PV239 #android2

cv1

Marek Sedlak
@msed__

STRV

ANDROID IS THE BEST OPERATING SYSTEM IN THE WORLD

STRV

FAKE NEWS

STRV

@me

- I like android, but I like iOS too (psssh)
- I am multiplatform because I my first real job was this:
  - Implement iOS + Android + Win + Linux + OSX + Product delivery or gtfo
  - About the product and people, not platforms => meet the demand
- Currently mostly STRV
  - Android dev, Java/Node.js Backend dev
  - Multiple product lead
- Quantity vs quality? Long term => good balance
- Quality first

Organisation of PV239 android2

- Thursdays at 8:00am @A218
- 5 weeks in a row, then once per 2 weeks
- We will tykať si
- Need something? Slack me
- Slack rules
  - Full name on profile
  - Be in #android2
  - Make use of it, it’s a great tool

Organisation of PV239 android2

- Projects
  - Groups of 2-3 ppl
    - Already have a group? OK
    - Nope? Find someone :) you can use slack.
  - Anyone already has a project? -> Tell me
- By the end of 3rd exercise lesson -> **groups decided**
- Once you have **a group** && **a project**, you can actually start
- No need to wait If you like
- Use git (**get a repo asap**, e.g. github/bitbucket/gitlab ..)
  - https://github.com/github/gitignore/blob/master/Android.gitignore

LET'S DO SOME ANDROID

ANDROID STUDIO

What do you need?

STRV

ANDROID STUDIO

It is really simple to get started.

- Android Studio 2.2.3
  - Includes Android SDK
- 8GB RAM (my recomm.)
- Testing device (can be an emulator)

New project:

File > New Project… Next, Next, Next, Finish

https://developer.android.com/

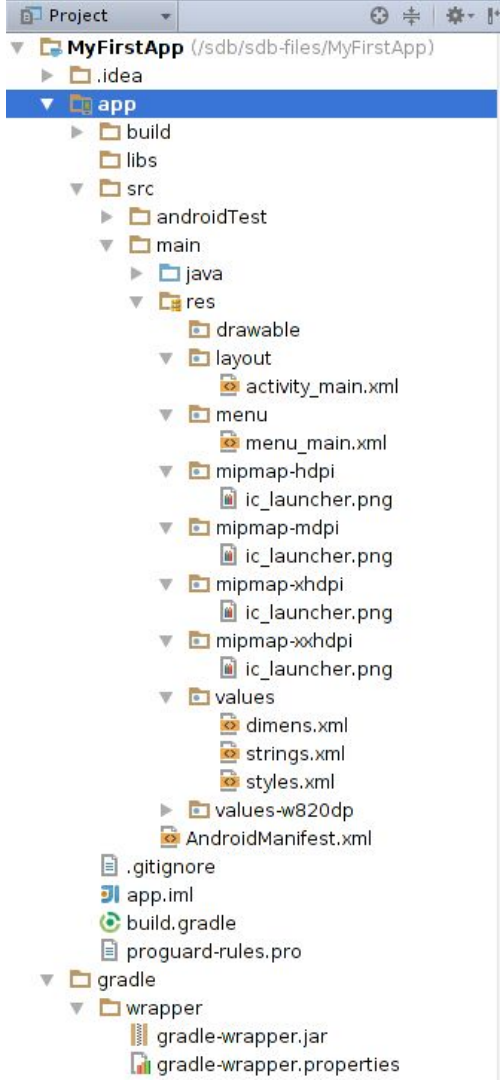
PROJECT STRUCTURE

What’s inside? Slides with many lines - sorry.

STRV

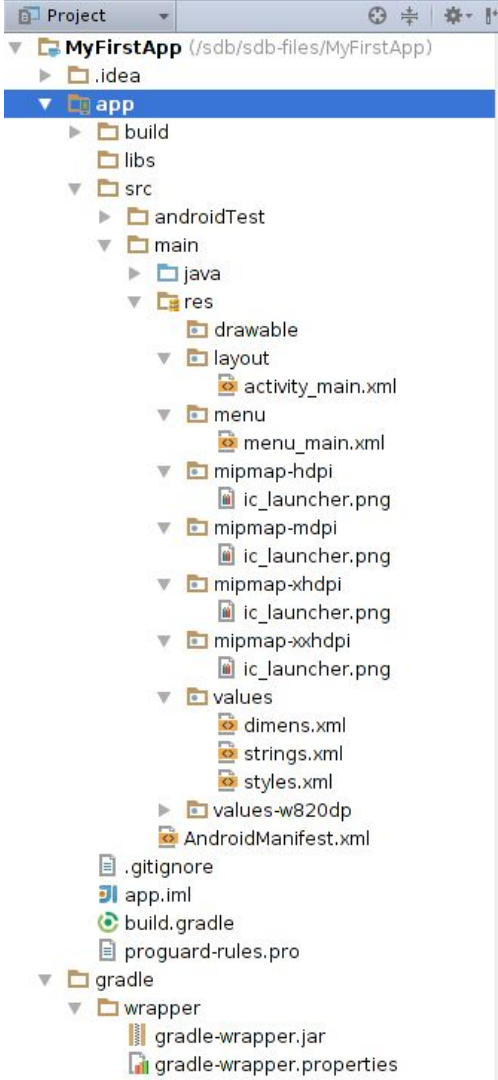
ANDROID STUDIO PROJECT STRUCTURE

- `.idea` - IDE specific (-> .gitignore)
  - This is the folder of IntelliJ IDEA IDE
- `.gradle` (-> gitignore)
  - Gradle data, state, cache (internal stuff)
  - Gradle - build tool, like Maven, ANT
- **`app` - no .gitignore this time**
  - **The Actual App**
- `gradle` - gradle wrapper
  - Java lib of a specific version of gradle



THE APP FOLDER - FINALLY THE ANDROID PROJECT

- It does not have to be named `app`
  - actually, it is a module within the android **studio** project
- `build.gradle` - important file, gradle settings
- `build` - output, once you build
- `libs` - place for libraries used within the project
- `src`
  - `main`
    - `java` - java classes
    - `res` - assets, images, strings, layouts,...
      - `drawable, mipmap - graphics`
      - `layout - xml layouts`
      - `values - place for strings, colors, styles`
    - `AndroidManifest.xml` - important, manifest file



UI COMPONENTS

Remember. UI Sells.

STRV

EACH JAVA OBJECT EXTENDS OBJECT CLASS

Typical Views:

- TextView
- Button
- EditText
- etc..

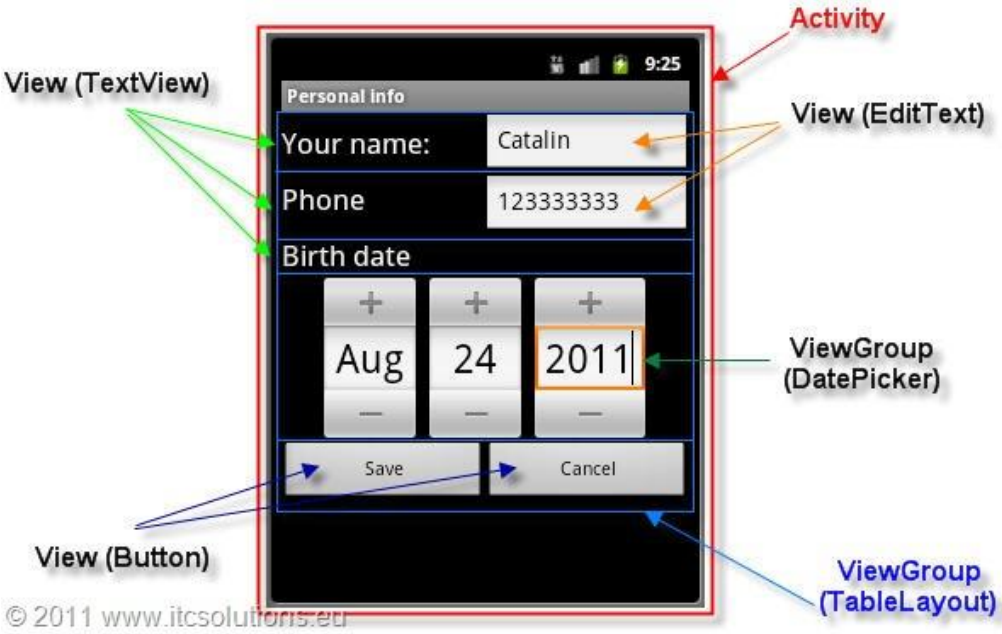
Typical ViewGroups:

- LinearLayout
- RelativeLayout
- FrameLayout
- TableLayout
- etc..

You typically use these elements in the:

- **Activity**
- Fragment
- etc..

LITTLE LESS ABSTRACT :)



http://www.itcsolutions.eu/2011/08/27/android-tutorial-4-procedural-vs-declarative-design-of-user-interfaces/

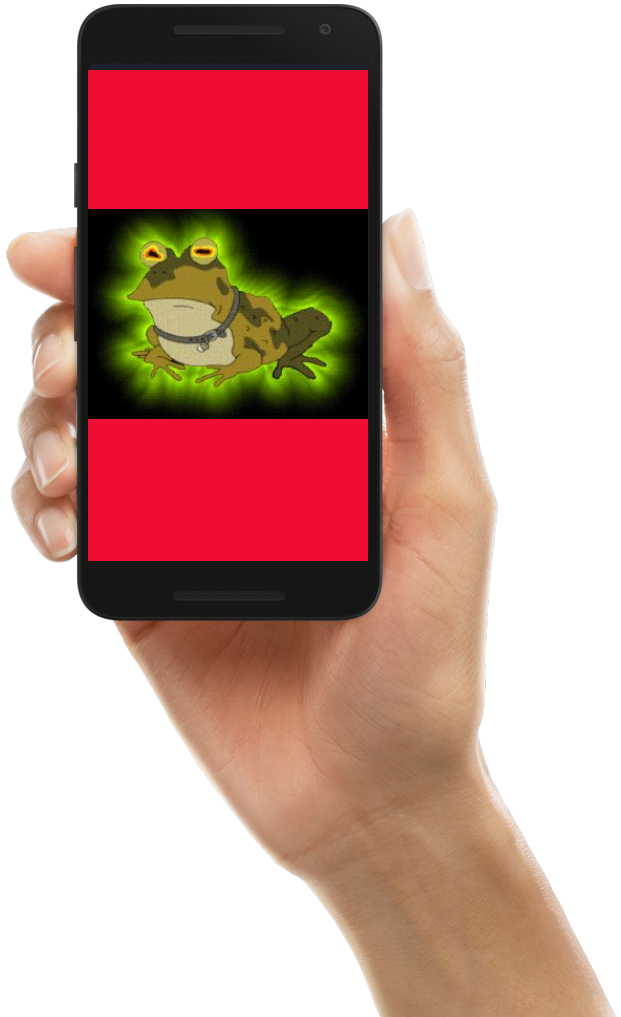
ACTIVITY

Activity has a lifecycle

- Activity Created
  - **onCreate()**
  - onStart()
  - **onResume()**
- Activity Running
  - **onPause()**
  - onStop()
  - **onDestroy()**
- Activity shut down

Navigation among activities

- android.content.Intent



INTERACTION WITH THE LAYOUT IN JAVA CODE

- XML

  ```
  android:id="@+id/loginButton"
  ```

- JAVA

  ```
  Button loginButton = (Button) findViewById( R.id.loginButton)
  loginButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
          // process with login (API call, etc..)
      }
  });
  ```

  - much better: `ButterKnife (try at home)`

EXERCISES

Let’s give it a try.
(The same as Android#1)

STRV

EXERCISE1

Use the activity to capture the input from the user

- Use vertical `LinearLayout`
- Put a text on the screen saying “Message”
  - Use `TextView`
- Place an input UI element with the button on the screen
  - They should be next to each other - horizontally (horizontal `LinearLayout`)
  - It’s `EditText` and `Button`
  - Name the button “Send”

EXERCISE2

a) **Use the “Send” button to show `Toast`**
b) **Use the “Send” button to pass the input to another activity and show it on the screen there.**
   - Use `Intent`

```
// 1st activity
Intent intent = new Intent(getBaseContext(), MessageActivity.class);
intent.putExtra("EXTRA_MESSAGE", message);
startActivity(intent);
// 2nd activity
String message = getIntent().getStringExtra("EXTRA_MESSAGE");
```

THANK YOU

See you on Slack.

STRV