

User Interface

[Previous](#) | Basics and Installation

[Top](#) | Quick Start

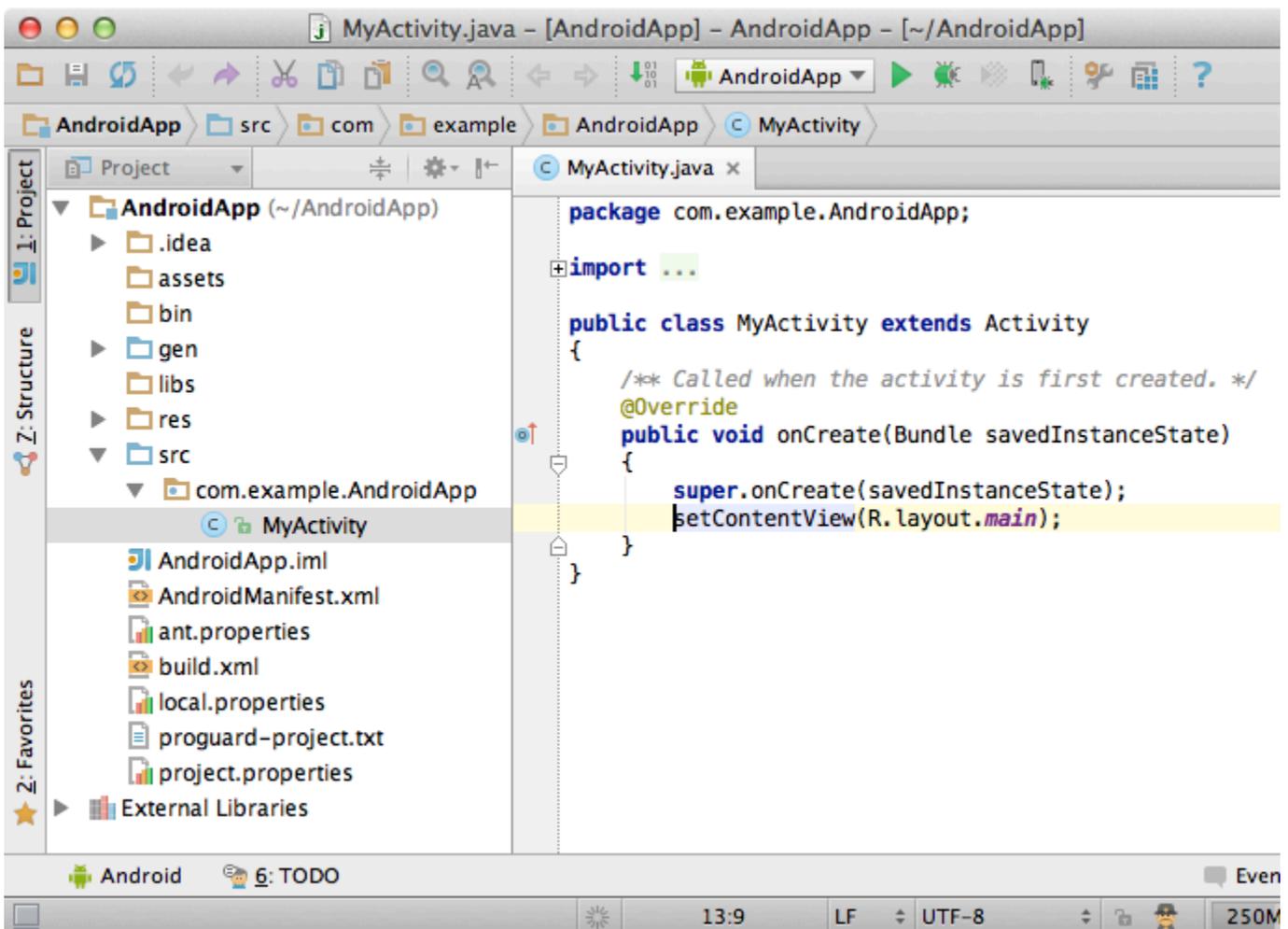
[Next](#) | Editor Basics

 **Redirection Notice**
This page will redirect to <https://www.jetbrains.com/idea/help/guided-tour-around-the-user-interface.html>.

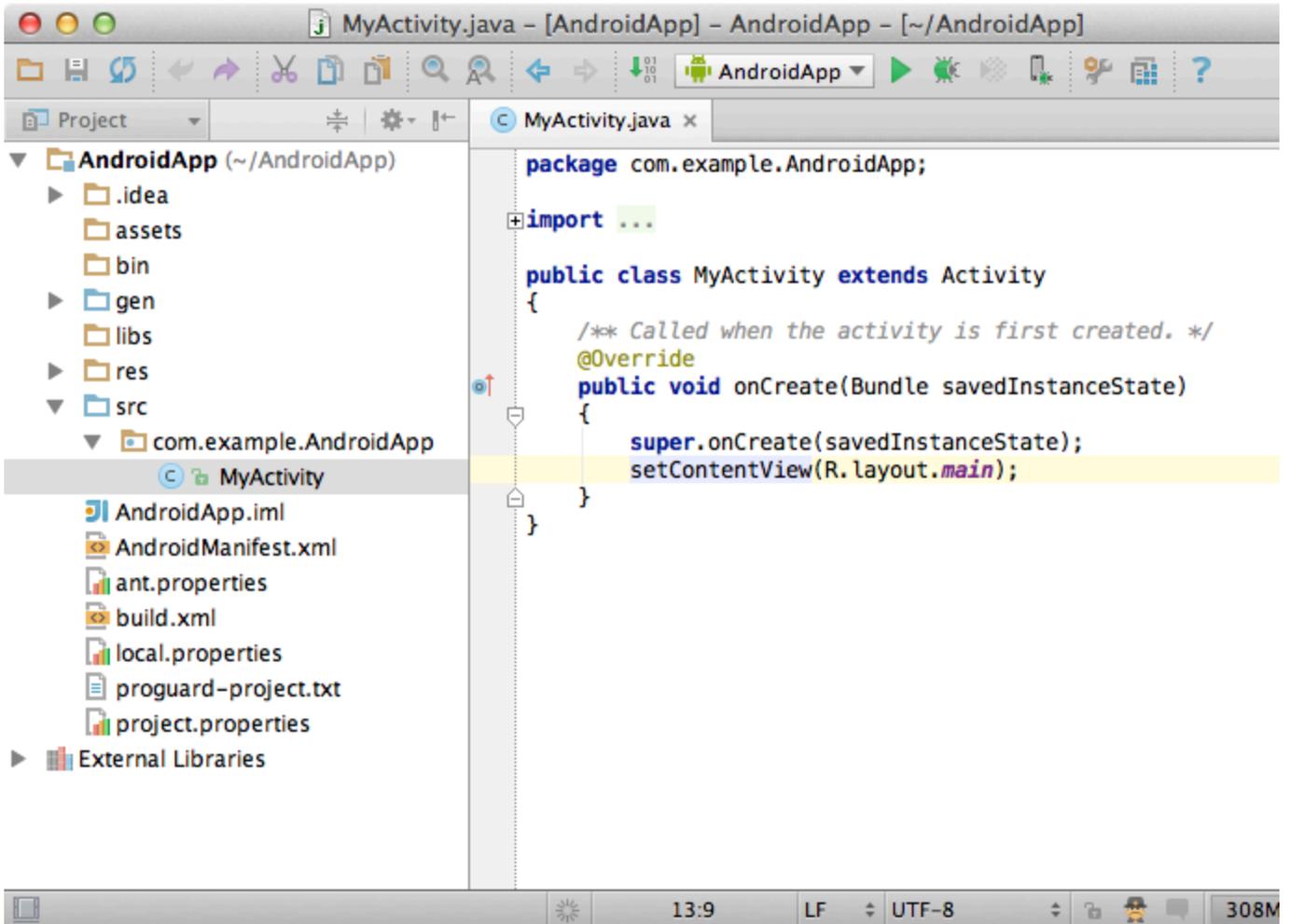
 User interface is not only what you see, but also what you interact with every single day of your coding. That's why it's important to know exactly how it works and what benefits it provides to you to make you more productive.

1. User interface for higher productivity

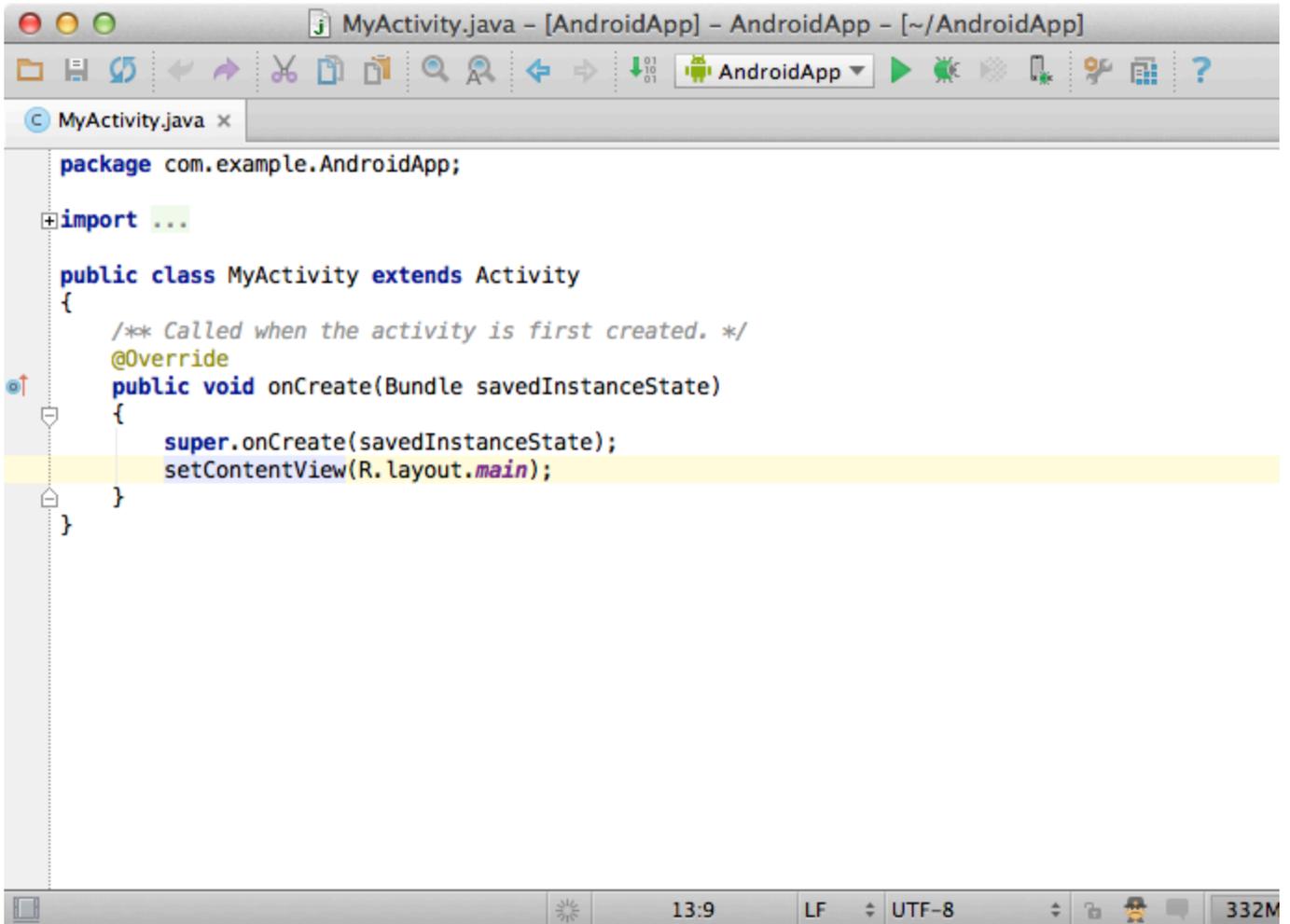
By default IntelliJ IDEA shows you the navigation toolbar, tool window bars and the project view. This is the minimum set of what you can use to navigate over the project files and developer tools.



Some of tool windows have a number in title. You can show or hide a tool window by pressing this number with Alt (Cmd for Mac). The navigation bar can be hidden to bring more space for the editor. Later we will show you how to access it quickly via a shortcut. The tool window bar can be also hidden by clicking the icon in the left bottom corner.



If you want to open a tool window when the tool window bar is hidden you can press `Alt` (or `Cmd` for Mac) twice. When you hide everything you don't need, the IDE looks just like a text editor, a very powerful one at that. You can always maximize the editor by pressing `Shift+Ctrl+F12` (or `Shift+Cmd+F12` for Mac).



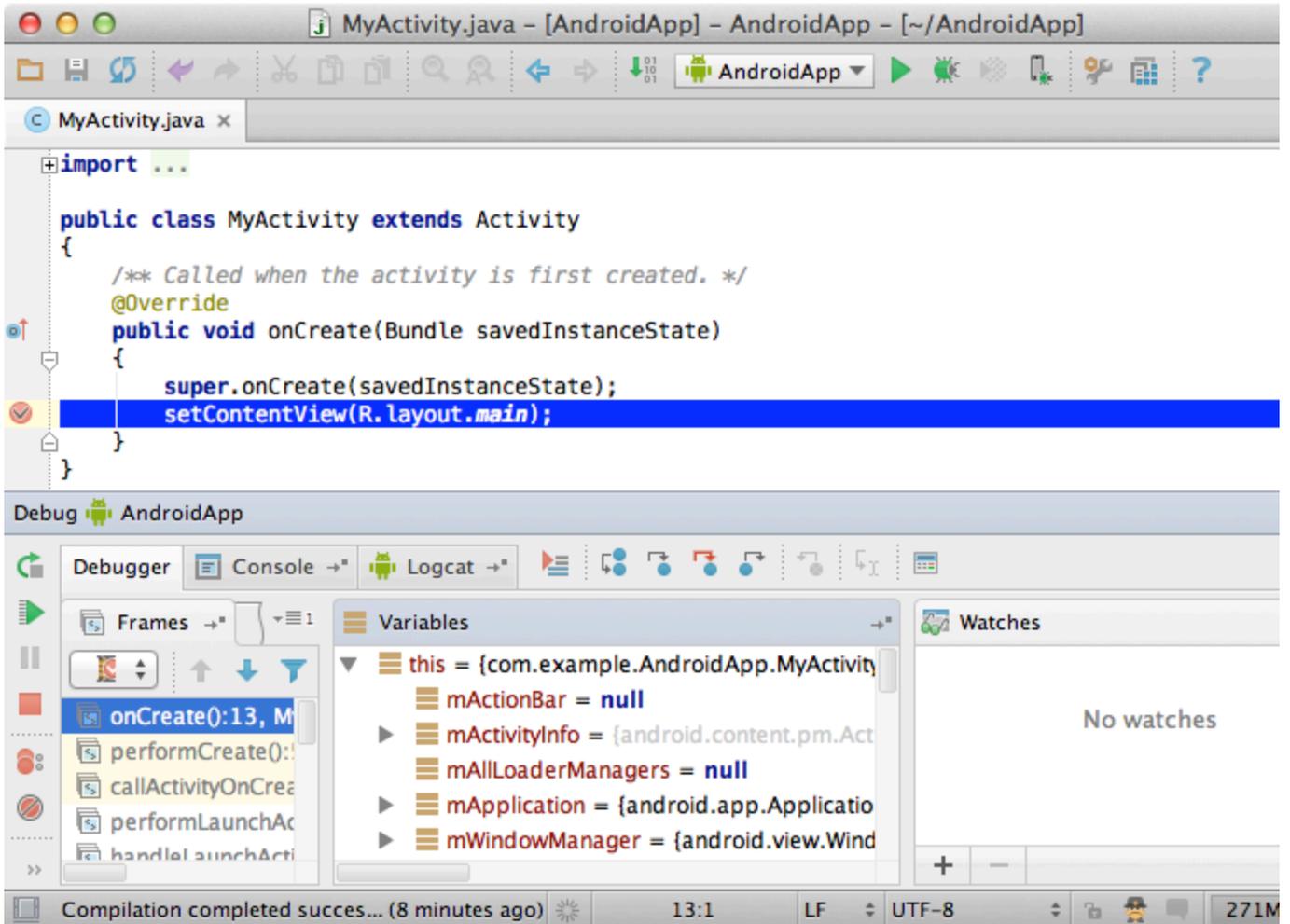
```
package com.example.AndroidApp;

import ...

public class MyActivity extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

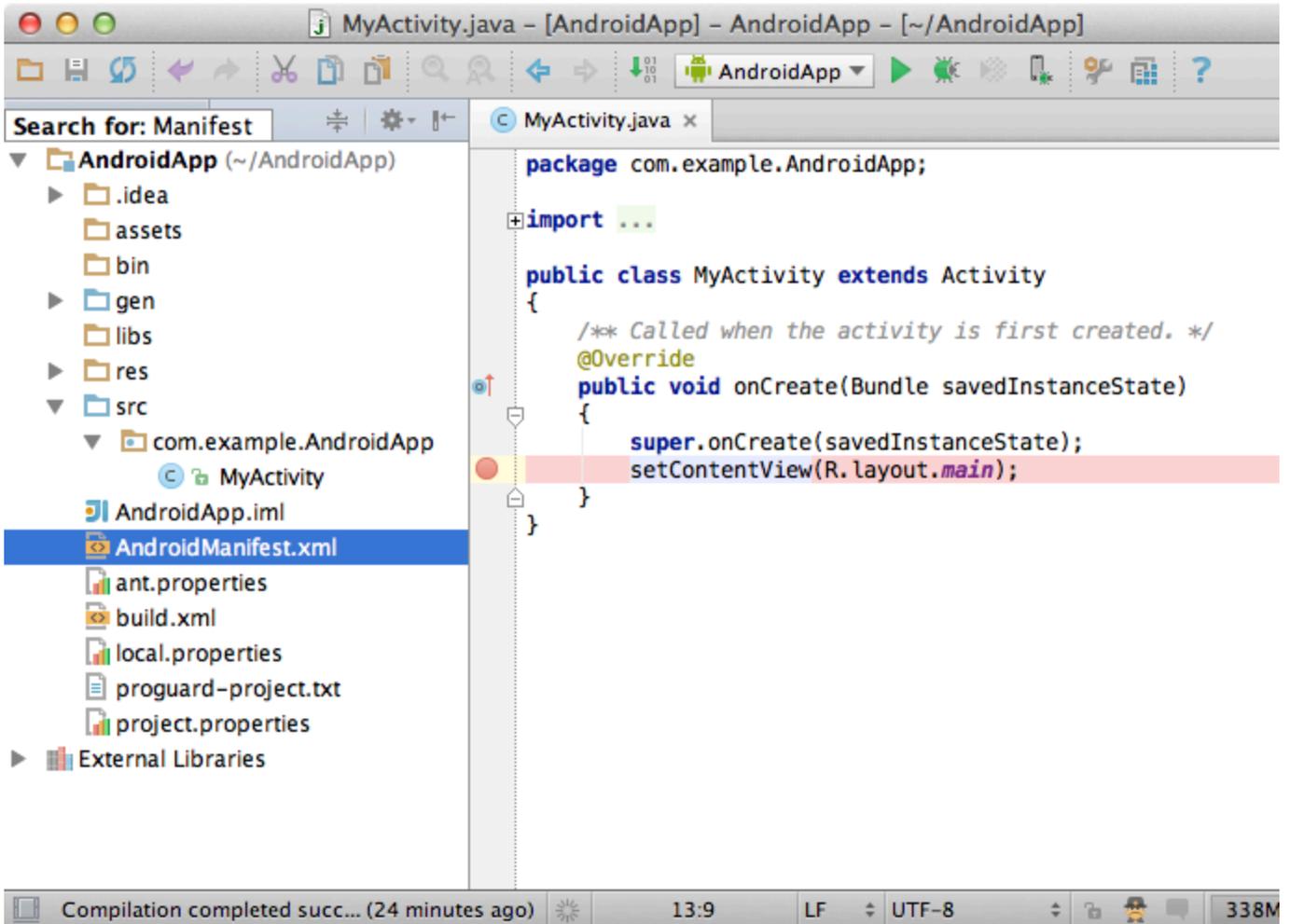
2. No switching between perspectives

IntelliJ IDEA has no perspectives like other IDEs do (e.g. Eclipse). It means that you don't need to switch between different workspace layouts manually to do different tasks. The IDE follows your context and brings up the corresponding tools automatically. Since there are no perspectives, the shortcuts and actions stay the same all the time.



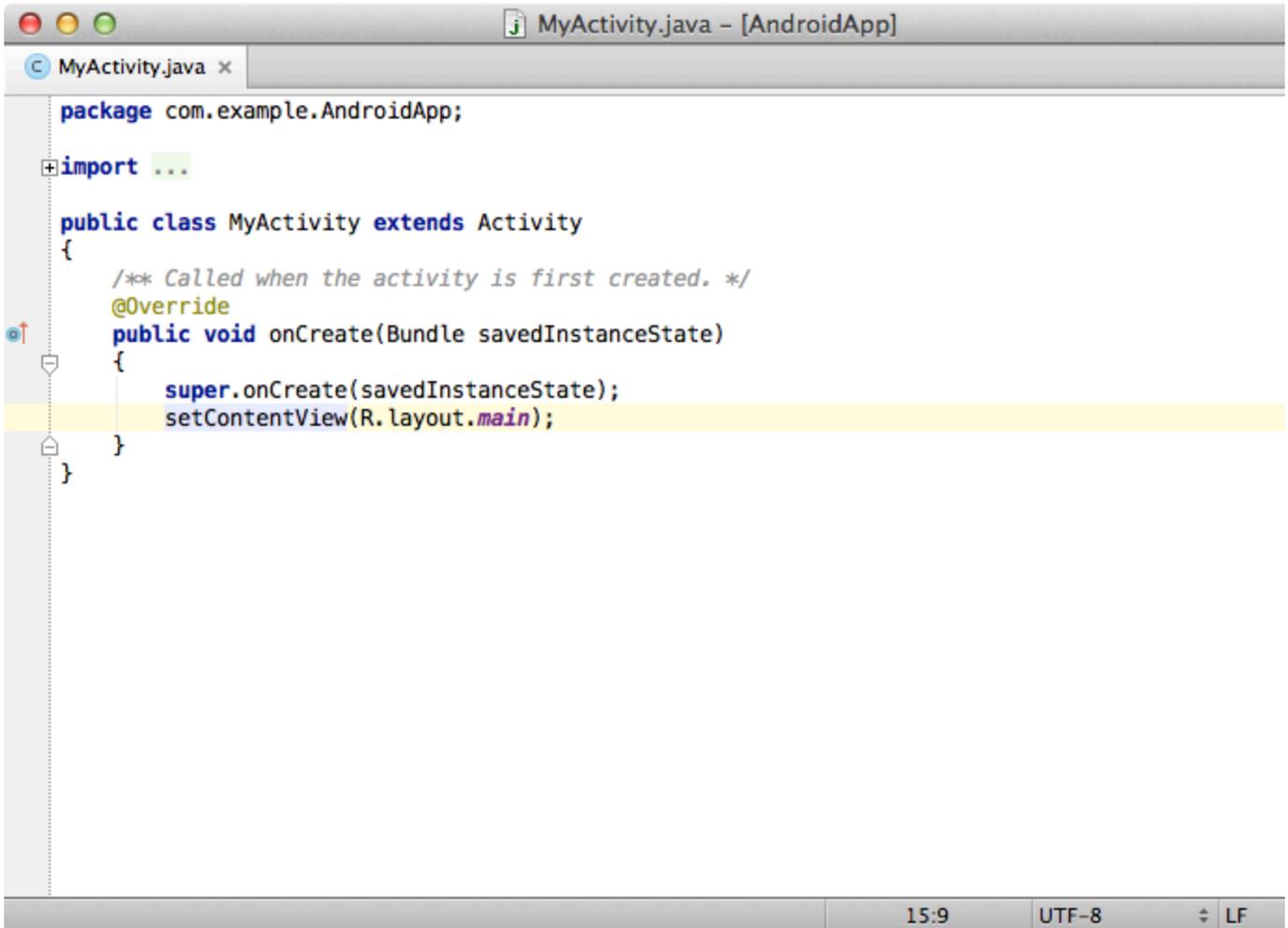
3. Speed search

Most components in IntelliJ IDEA provide speedy search. This feature allows you to filter the list or navigate to a particular item by using a search query. If your focus is in the project view tool window and you would like to navigate to some item without using the mouse — just start typing its name and the IDE will do what you want.



4. Multiple windows

If you want to work with some files in a separate window, just drag the editor tab out of the window and the IDE will detach this tab into a window. You can easily drag your tabs from one window to another.



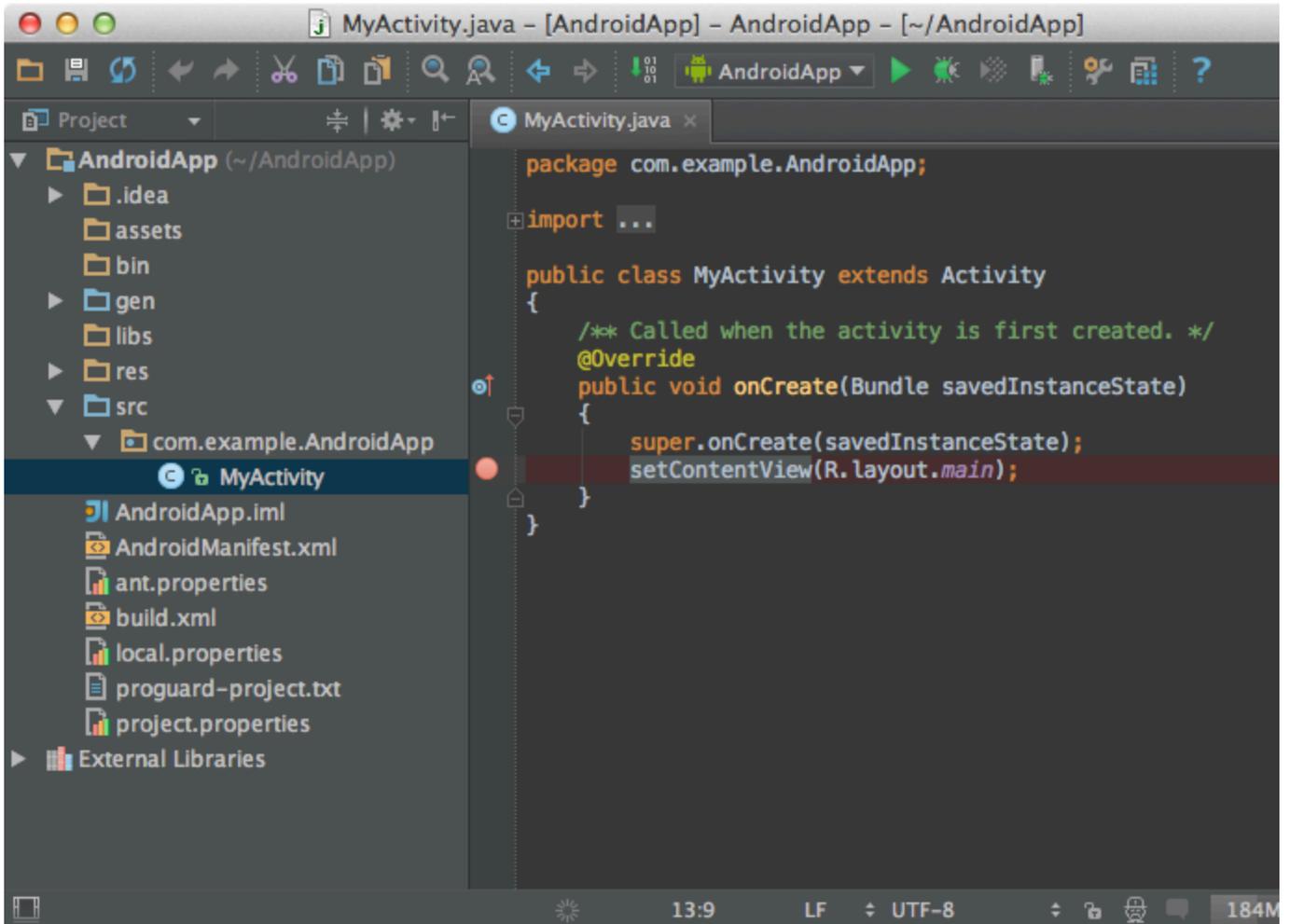
```
package com.example.AndroidApp;

import ...

public class MyActivity extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

5. Darcula color scheme

Since v12, IntelliJ IDEA provides a native darker look and feel called Darcula. You can always change your look and feel in Settings Appearance Theme.



6. Find action by name

If you don't know how to call a particular action, you can find it by name via `Shift+Ctrl+A` (`Shift+Cmd+A` for Mac).

MyActivity.java - [AndroidApp] - AndroidApp - [~/AndroidApp]

MyActivity.java x

```
package com.example.AndroidApp;

import ...

public class MyActivity extends Activity
{
    /** Called w
    @Override
    public void
    {
        super.on
        setConte
    }
}
```

Enter action or option name: Include non-menu actions (⇧⌘A)

Make

- Make Project (⌘F9) Build
- Invert Boolean... Refactoring Actions (2)
- Make Module Build
- Make Static... Refactoring Actions (1)
- Implicit Conversion (⌘↵) Goto by Reference Actions
- Google App Engine Integration Settings
- Make Call Chain Into Call Sequence Settings
- Make generated local variables final Settings
- Make generated parameters final Settings
- Make Package-Local Settings

15:9 LF UTF-8 152M