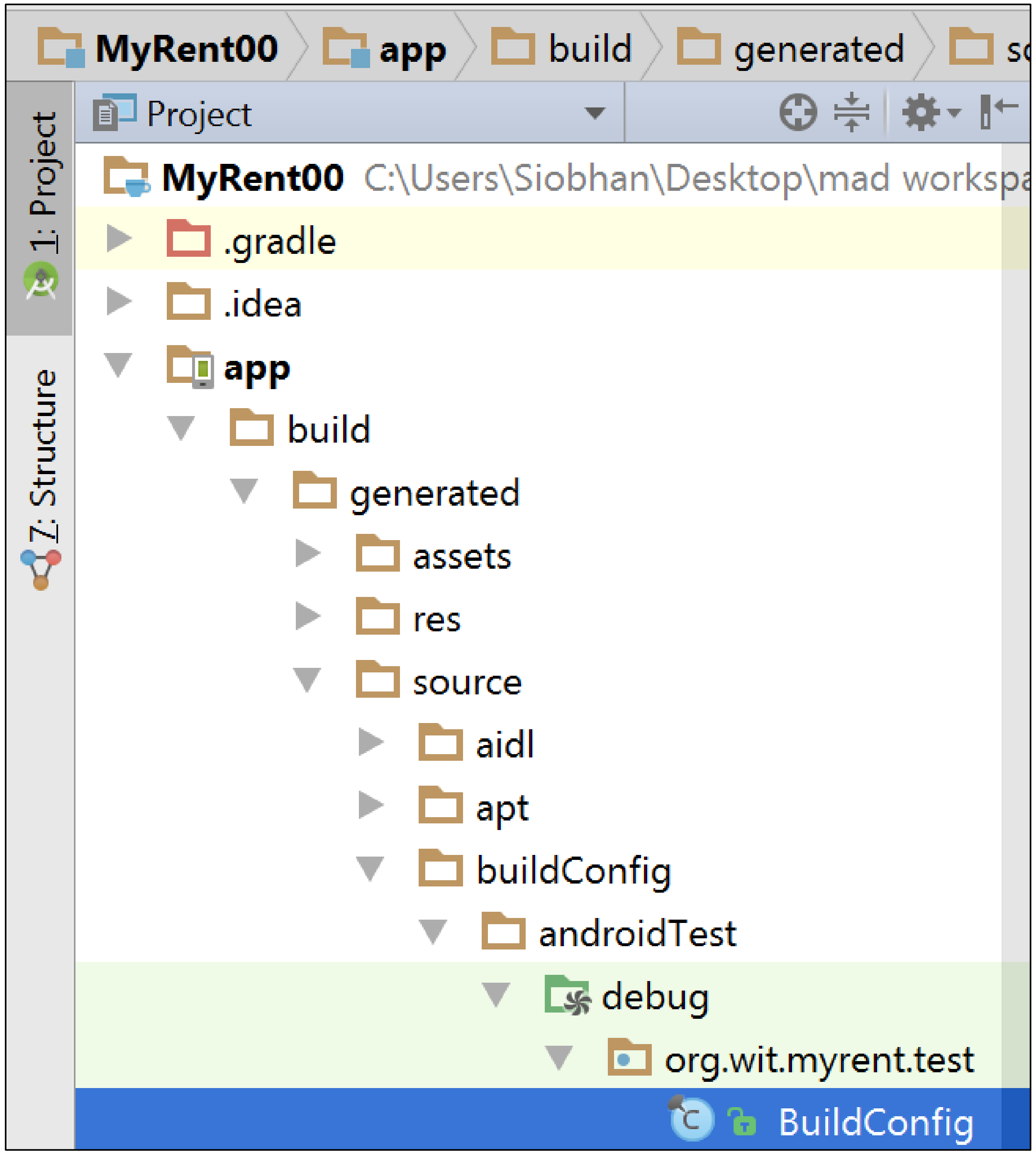
```
<resources>
    <string name="app_name">MyRent</string>
    <string name="hello_world">Hello World!</string>
</resources>
```

File: strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="app.myrent" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="MyRent"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MyRent"
            android:label="MyRent" >
            android:label="@string/app_name" >
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Application name

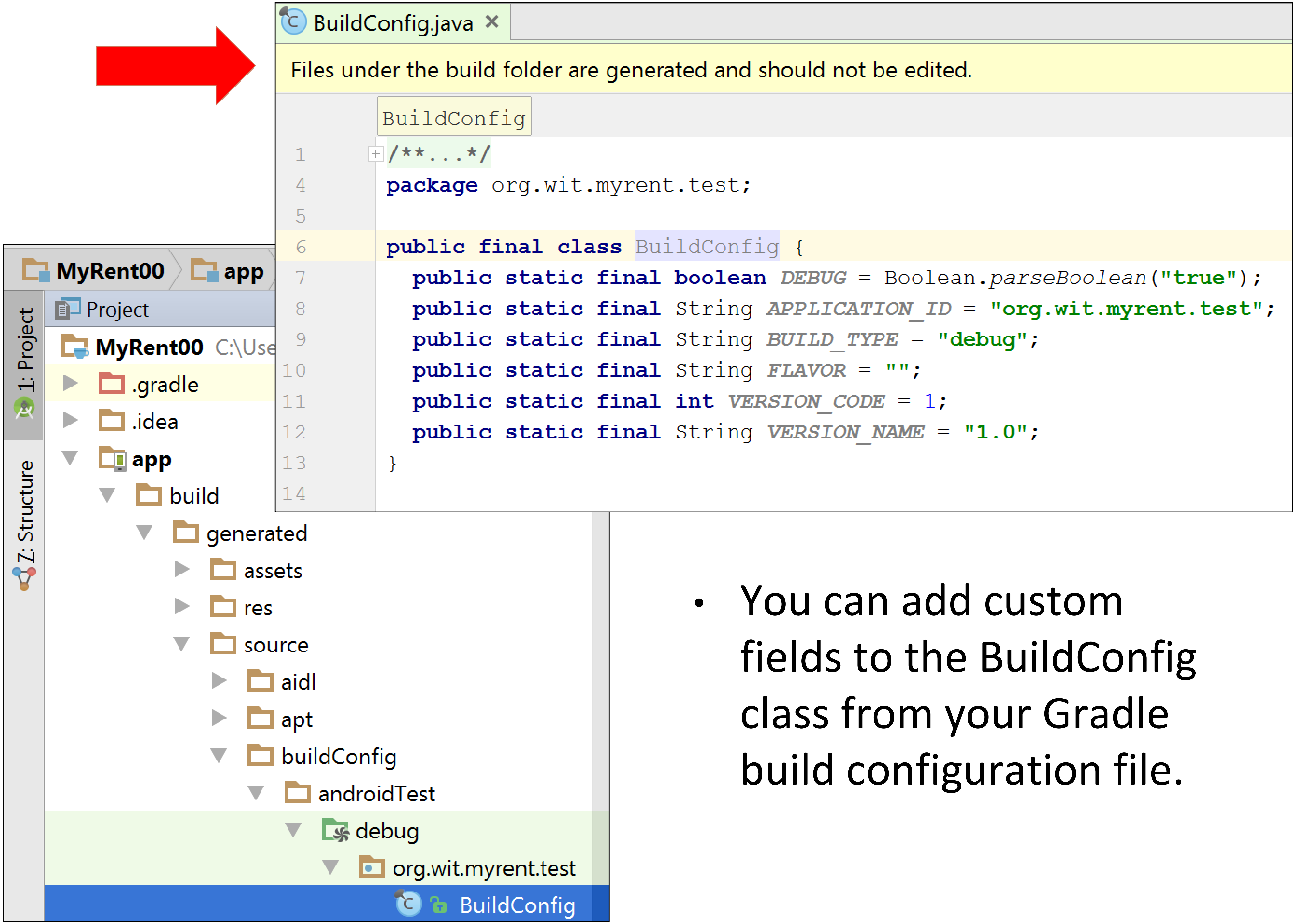
BuildConfig.java



```
MyRent00 > app > build > generated > sc
Project
MyRent00  C:\Users\Siobhan\Desktop\mad workspa
  ▶ .gradle
  ▶ .idea
  ▼ app
      ▼ build
          ▼ generated
              ▶ assets
              ▶ res
              ▼ source
                  ▶ aidl
                  ▶ apt
                  ▼ buildConfig
                      ▼ androidTest
                          ▼ debug
                              ▼ org.wit.myrent.test
                                    BuildConfig
```

- At build time, Gradle generates the BuildConfig class so your app code can inspect information about the current build.

https://developer.android.com/studio/build/gradle-tips.html



BuildConfig.java ×

Files under the build folder are generated and should not be edited.

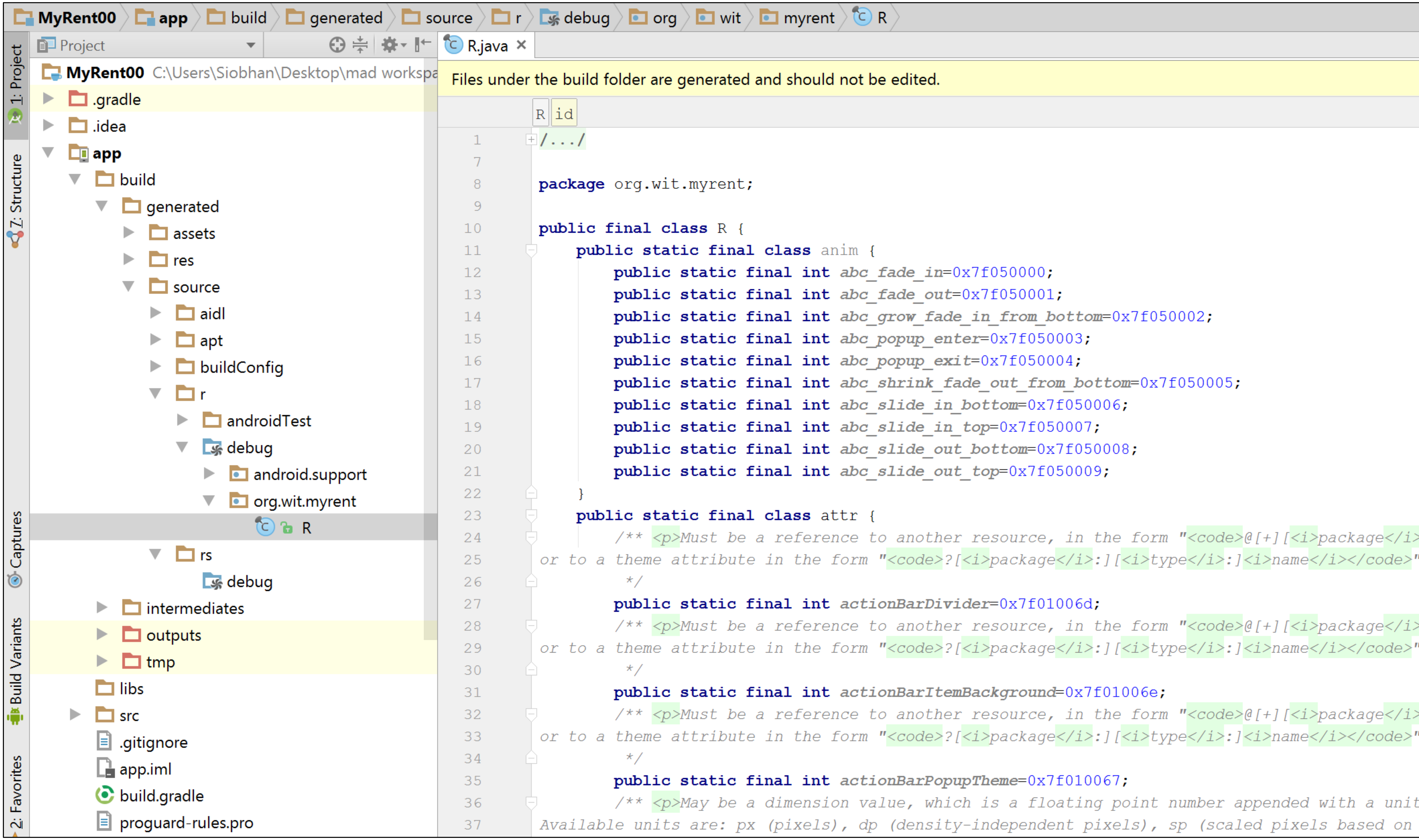
BuildConfig

```
/**...*/
package org.wit.myrent.test;

public final class BuildConfig {
  public static final boolean DEBUG = Boolean.parseBoolean("true");
  public static final String APPLICATION_ID = "org.wit.myrent.test";
  public static final String BUILD_TYPE = "debug";
  public static final String FLAVOR = "";
  public static final int VERSION_CODE = 1;
  public static final String VERSION_NAME = "1.0";
}
```

- You can add custom fields to the BuildConfig class from your Gradle build configuration file.

https://developer.android.com/studio/build/gradle-tips.html

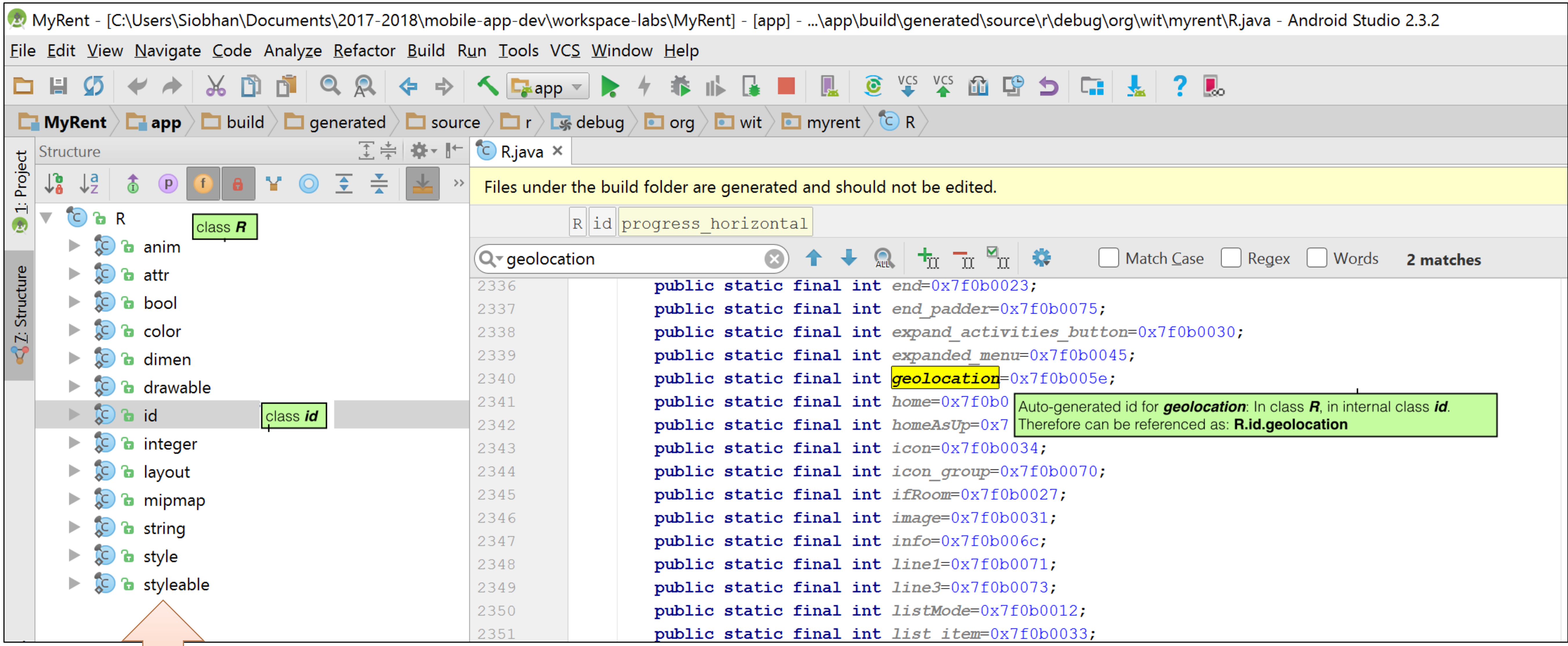
R.Java



Files under the build folder are generated and should not be edited.

```java
/.../

package org.wit.myrent;

public final class R {
    public static final class anim {
        public static final int abc_fade_in=0x7f050000;
        public static final int abc_fade_out=0x7f050001;
        public static final int abc_grow_fade_in_from_bottom=0x7f050002;
        public static final int abc_popup_enter=0x7f050003;
        public static final int abc_popup_exit=0x7f050004;
        public static final int abc_shrink_fade_out_from_bottom=0x7f050005;
        public static final int abc_slide_in_bottom=0x7f050006;
        public static final int abc_slide_in_top=0x7f050007;
        public static final int abc_slide_out_bottom=0x7f050008;
        public static final int abc_slide_out_top=0x7f050009;
    }
    public static final class attr {
        /** <p>Must be a reference to another resource, in the form "<code>@[+][<i>package</i>
or to a theme attribute in the form "<code>?[<i>package</i>:][<i>type</i>:]<i>name</i></code>"
         */
        public static final int actionBarDivider=0x7f01006d;
        /** <p>Must be a reference to another resource, in the form "<code>@[+][<i>package</i>
or to a theme attribute in the form "<code>?[<i>package</i>:][<i>type</i>:]<i>name</i></code>"
         */
        public static final int actionBarItemBackground=0x7f01006e;
        /** <p>Must be a reference to another resource, in the form "<code>@[+][<i>package</i>
or to a theme attribute in the form "<code>?[<i>package</i>:][<i>type</i>:]<i>name</i></code>"
         */
        public static final int actionBarPopupTheme=0x7f010067;
        /** <p>May be a dimension value, which is a floating point number appended with a unit
Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on
```

R.Java



MyRent - [C:\Users\Siobhan\Documents\2017-2018\mobile-app-dev\workspace-labs\MyRent] - [app] - ...\app\build\generated\source\r\debug\org\wit\myrent\R.java - Android Studio 2.3.2

Files under the build folder are generated and should not be edited.

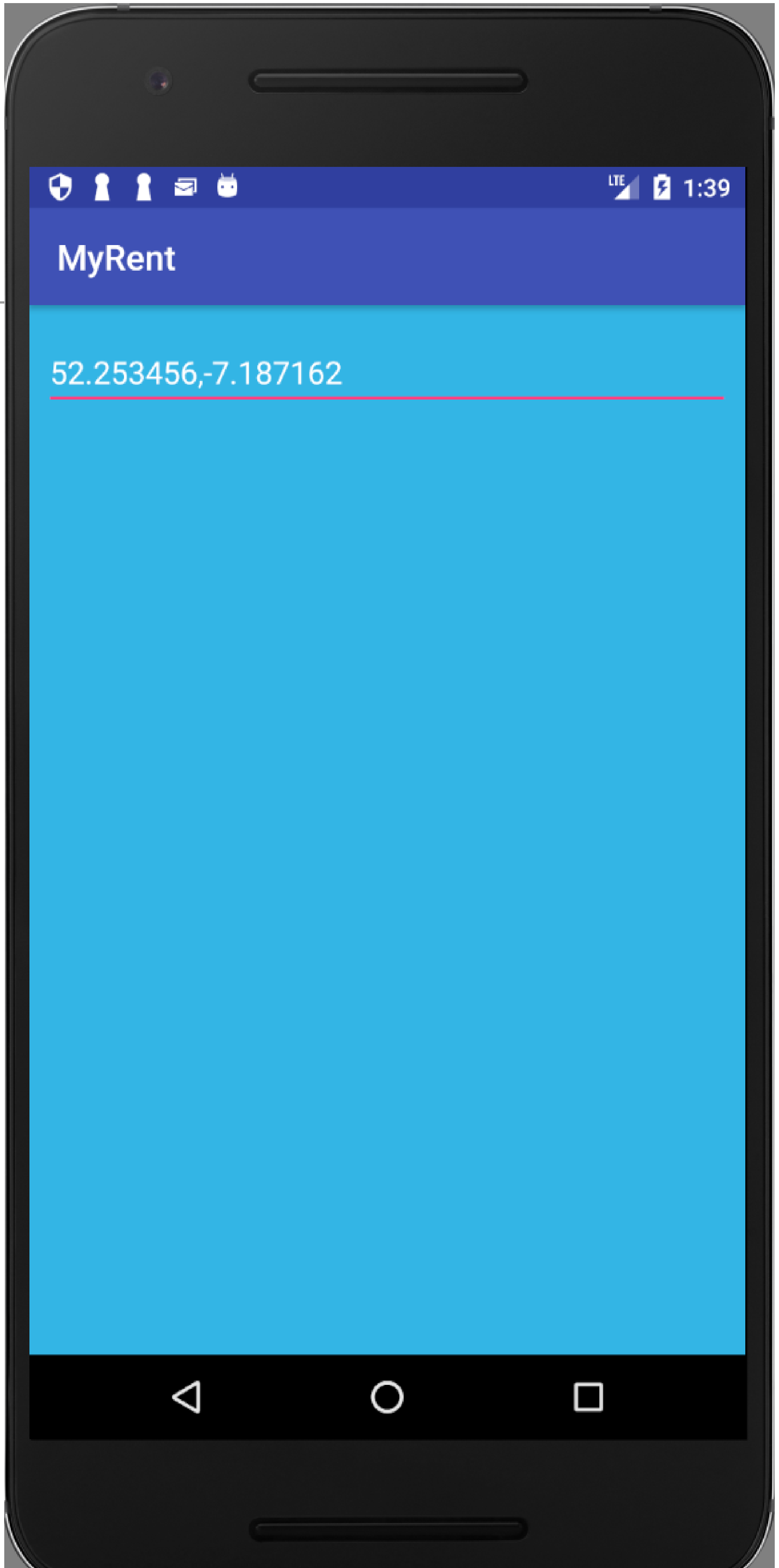
```
    public static final int end=0x7f0b0023;
    public static final int end_padder=0x7f0b0075;
    public static final int expand_activities_button=0x7f0b0030;
    public static final int expanded_menu=0x7f0b0045;
    public static final int geolocation=0x7f0b005e;
    public static final int home=0x7f0b0
    public static final int homeAsUp=0x7
    public static final int icon=0x7f0b0034;
    public static final int icon_group=0x7f0b0070;
    public static final int ifRoom=0x7f0b0027;
    public static final int image=0x7f0b0031;
    public static final int info=0x7f0b006c;
    public static final int line1=0x7f0b0071;
    public static final int line3=0x7f0b0073;
    public static final int listMode=0x7f0b0012;
    public static final int list_item=0x7f0b0033;
```

Auto-generated id for ***geolocation***: In class ***R***, in internal class ***id***.
Therefore can be referenced as: **R.id.geolocation**

Nested classes i.e. classes defined within another class.

MyRent V00

- Java classes:
  - Residence
  - MyRentActivity



52.253456,-7.187162

Residence

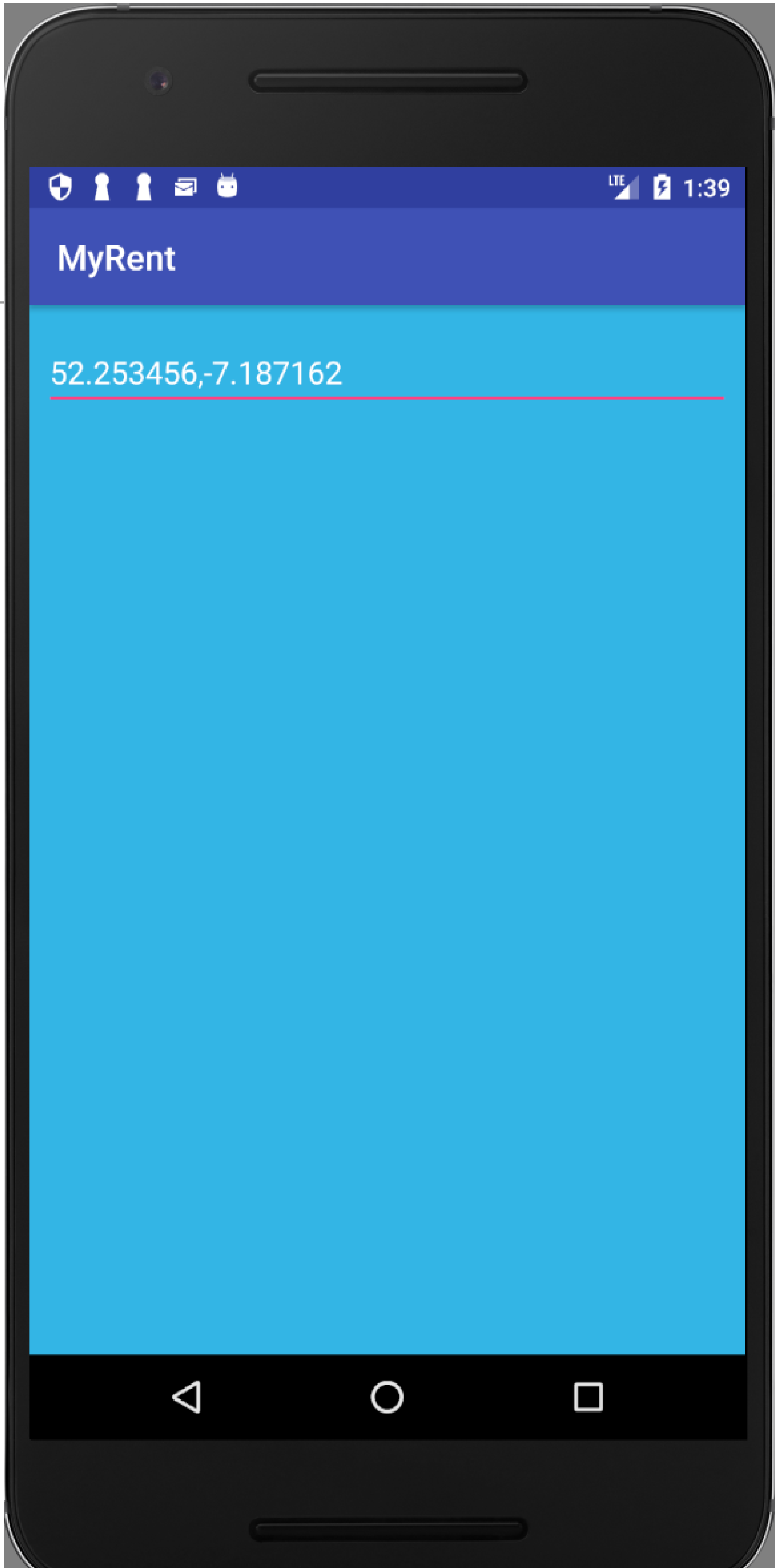
```java
package org.wit.myrent;
import java.util.Random;

public class Residence {

    private Long id;
    //a latitude longitude pair
    //example "52.4566,-6.5444"
    private String geolocation;

    public Residence(){
        id = unsignedLong();
    }

     /** Generate a long greater than zero
     * @return Unsigned Long value greater than zero
     */
    private Long unsignedLong() {
        long rndVal = 0;
        do {
            rndVal = new Random().nextLong();
        } while (rndVal <= 0);
        return rndVal;
    }

    public void setGeolocation(String geolocation){
        this.geolocation = geolocation;
    }

    public String getGeolocation(){
        return geolocation;
    }
}
```

Listeners in MyRent V00

- Listen for changes to the Geolocation input control.

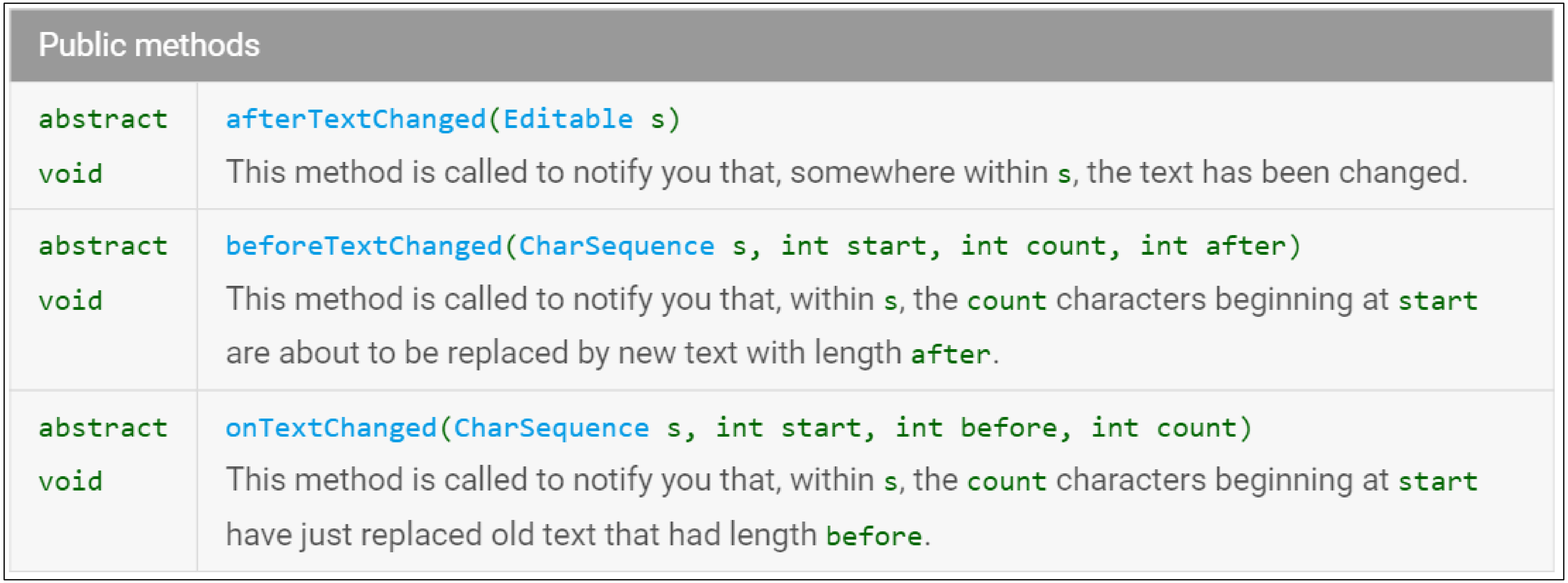
- When changes are detected (a user enters data), the listener will receive the data and transmit it to the model class, Residence.

- As Geolocation is a **EditText** field, you can monitor changes to it by using the **TextWatcher** listener.



52.253456,-7.187162

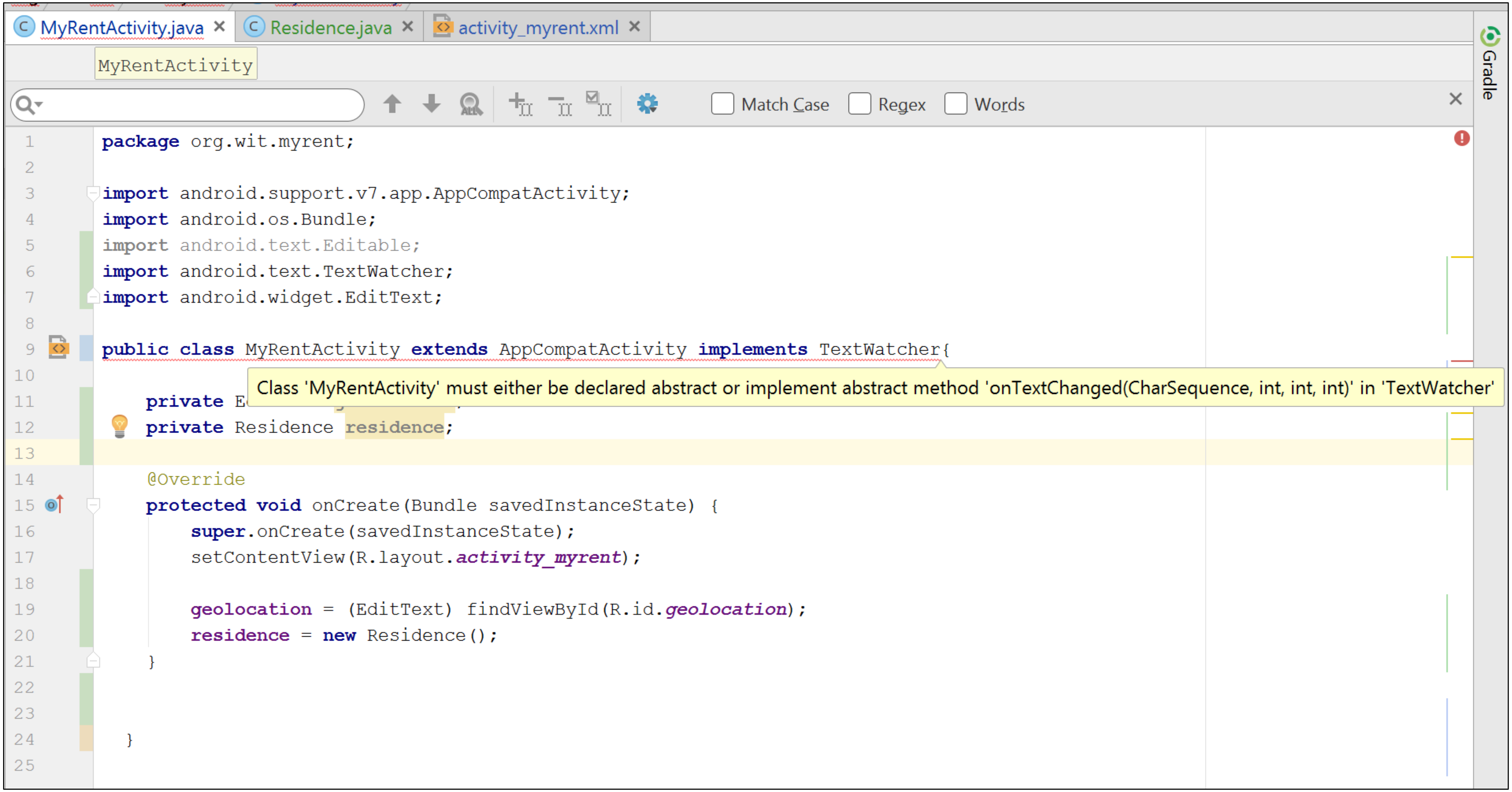
TextWatcher Interface



Public methods	
abstract void	`afterTextChanged(Editable s)` This method is called to notify you that, somewhere within `s`, the text has been changed.
abstract void	`beforeTextChanged(CharSequence s, int start, int count, int after)` This method is called to notify you that, within `s`, the `count` characters beginning at `start` are about to be replaced by new text with length `after`.
abstract void	`onTextChanged(CharSequence s, int start, int before, int count)` This method is called to notify you that, within `s`, the `count` characters beginning at `start` have just replaced old text that had length `before`.

https://developer.android.com/reference/android/text/TextWatcher.html

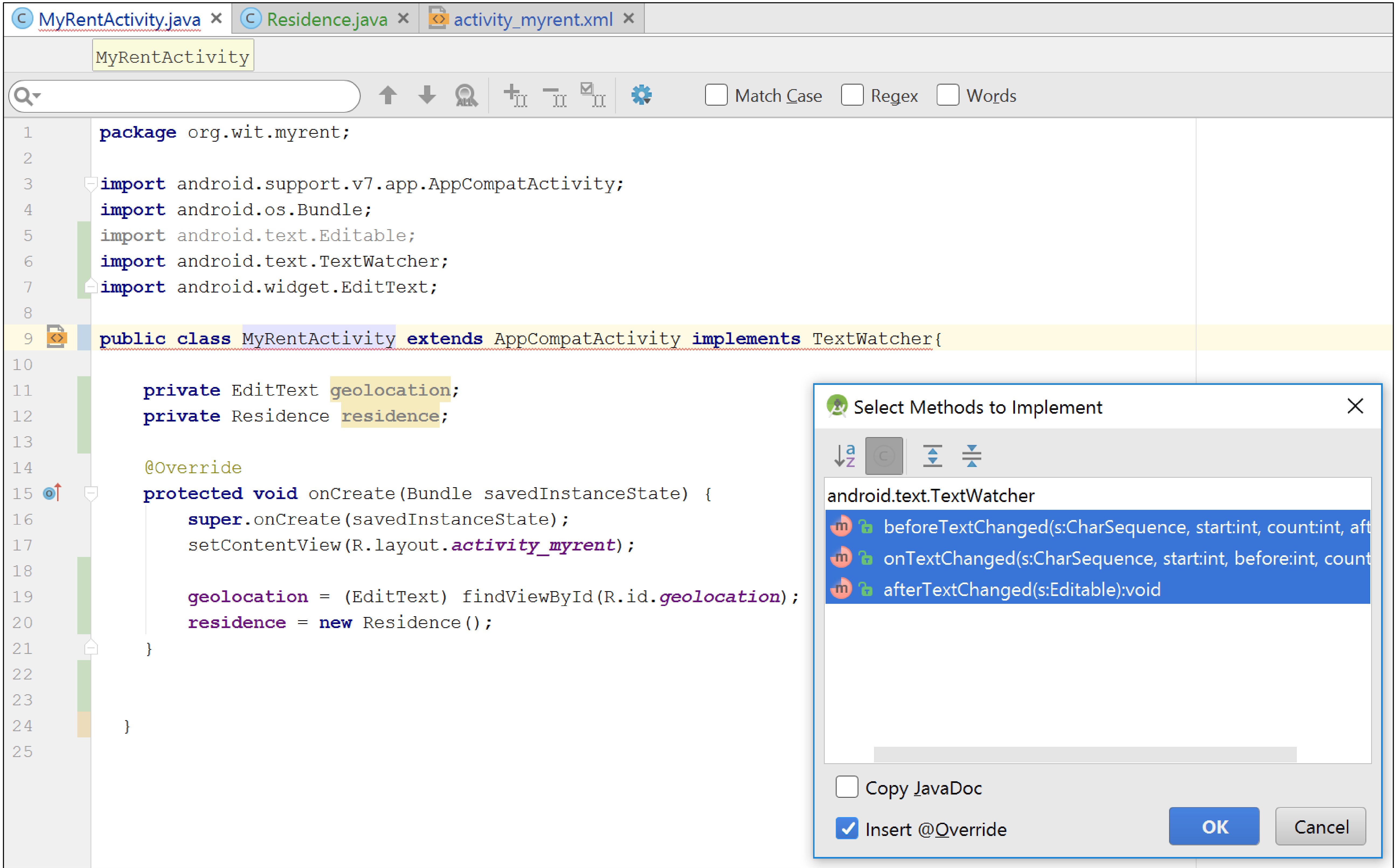
Implementing TextWatcher Listener



```
package org.wit.myrent;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.EditText;

public class MyRentActivity extends AppCompatActivity implements TextWatcher{

    private E
    private Residence residence;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_myrent);

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence = new Residence();
    }


  }

```

Class 'MyRentActivity' must either be declared abstract or implement abstract method 'onTextChanged(CharSequence, int, int, int)' in 'TextWatcher'

Alt+Enter on class header…



```
package org.wit.myrent;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.EditText;

public class MyRentActivity extends AppCompatActivity implements TextWatcher{

    private EditText geolocation;
    private Residence residence;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_myrent);

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence = new Residence();
    }


}
```

Select Methods to Implement

android.text.TextWatcher
- beforeTextChanged(s:CharSequence, start:int, count:int, aft
- onTextChanged(s:CharSequence, start:int, before:int, count
- afterTextChanged(s:Editable):void

□ Copy JavaDoc

☑ Insert @Override

OK   Cancel

Method stubs added…compiler is happy!

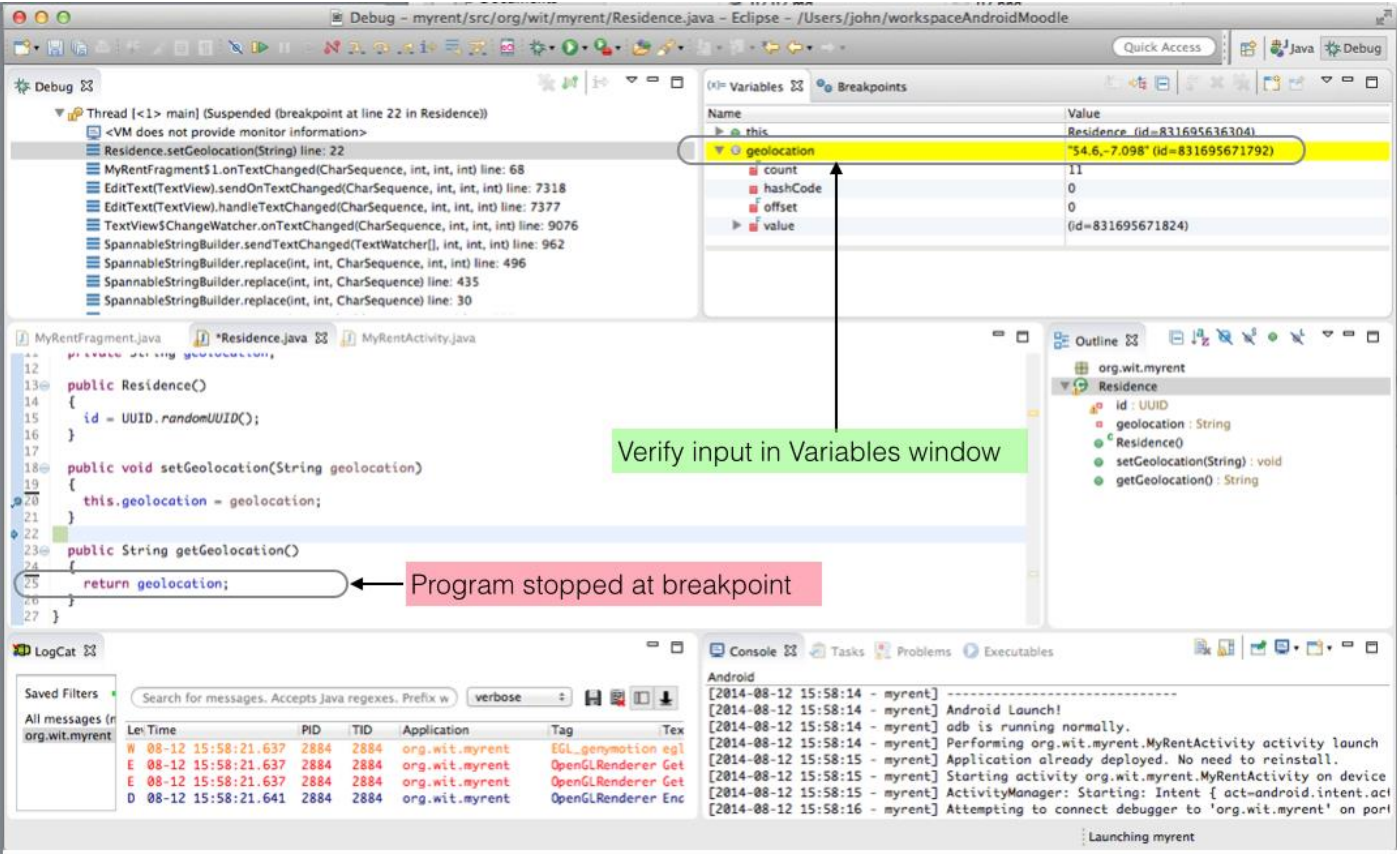
```
public class MyRentActivity extends AppCompatActivity implements TextWatcher{

    private EditText geolocation;
    private Residence residence;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_myrent);

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence = new Residence();
    }


    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {

    }

    @Override
    public void afterTextChanged(Editable s) {

    }
}
```

Completed MyRentActivity

```java
package org.wit.myrent;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.EditText;

public class MyRentActivity extends AppCompatActivity implements TextWatcher{

    private EditText geolocation;
    private Residence residence;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_myrent);

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence = new Residence();
        geolocation.addTextChangedListener(this);
    }

    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

    }

    @Override
    public void afterTextChanged(Editable editable) {
        residence.setGeolocation(editable.toString());
    }
}
```

Register the TextWatcher listener to the geolocation widget.

Testing MyRentActivity using the Debugger



Debug - myrent/src/org/wit/myrent/Residence.java - Eclipse - /Users/john/workspaceAndroidMoodle

```
Thread [<1> main] (Suspended (breakpoint at line 22 in Residence))
    <VM does not provide monitor information>
    Residence.setGeolocation(String) line: 22
    MyRentFragment$1.onTextChanged(CharSequence, int, int, int) line: 68
    EditText(TextView).sendOnTextChanged(CharSequence, int, int, int) line: 7318
    EditText(TextView).handleTextChanged(CharSequence, int, int, int) line: 7377
    TextView$ChangeWatcher.onTextChanged(CharSequence, int, int, int) line: 9076
    SpannableStringBuilder.sendTextChanged(TextWatcher[], int, int, int) line: 962
    SpannableStringBuilder.replace(int, int, CharSequence, int, int) line: 496
    SpannableStringBuilder.replace(int, int, CharSequence) line: 435
    SpannableStringBuilder.replace(int, int, CharSequence) line: 30
```

Name	Value
this	Residence (id=831695636304)
geolocation	"54.6,-7.098" (id=831695671792)
count	11
hashCode	0
offset	0
value	(id=831695671824)

```java
11    private String geolocation;
12
13    public Residence()
14    {
15      id = UUID.randomUUID();
16    }
17
18    public void setGeolocation(String geolocation)
19    {
20      this.geolocation = geolocation;
21    }
22
23    public String getGeolocation()
24    {
25      return geolocation;
26    }
27  }
```

Verify input in Variables window

Program stopped at breakpoint

```
org.wit.myrent
  Residence
    id : UUID
    geolocation : String
    Residence()
    setGeolocation(String) : void
    getGeolocation() : String
```

Saved Filters
All messages (n
org.wit.myrent

Search for messages. Accepts Java regexes. Prefix w   verbose

Le	Time	PID	TID	Application	Tag	Tex
W	08-12 15:58:21.637	2884	2884	org.wit.myrent	EGL_genymotion	egl
E	08-12 15:58:21.637	2884	2884	org.wit.myrent	OpenGLRenderer	Get
E	08-12 15:58:21.637	2884	2884	org.wit.myrent	OpenGLRenderer	Get
D	08-12 15:58:21.641	2884	2884	org.wit.myrent	OpenGLRenderer	Enc

```
Android
[2014-08-12 15:58:14 - myrent] ------------------------------
[2014-08-12 15:58:14 - myrent] Android Launch!
[2014-08-12 15:58:14 - myrent] adb is running normally.
[2014-08-12 15:58:14 - myrent] Performing org.wit.myrent.MyRentActivity activity launch
[2014-08-12 15:58:15 - myrent] Application already deployed. No need to reinstall.
[2014-08-12 15:58:15 - myrent] Starting activity org.wit.myrent.MyRentActivity on device
[2014-08-12 15:58:15 - myrent] ActivityManager: Starting: Intent { act=android.intent.act
[2014-08-12 15:58:16 - myrent] Attempting to connect debugger to 'org.wit.myrent' on port
```

Launching myrent

Questions?