COMP 90018 Mobile Computing Systems Programming

Tutorial on Android Development

Chu Luo, Eman Bin Khunayn

{chu.luo, eman.bin}@unimelb.edu.au

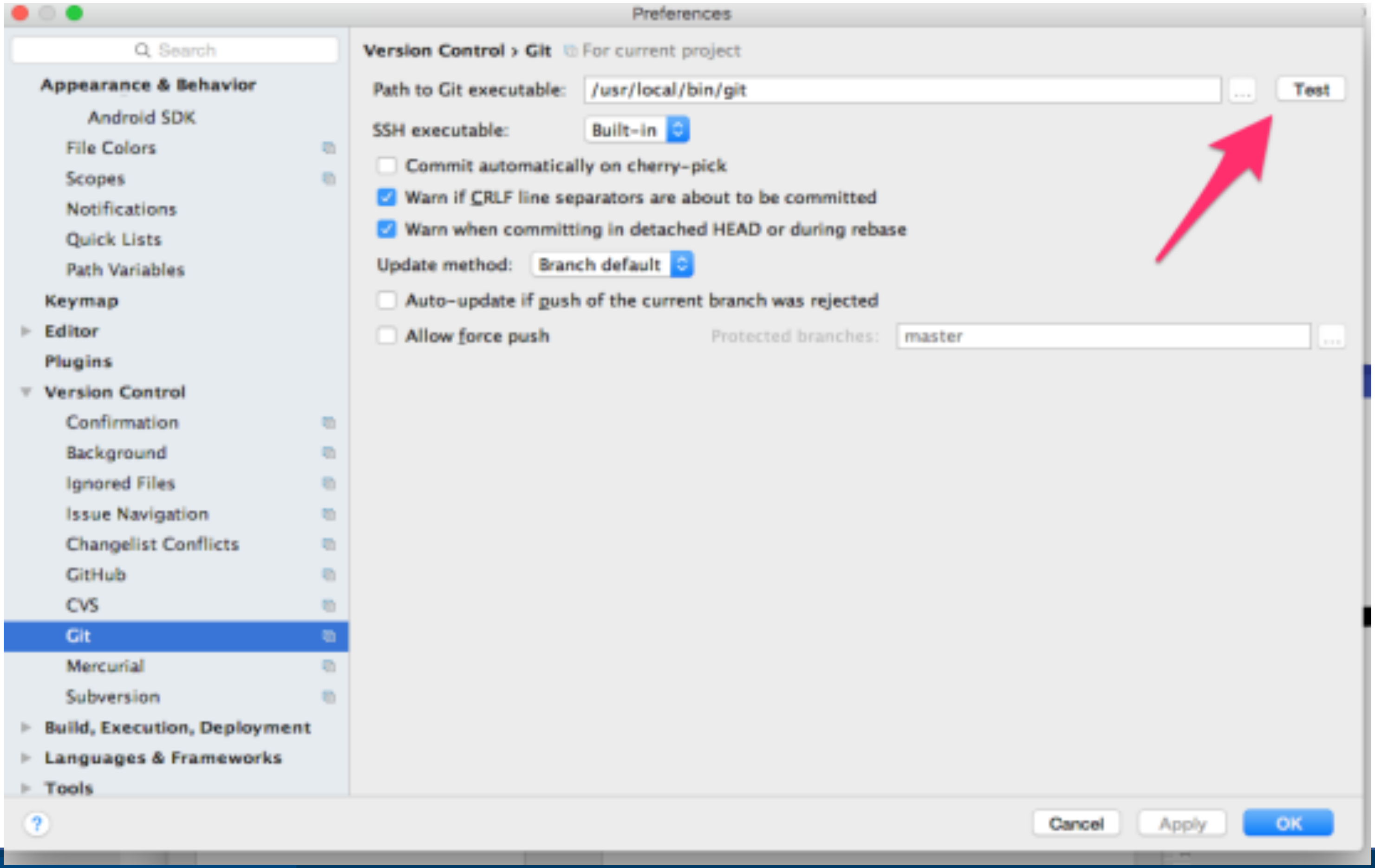
Welcome!

Outcomes of this tutorial:

1. Learn to use – Github (via Android Studio)
2. Android UI Design and Control
3. Background Tasks

Github with Android Studio

Install git and create a GitHub account, then connect GitHub account to Android Studio.



Version Control › Git For current project

Path to Git executable: /usr/local/bin/git

SSH executable: Built-in

☐ Commit automatically on cherry-pick

☑ Warn if CRLF line separators are about to be committed

☑ Warn when committing in detached HEAD or during rebase

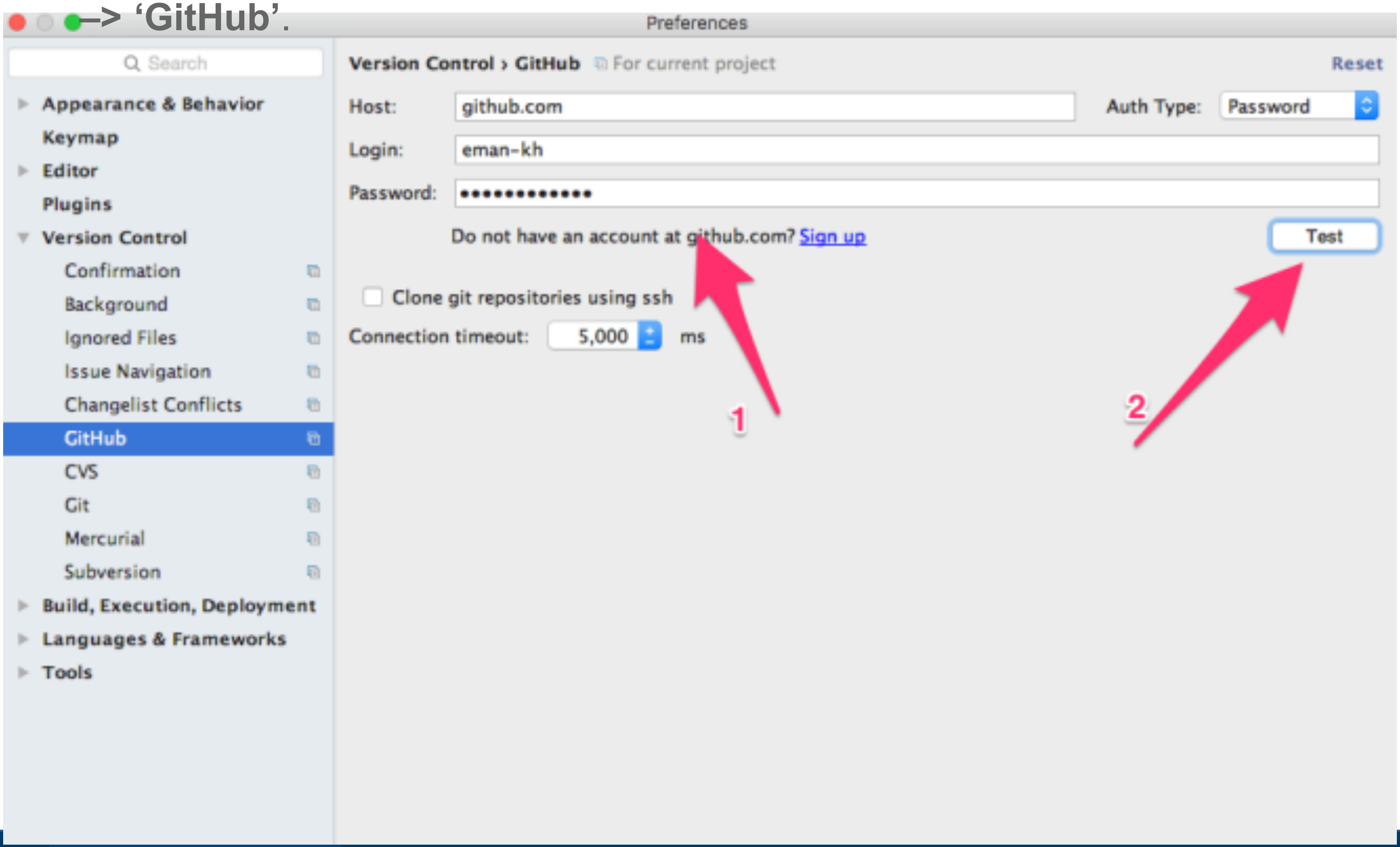
Update method: Branch default

☐ Auto-update if push of the current branch was rejected

☐ Allow force push Protected branches: master

THE UNIVERSITY OF MELBOURNE

Login to your GitHub Account, go to **‘Preferences’ –> ‘Version Control’ –> ‘GitHub’.**


Preferences

Version Control › GitHub For current project Reset

Host: github.com Auth Type: Password

Login: eman-kh

Password: •••••••••••••

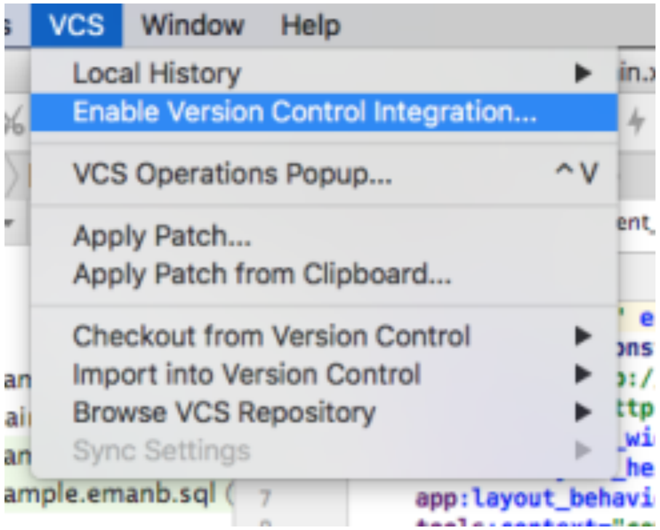
Do not have an account at github.com? Sign up Test

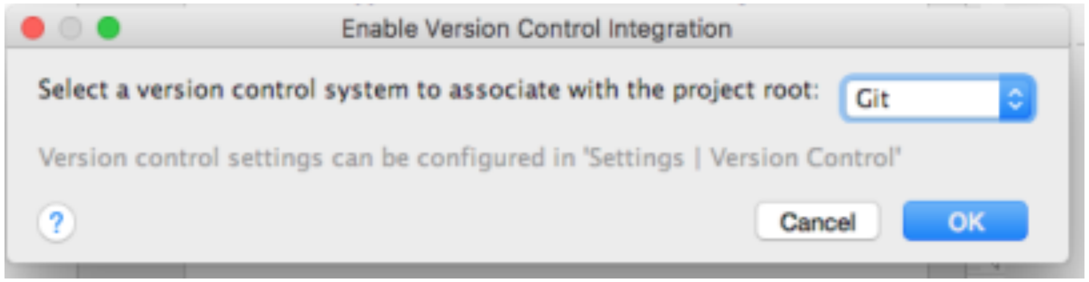
☐ Clone git repositories using ssh

Connection timeout: 5,000 ms

Enable Version Control Integration

Select **'VCS' –> 'Enable Version Control Integration…'** to enable version control for the current project. Then, choose the version control system for your project.


VCS
Window
Help
Local History
Enable Version Control Integration...
VCS Operations Popup... ^V
Apply Patch...
Apply Patch from Clipboard...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Sync Settings

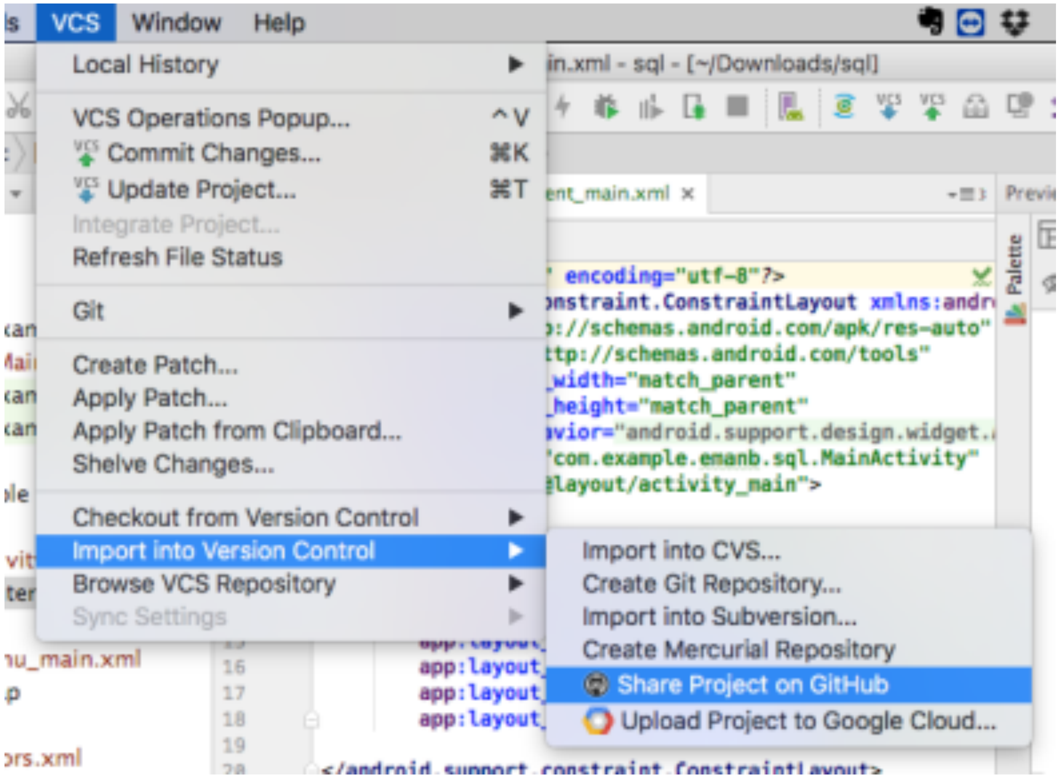


Select a version control system to associate with the project root: Git

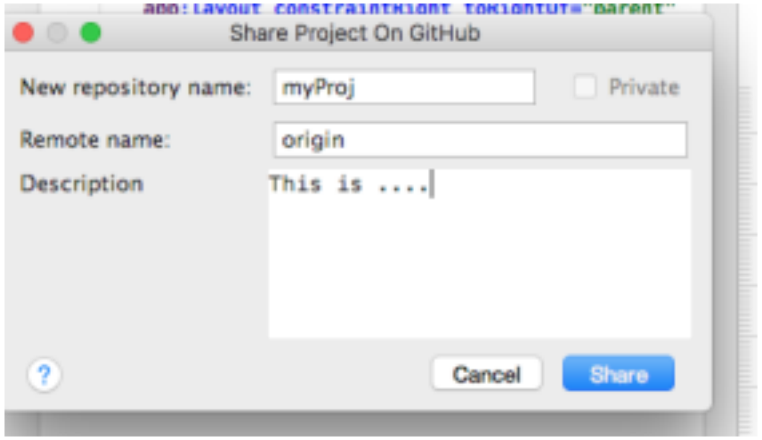
Version control settings can be configured in 'Settings | Version Control'

Share project on your VCS

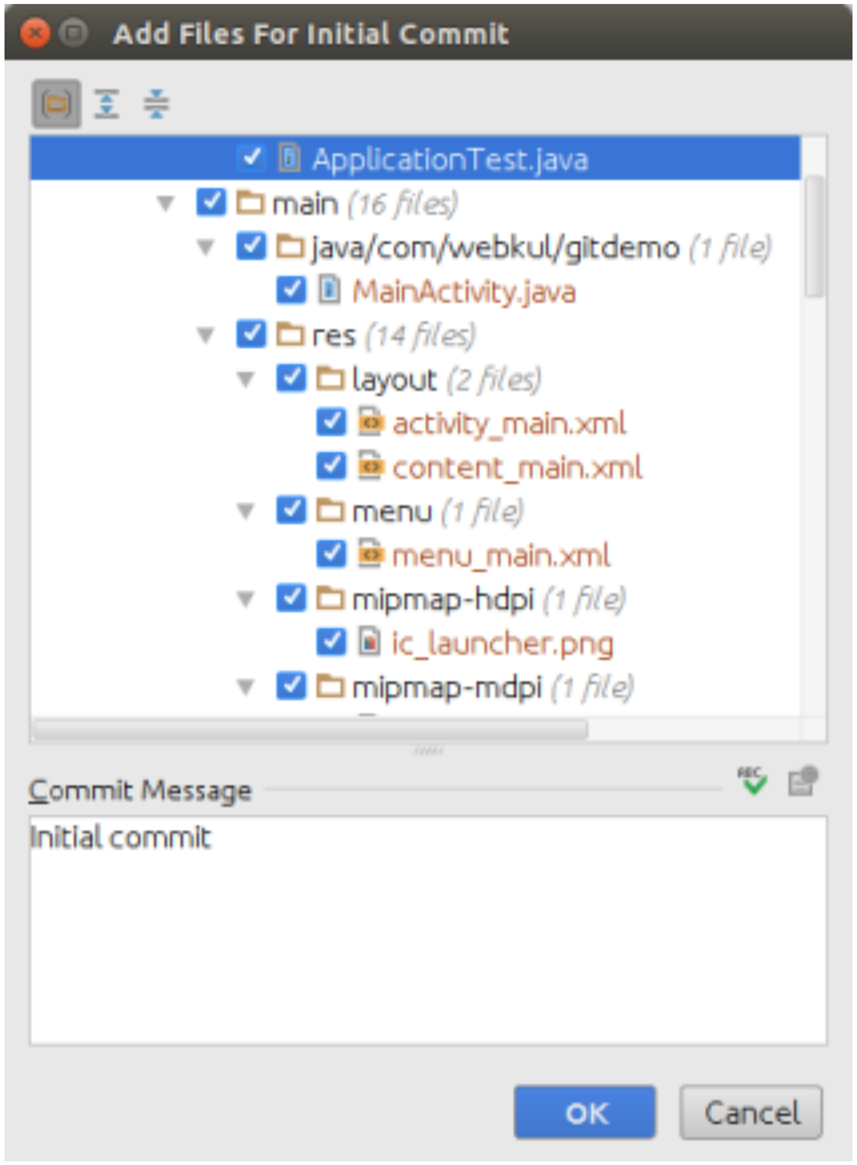
Go to **'VCS' –> 'Import into Version Control' – > 'Share Project on GitHub'** for creating a repository on GitHub.
Choose a repository name for your project.
An automated dialog will allow you to add files for the initial commit.



```
' encoding="utf-8"?>
onstraint.ConstraintLayout xmlns:andr
p://schemas.android.com/apk/res-auto"
ttp://schemas.android.com/tools"
_width="match_parent"
_height="match_parent"
avior="android.support.design.widget.
'com.example.emanb.sql.MainActivity"
@layout/activity_main">
```



New repository name: myProj
Remote name: origin
Description: This is ....


Add Files For Initial Commit

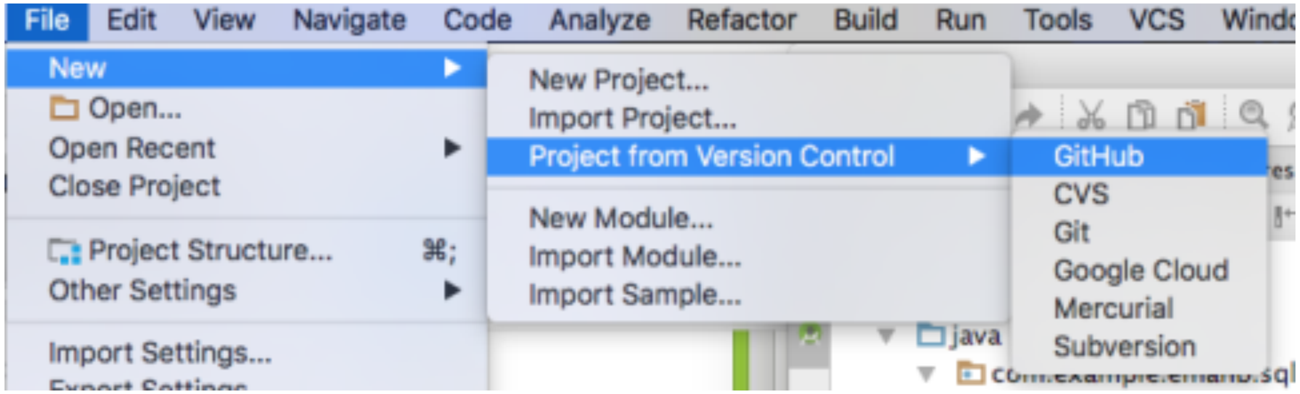
```
✓ ApplicationTest.java
▼ ✓ main (16 files)
  ▼ ✓ java/com/webkul/gitdemo (1 file)
      ✓ MainActivity.java
  ▼ ✓ res (14 files)
    ▼ ✓ layout (2 files)
        ✓ activity_main.xml
        ✓ content_main.xml
    ▼ ✓ menu (1 file)
        ✓ menu_main.xml
    ▼ ✓ mipmap-hdpi (1 file)
        ✓ ic_launcher.png
    ▼ ✓ mipmap-mdpi (1 file)
```

Commit Message

Initial commit

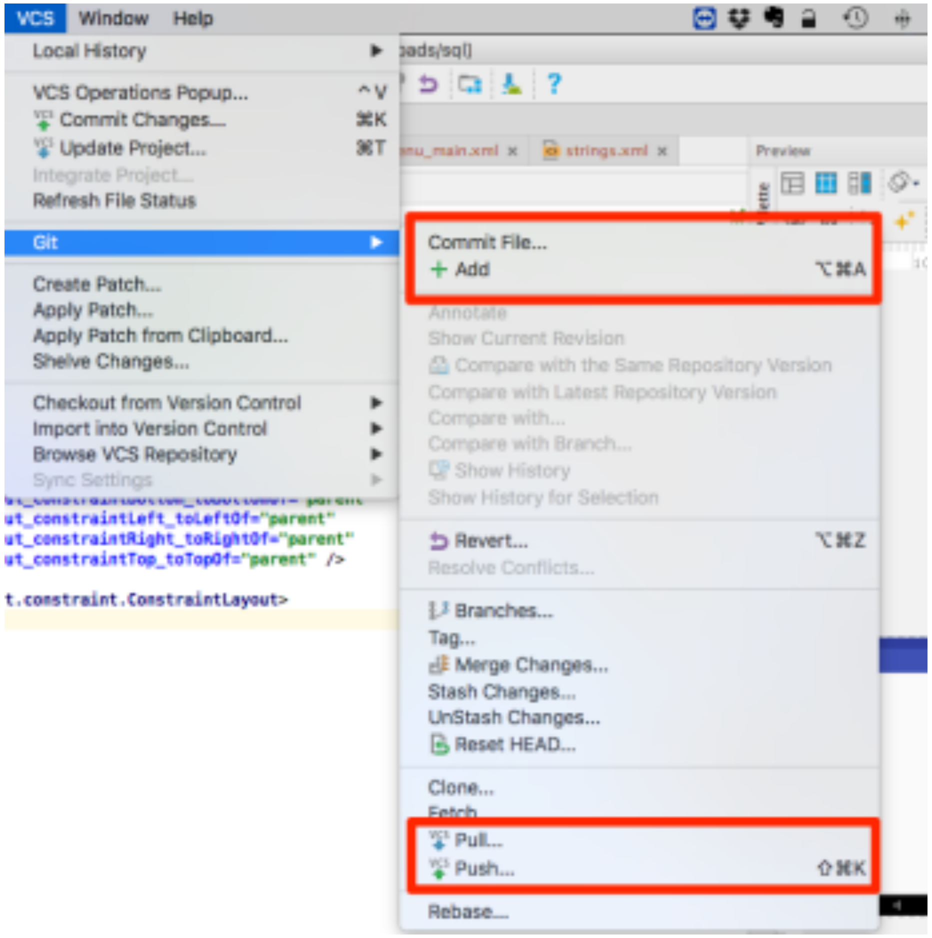
Clicking **Share** performs a git commit to do an initial local commit, and then a git push to push those contents to the remote repository that you created.

Clone an existed project from GitHub



Go to the GitHub page and get the HTTPs path to your repository. Add it.
It looks like: https://github.com/*.git
Then enter your github username and password.Select the repository and hit **clone.**

Git commands: add, commit, push, pull/Fetch, merge, branch, checkout.



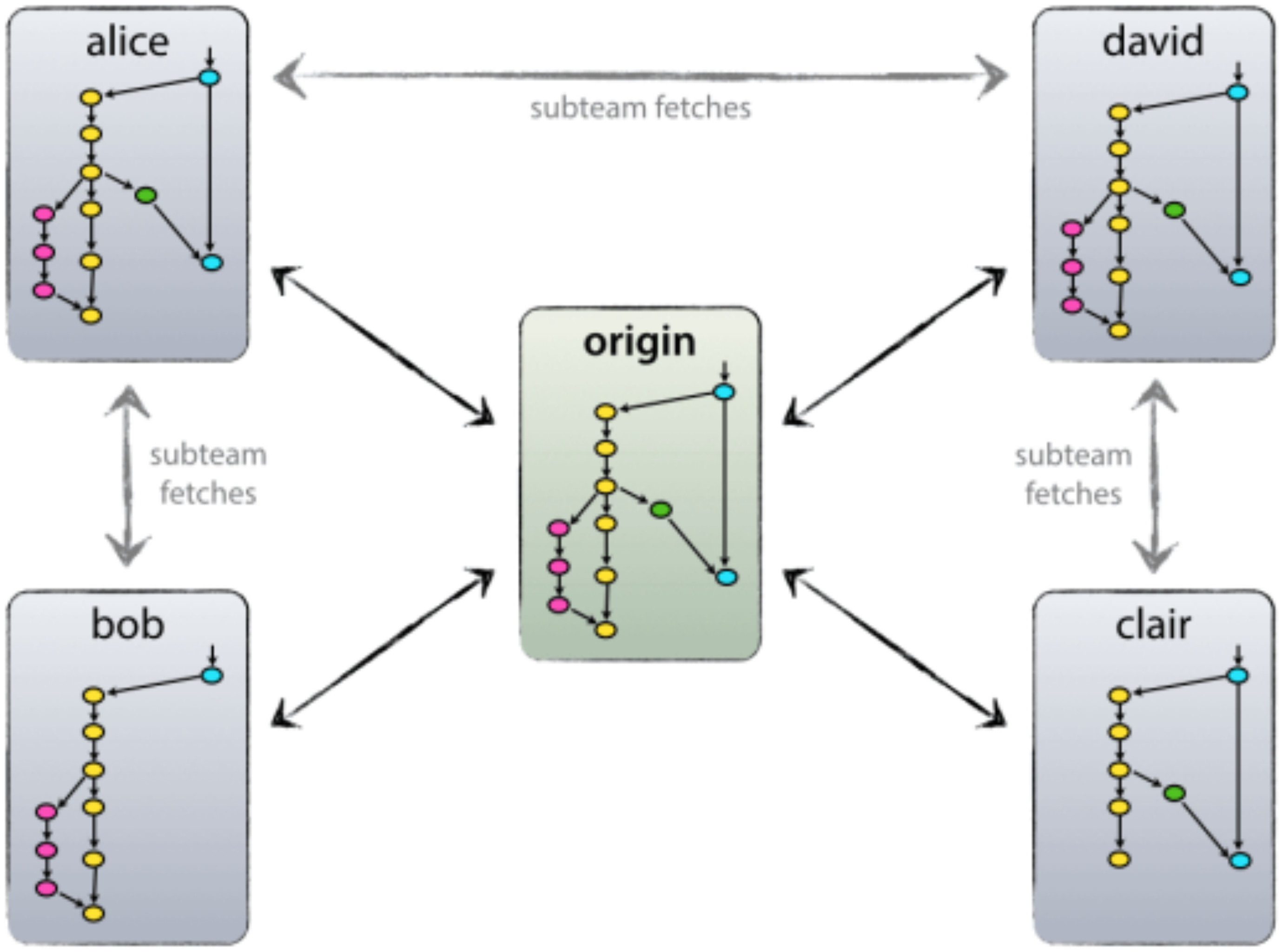
```
ut_constraintLeft_toLeftOf="parent"
ut_constraintRight_toRightOf="parent"
ut_constraintTop_toTopOf="parent" />

t.constraint.ConstraintLayout>
```

Important

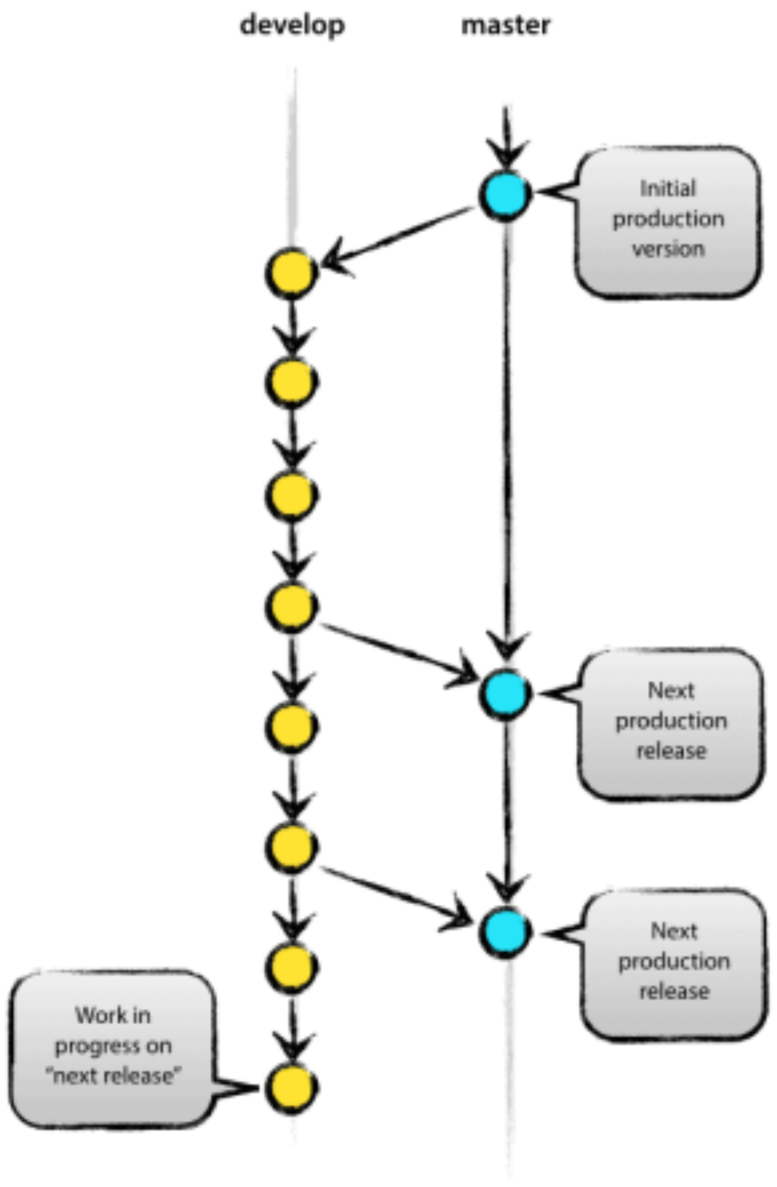
Never push changes to remote before you fetch and merge and resolve the conflict (manually if needed).

Git branching model

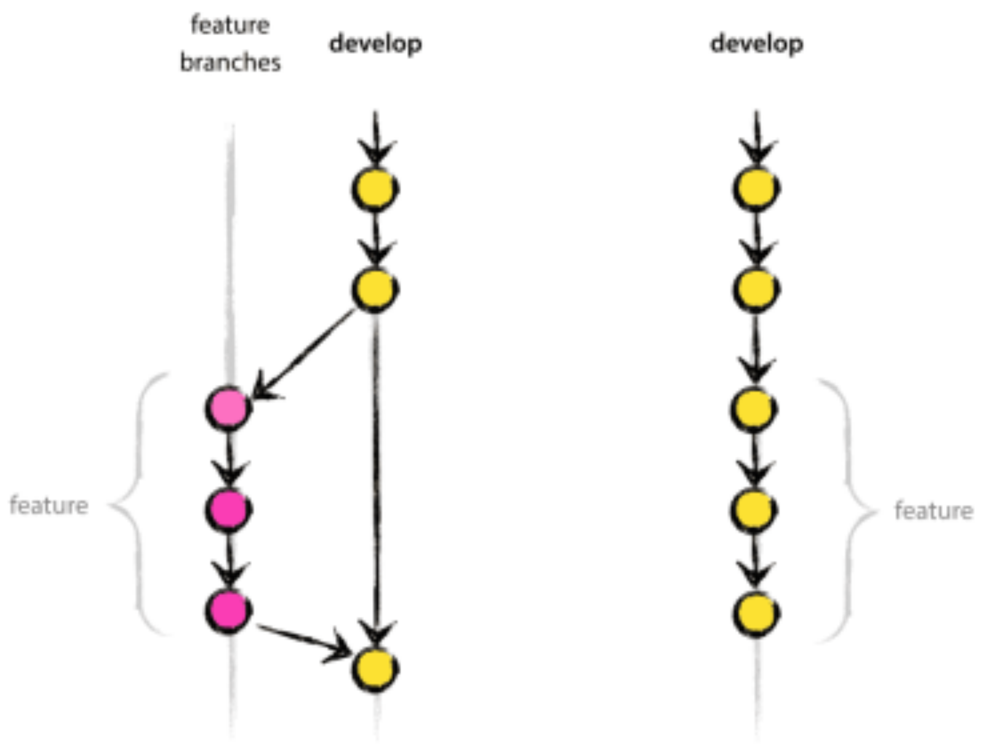


THE UNIVERSITY OF MELBOURNE

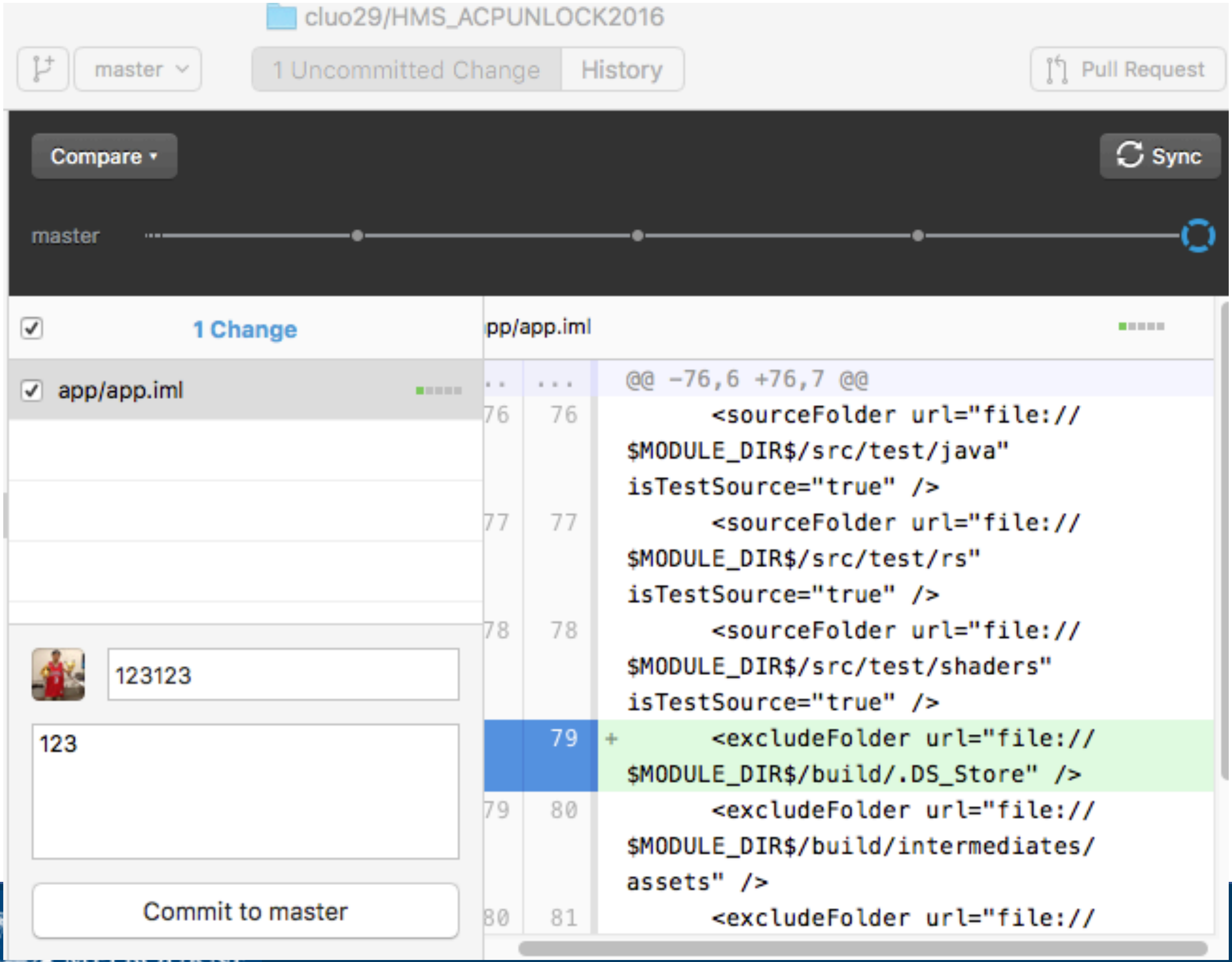
main branches


develop
master
Initial production version
Next production release
Next production release
Work in progress on "next release"

Supporting branches



Don’t like Android Studio? Just use Github directly!



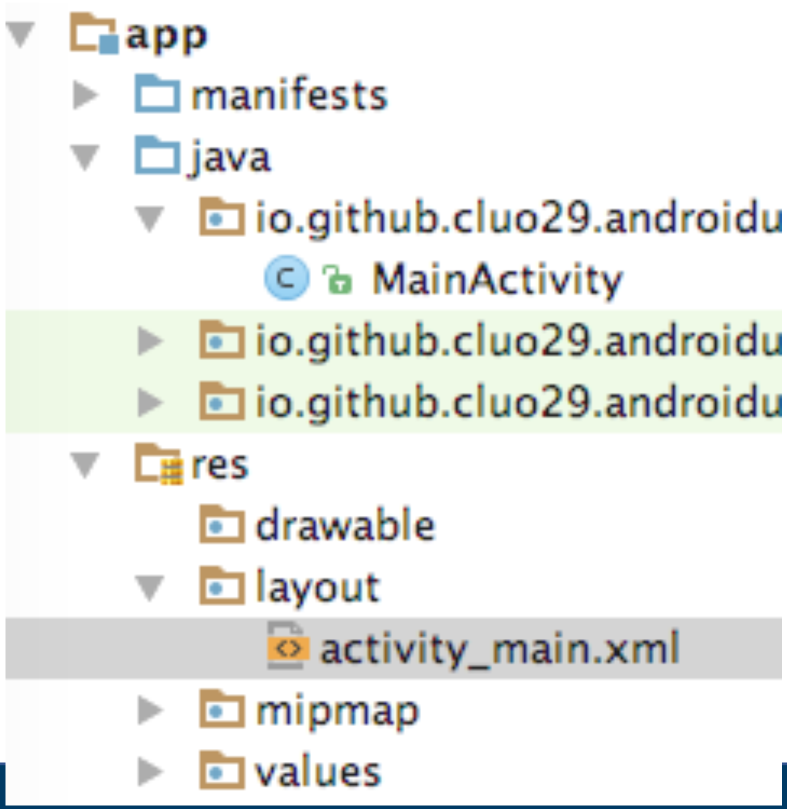
```
@@ -76,6 +76,7 @@
        <sourceFolder url="file://
$MODULE_DIR$/src/test/java"
isTestSource="true" />
        <sourceFolder url="file://
$MODULE_DIR$/src/test/rs"
isTestSource="true" />
        <sourceFolder url="file://
$MODULE_DIR$/src/test/shaders"
isTestSource="true" />
+       <excludeFolder url="file://
$MODULE_DIR$/build/.DS_Store" />
        <excludeFolder url="file://
$MODULE_DIR$/build/intermediates/
assets" />
        <excludeFolder url="file://
```

IMPORTANT!

1. Don’t rely on Github only. It may have bugs or server crashes.
2. Duplicate your files (every version) on your disk, cloud storage (e.g., Google Drive)

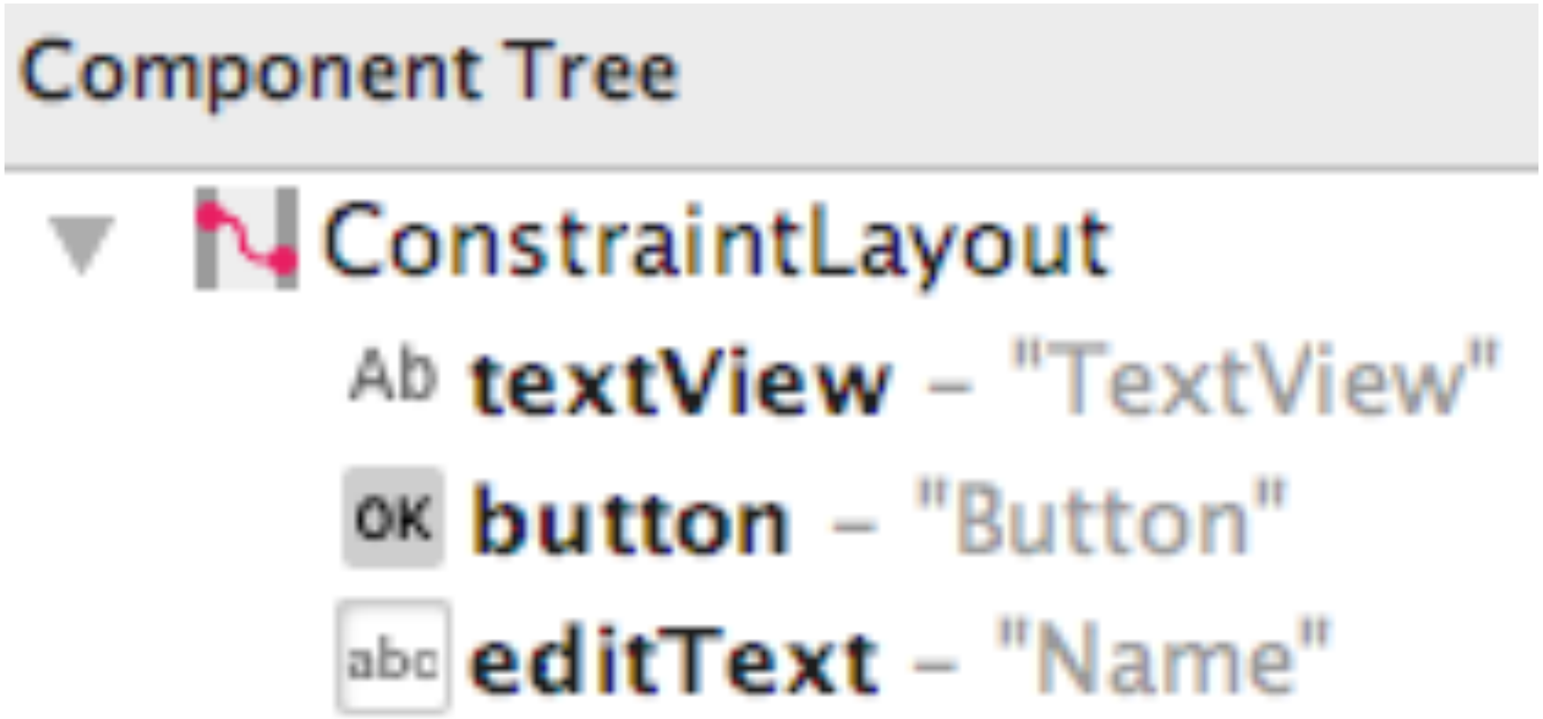
Android UI

Described by XML Files



```
app
  manifests
  java
    io.github.cluo29.androidu
      MainActivity
    io.github.cluo29.androidu
    io.github.cluo29.androidu
  res
    drawable
    layout
      activity_main.xml
    mipmap
    values
```

In UI: Elements under Layout E.g., ConstraintLayout


Component Tree

ConstraintLayout

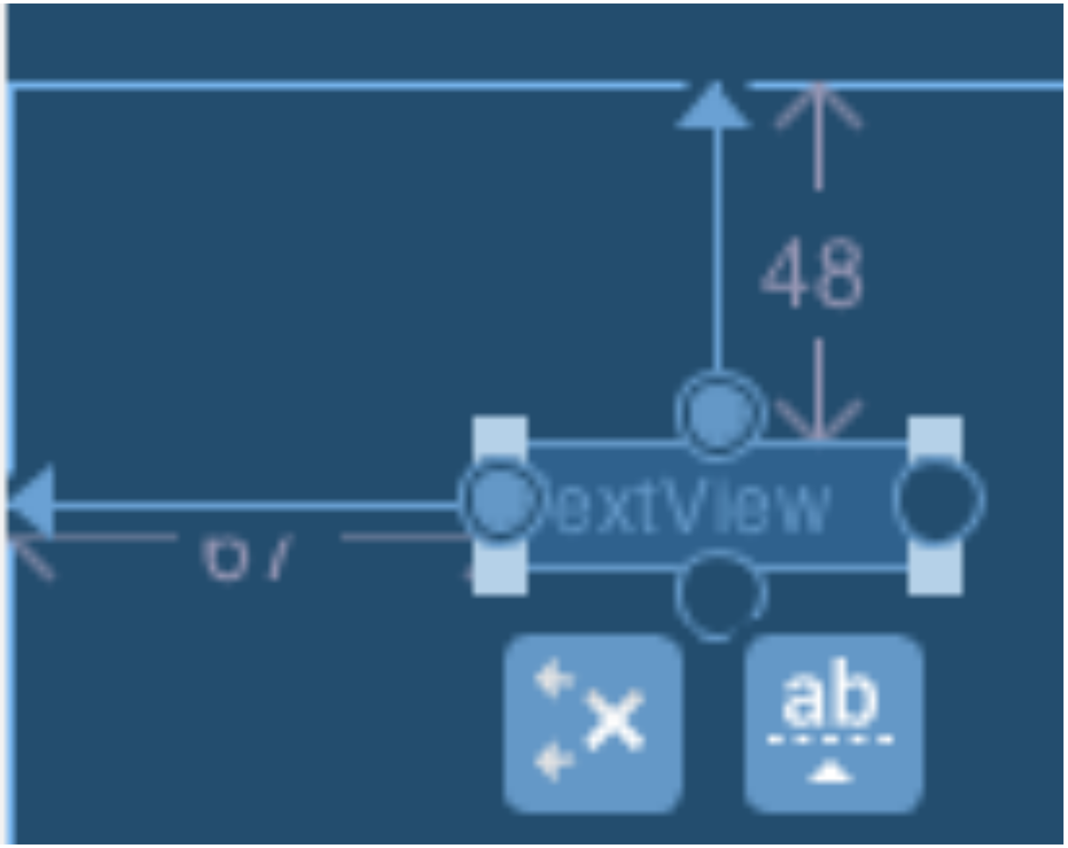
textView – "TextView"

button – "Button"

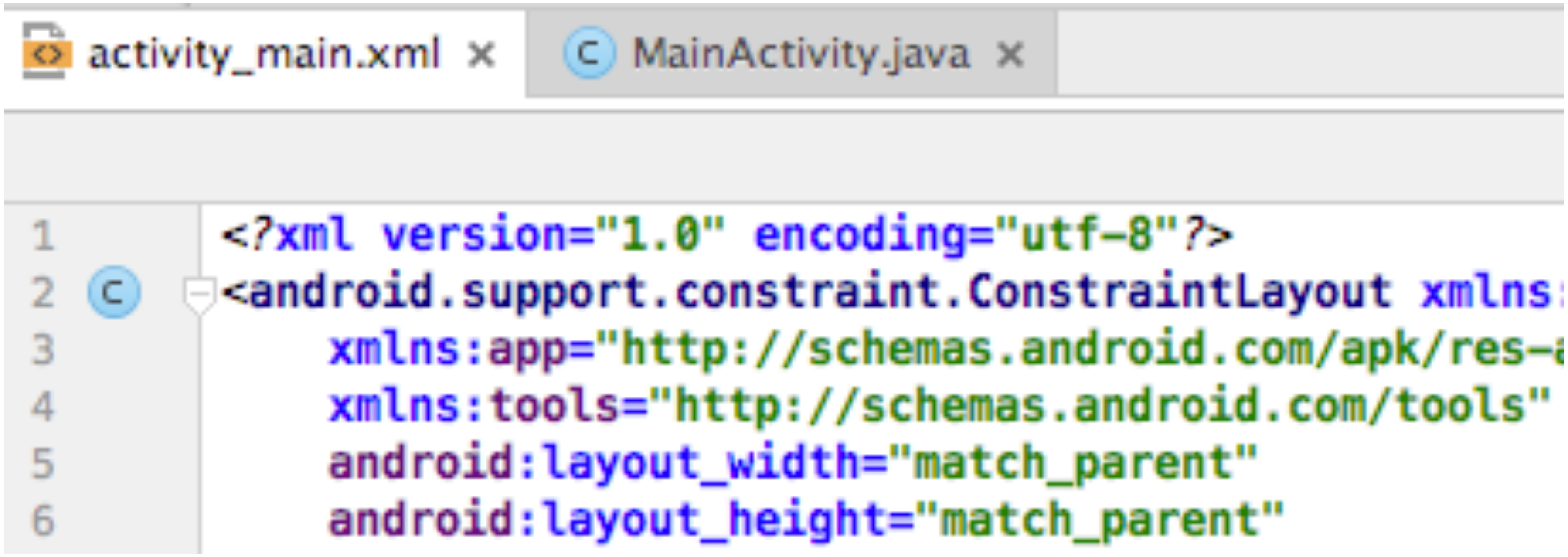
editText – "Name"

Google Tutorials https://developer.android.com/training/constraint-layout/index.html#add-constraintlayout-to-your-project

Change Position using Blue thingy

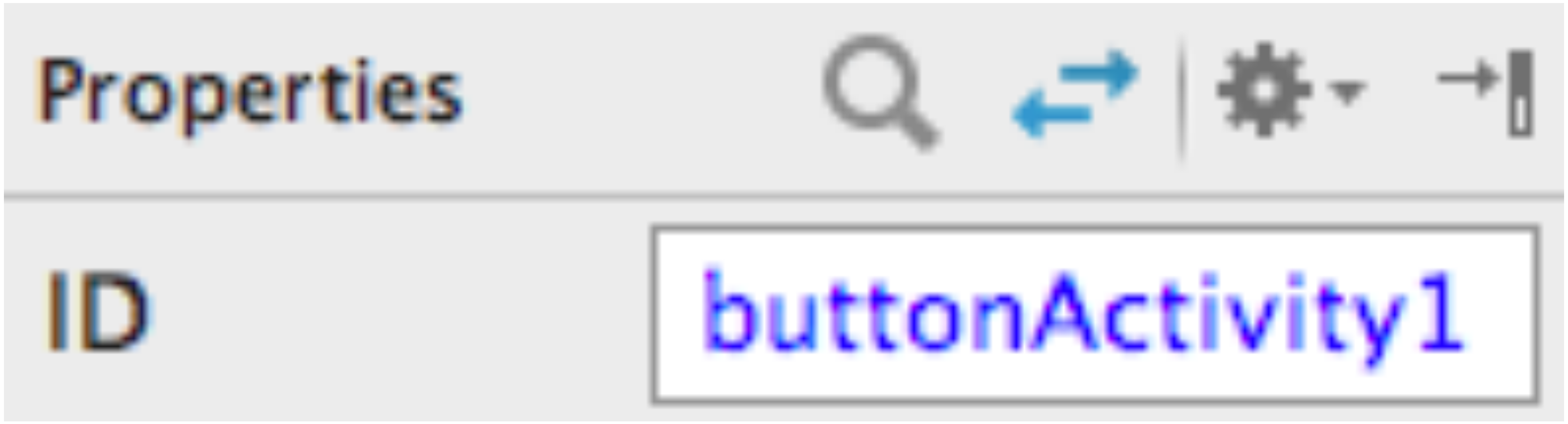


Alternatively, write XML directly



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:
    xmlns:app="http://schemas.android.com/apk/res-a
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

Still, give a good ID for each item



Properties

ID	buttonActivity1

Exercise:

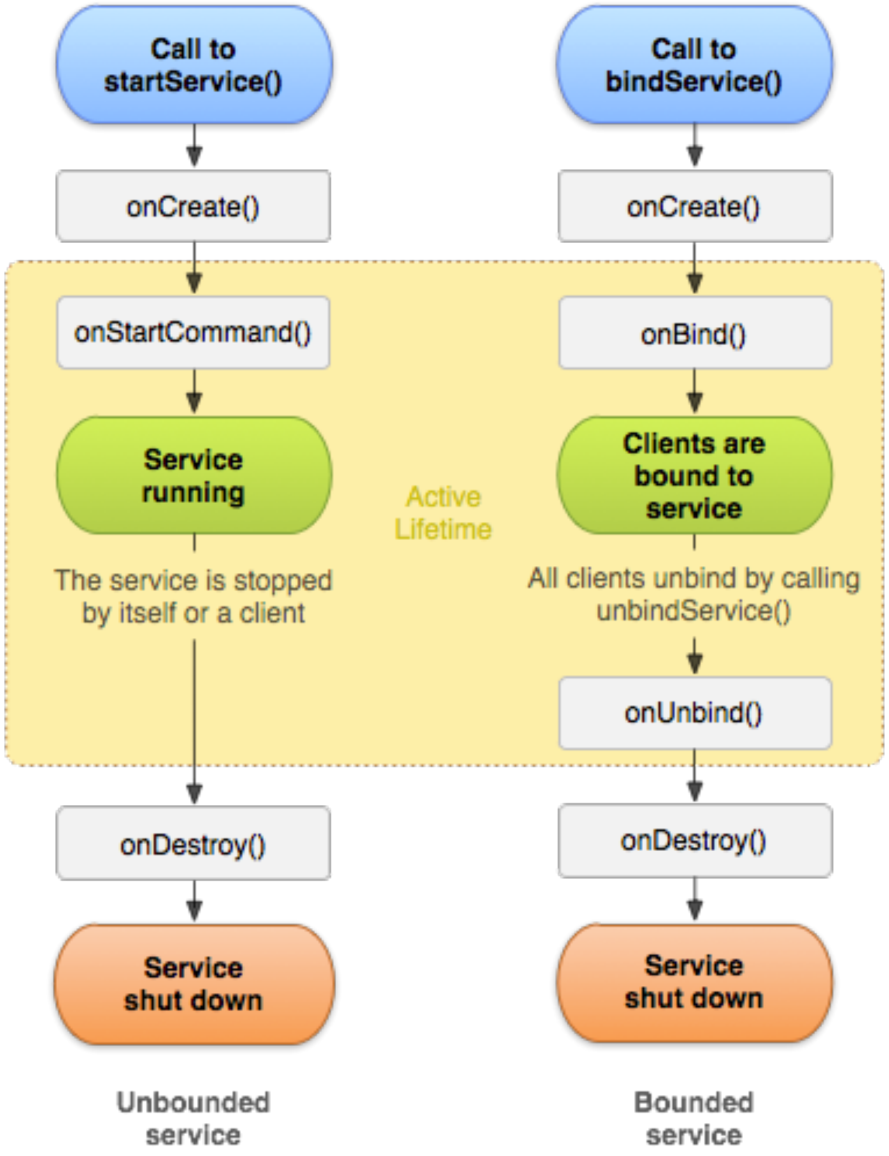
1. Put a **textView**, **editText** and **button** in an activity UI.

2. Input text in editText and press button, the textView shows the text. (Search Google for examples & Help)

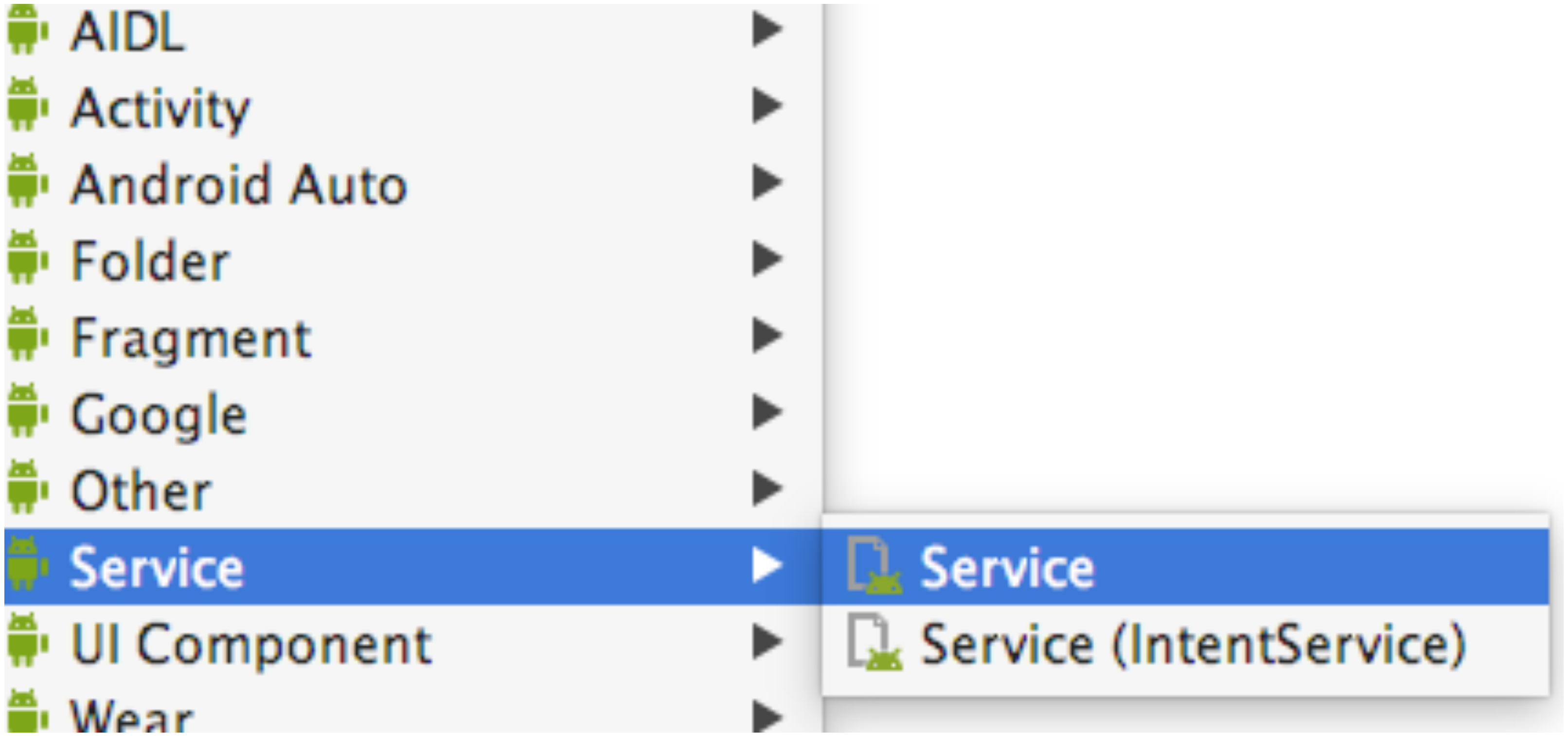
Background Program: Services

Intro:

https://developer.android.com/guide/components/services.html


Call to startService()
Call to bindService()
onCreate()
onCreate()
onStartCommand()
onBind()
Service running
Active Lifetime
Clients are bound to service
The service is stopped by itself or a client
All clients unbind by calling unbindService()
onUnbind()
onDestroy()
onDestroy()
Service shut down
Service shut down
Unbounded service
Bounded service

Create Service Class



Make it run using your Code

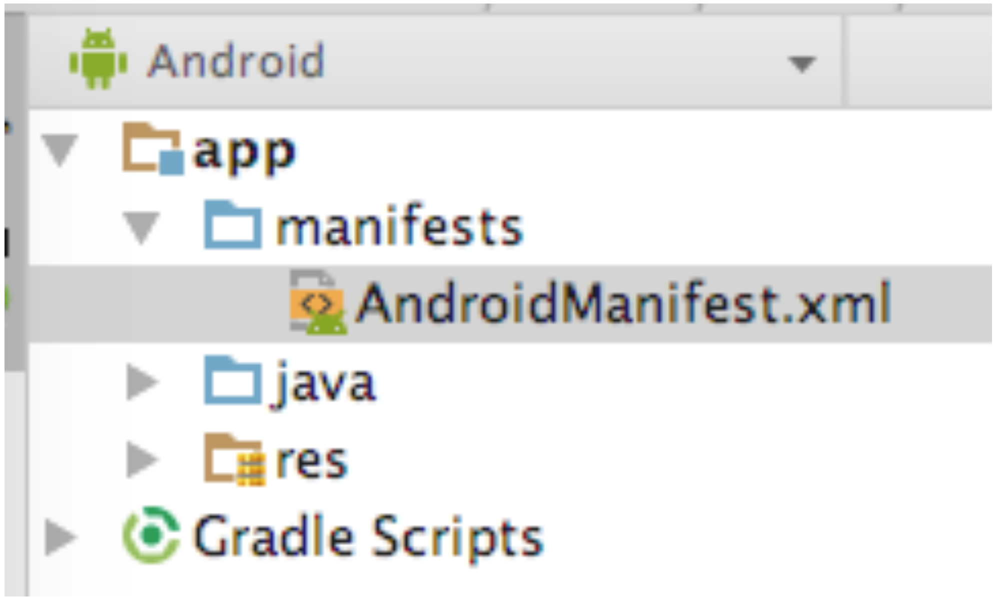
```
startService(new
Intent(MainActivity.this,
MyService.class));
//To stop
//stopService(MyService);
```

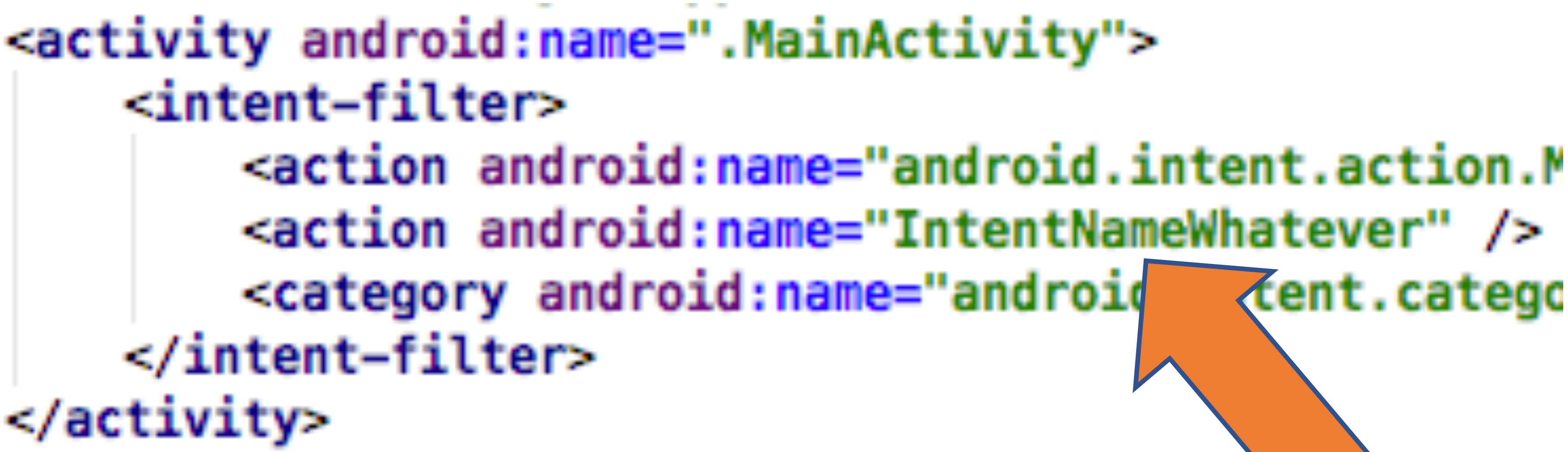
Service Activity Communication

Many ways. Intents are easy. Also, Intents can work as one-to-many broadcasts.

Send Data via Intents

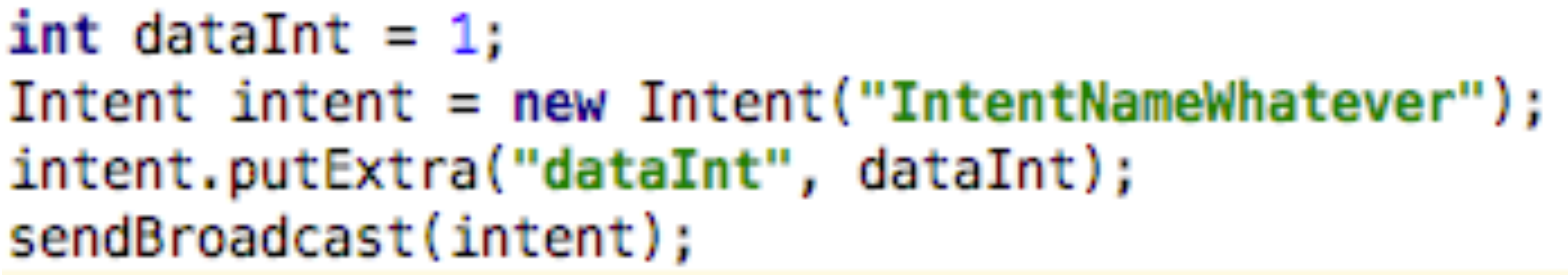
1. Name your intent
2. Declare a filter within the receiving program (e.g., Activity) in Manifests file





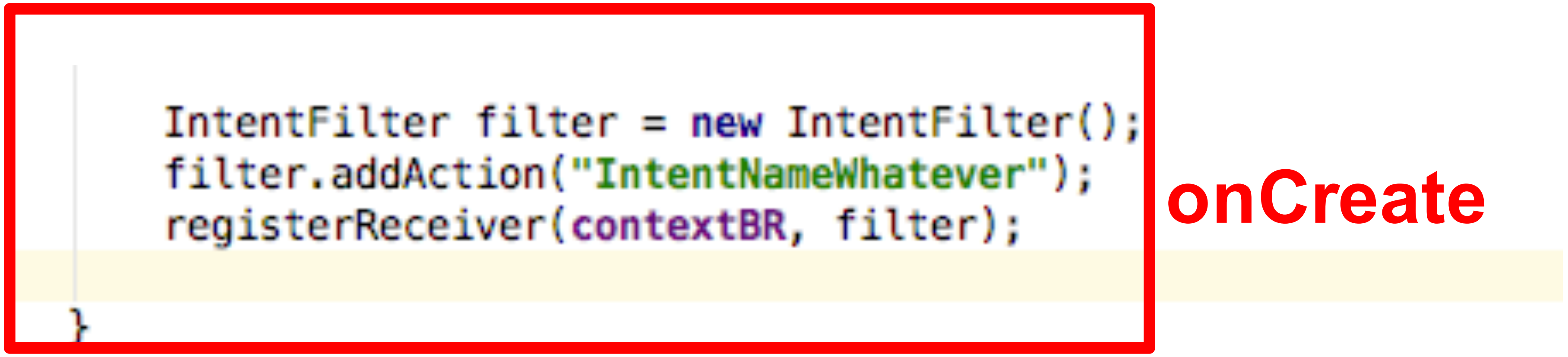
```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.M
        <action android:name="IntentNameWhatever" />
        <category android:name="androi        tent.catego
    </intent-filter>
</activity>
```

Send Data via Intents



```java
int dataInt = 1;
Intent intent = new Intent("IntentNameWhatever");
intent.putExtra("dataInt", dataInt);
sendBroadcast(intent);
```

Receive Data from Intents

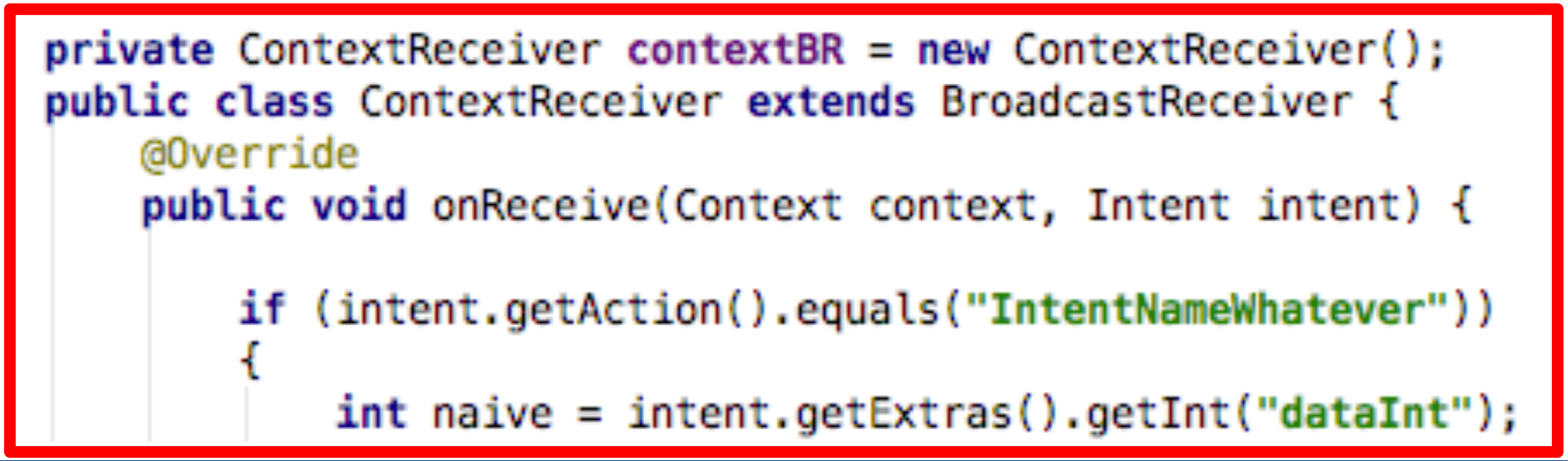


```
    IntentFilter filter = new IntentFilter();
    filter.addAction("IntentNameWhatever");
    registerReceiver(contextBR, filter);

}
```

onCreate

Activity class



```
private ContextReceiver contextBR = new ContextReceiver();
public class ContextReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {

        if (intent.getAction().equals("IntentNameWhatever"))
        {
            int naive = intent.getExtras().getInt("dataInt");
```

Exercise:
1. Create a Service B in an Activity A.

2. Send some data from B to A (verify received data using Log.d).

3. After receiving data, A sends some data to B (also Log.d).

More learning directions:

1. Learn to collect sensor data.

2. To store and read data from ContentProvider.

3. To use Azure (e.g., SQL database).

See you next week

COMP 90018

Tutorial on Android Development

Chu Luo, Eman Bin Khunayn
{chu.luo, eman.bin}@unimelb.edu.au