

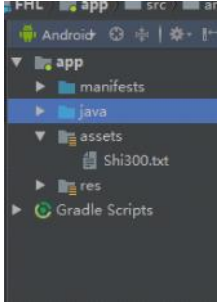
创建第一个安卓项目

安卓四大组件：

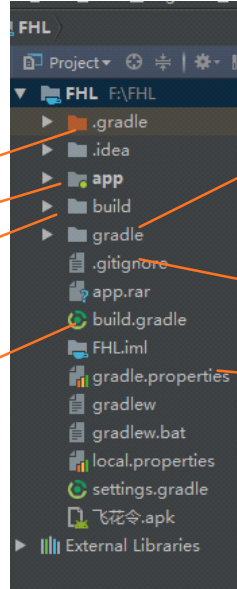
活动（Activity）、服务（Service）、广播接收器（BroadCast）和内容提供者（Content Provider）

2018年3月25日 13:42

Android模式的项目结构



Project模式的项目结构



.gradle和.idea

放置as自动生成的文件，不用去管也不要修改

项目的代码、资源等内容放置在里面

不用过多关心，放置编译时自动生成的文件

项目全局的gradle构建脚本，一般不需要修改

放的是gradle warpper的配置文件，使用gradle warpper的方式不需要提前把gradle下载好，而是根据本地缓存情况决定是否要联网下载gradle，as默认没有启动gradle warpper的方式，需要打开时可以点File - Setting - Build,Execution - Gradle进行配置更改

用来将指定的目录或文件排除在版本控制之外（char5 深入学习）

全局的gradle配置文件，这里配置的属性将会影响项目中所有的gradle编译脚本

其他详见《第一行代码》P20、19（pdf P28 30）

重点App目录

- 1.build（编译相关）
- 2.libs(第三方jar包)
- 3.adnroidTest:用来编写andr Test用的,可以对项目进行一些自动化测试
- 4.java：我们的代码
- 5.res：这个目录的内容很多，简单来说，项目使用到的所有图片、布局、字符串等资源都放在之歌文件夹下面。有很多的子目录，图片放在drawable下，布局放在layout下，字符串放在values下
- 6.AndroidManifest.xml 这是整个Android项目的配置文件，在程序中定义的四大组件都需要在这个文件里注册、另外在这个文件里添加权限声明（经常使用）
- 7.test
用来编写Unit Text（自动化测试的另一种方式）
其他参考pdf P30（原书P20）

注意：HelloWorld很多教程都有就不再说了

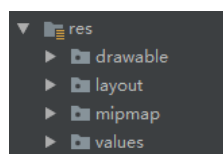
详解项目的资源

主要涉及res目录

所有drawavle开头的文件夹都是放图片的

所有mipmap开头的文件夹都是放图标的

后面的hdpi、xhdpi、xxhbp、xxxhbp表示分辨率的不同，当程序运行的时候，系统自动根据当前运行设备分辨率选择哪一分辨率的版本。只有一张则放在xxhdpi下面。通常系统不会自动生成带后缀的，建议自己建立这些文件夹



android studio drawable目录的分辨率

原创

2014年05月26日 10:12:04

1956

hdpi里面主要放高分辨率的图片,如WVGA (480x800),FWVGA (480x854)

mdpi里面主要放中等分辨率的图片,如HVGA (320x480)

xhdpi里面主要放低分辨率的图片,如QVGA (720X1280)

xxhdpi里面主要放低分辨率的图片,如QVGA (1080x1920)

dpi是"dot per inch"的缩写, 每英寸像素数。

四种密度分类: mdpi (medium), hdpi (high), xhdpi (extra high) and xxhdpi (Xextra high)

drawable-hdpi 对应DPI为240

drawable-mdpi 对应DPI为160

drawable-xhdpi 对应DPI为320

drawable-xxhdpi 对应DPI为480

获取屏幕DPI (屏幕密度) 方法:

```
[java]
• DisplayMetrics dm = new DisplayMetrics();
• dm = getResources().getDisplayMetrics();
• float density = dm.density;//屏幕密度 (像素比例: 1.0, 1.5, 2.0,3.0)
• int densityDpi = dm.densityDpi;//屏幕密度 (每寸像素: 160, 240, 320, 480)
```

放心, 这个滚动条下面什么也没有

使用字符串资源

下面是hello world里的字符串string.xml

```
<resources>
  <string name="app_name">HelloWorld</string>
</resources>
```

这两行代码定义了一个应用程序的字符串, 注意: "" 双引号里面的是字符的名称, 后面的HelloWorld才是你需要的字符串。

引用的方法:

1在代码中引用 R.string.app_name可以获得该字符串的引用

2在xml中通过@string/hello_worldapp_name式, 其中string是可以修改的如果引用图片就换成drawable, 引用的是应用图标就是mipmap, 引用的是布局文件就是layout以此类推。

示例:

下面举一个简单的例子来帮助你理解, 打开 AndroidManifest.xml 文件, 找到如下代码:

```
<application
  android:allowBackup="true"
  android:icon="@mipmap/ic_launcher"
  android:label="@string/app_name"
  android:supportsRtl="true"
  android:theme="@style/AppTheme"
  ...
</application>
```

其中, HelloWorld 项目的应用图标就是通过 android:icon 属性来指定的, 应用的名称则是通过 android:label 属性指定的。可以看到, 这里对资源引用的方式正是我们刚刚学过的在 XML 中引用资源的语法。

经过本小节的学习, 如果你想修改应用的图标或者名称, 相信已经知道该怎么办了吧。

详解build.gradle文件

Gradle是一个非常先进的项目构建工具。

这是最外层的build.gradle

```

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.2.0'
    }
}

allprojects {
    repositories {
        jcenter()
    }
}

```

jcenter():jcenter是一个代码托管仓库，Android的很多开源项目会放到它上面，声明这项配置之后我们就可以在项目中引用任何jcenter上的开源项目了。

dependencies闭包里声明了一个Gradle插件。因为Gradle不是专门为Android开发的，java，c++都可以使用Gradle来构建。

通常情况这个文件不需要修改

app目录下的build.gradle

```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 24
    buildToolsVersion "24.0.2"
    defaultConfig {
        applicationId "com.example.helloworld"
        minSdkVersion 15
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
                'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:24.2.1'
    testCompile 'junit:junit:4.12'
}

```

详细分析见pdf P34（原书P24）

使用Android日志工具Log

Log指的是android.util.Log

这个类里有五个方法

- Log.v() 用于打印那些琐碎的，意义的最小的日志信息。对应级别verbose，是Android日志里最低级的一种。
- Log.d() 用于打印一些调试信息，这些信息对你调试程序和分析问题应该是有帮助的，对应级别debug，比verbose高一级
- Log.i() 用于打印一些比较重要的数据，如果很想看到这些数据则使用这个方法，级别info，比debug高一级
- Log.w() 用于打印一些警告信息，提示有潜在危险，最好修复，级别warn，比info高一级
- Log.e() 用于打印程序中出现的错误信息，常常在catch中使用，表示严重问题需要立刻修复，级别error，比warn高一级

示例：

打开 HelloWorldActivity，在 onCreate()方法中添加一行打印日志的语句，如下所示：

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.hello_world_layout);
    Log.d("HelloWorldActivity", "onCreate execute");
}

```

参数：第一个为tag，一般是当前的类名，用于对打印信息进行过滤，第二个是msg，具体要打印什么。
运行后在logcat中看到打印信息。Logcat是Android Studio的一个小窗口，而不是app里的东西。