

Android Studio 3.0 使用 Annotation Processor 自动生成代码配置

Android

[Android Studio 3.0 使用Annotation Processor 自动生成代码配置](#)

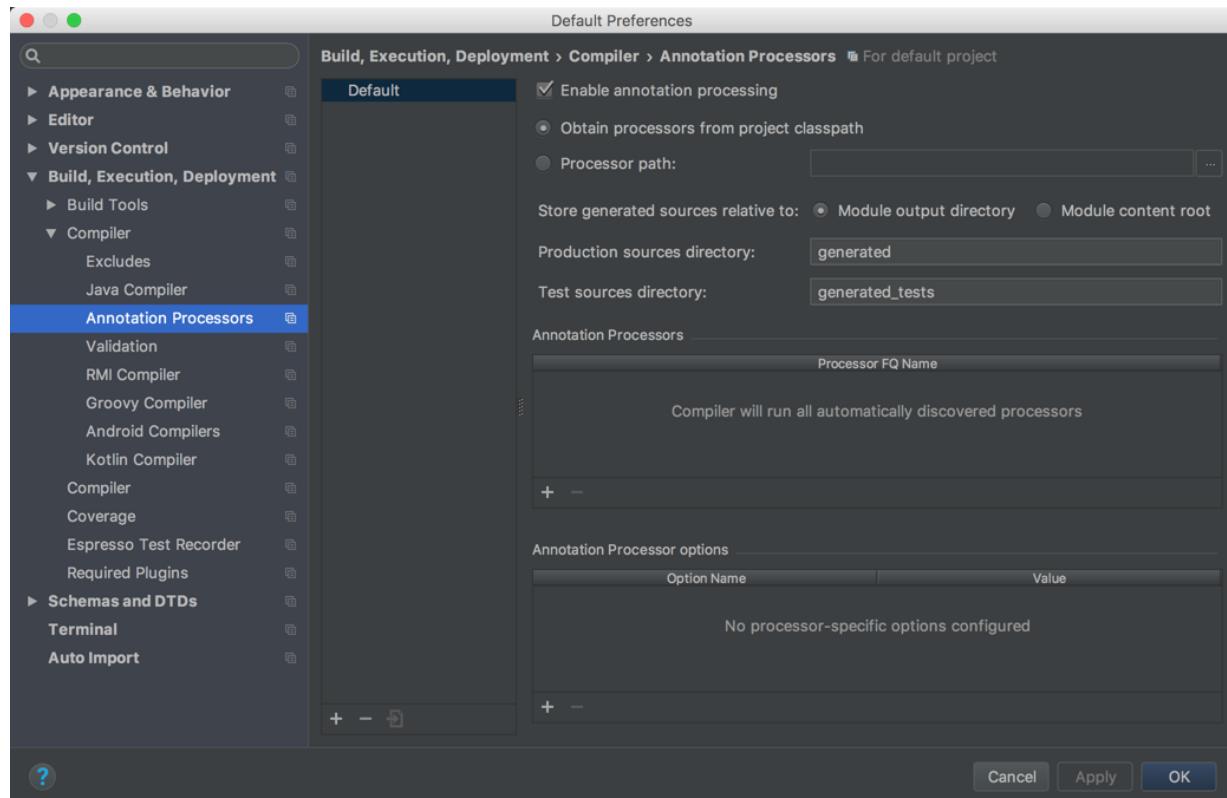
[Android Studio 配置](#)

[创建Processor处理类](#)

[配置依赖](#)

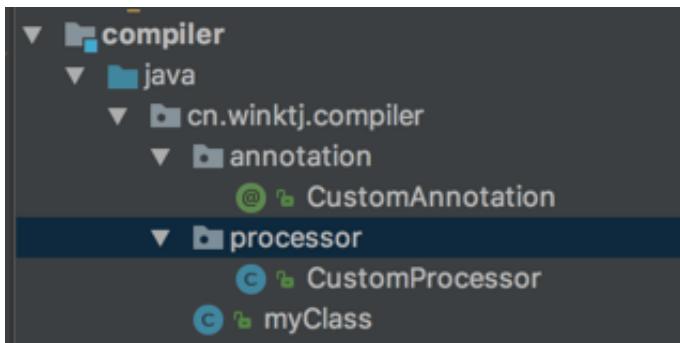
Android Studio 配置

需要打开IDE的Annotation Processes功能，以便在开发的时候可以使用自动出来的类。
打开”File”->”Other Settings”->”Default Settings”,打开如下标签，并使能Annotation Processor配置



创建Processor处理类

在AS中新建一个Java Module, 创建一个CustomAnnotation和CustomProcessor, 架构如图



CustomProcessor代码如下：

```
package cn.winktj.compiler.processor;

import com.google.auto.service.AutoService;

import java.io.IOException;
import java.io.Writer;
import java.util.Set;

import javax.annotation.processing.AbstractProcessor;
import javax.annotation.processing.Processor;
import javax.annotation.processing.RoundEnvironment;
import javax.annotation.processing.SupportedAnnotationTypes;
import javax.lang.model.element.TypeElement;
import javax.tools.JavaFileObject;

/**
 * Created by wink on 2017/11/9.
 */

@AutoService(Processor.class) //AutoService的注解，表示在编译阶段会执行该Processor
@SupportedAnnotationTypes("cn.winktj.compiler.annotation.CustomAnnotation") //支持注解的类型，可以填多个
public class CustomProcessor extends AbstractProcessor {
    @Override
    public boolean process(Set<? extends TypeElement> set, RoundEnvironment roundEnvironment) {
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append("package cn.winktj.compiler.processor.generated;\n");
        stringBuilder.append("public class GeneratedClass {\n");
        stringBuilder.append("}\n");
        try {
            JavaFileObject javaFileObject = processingEnv.getFile()
                .createSourceFile("cn.winktj.compiler.processor.generated.GeneratedClass");
            Writer writer = javaFileObject.openWriter();
            writer.write(stringBuilder.toString());
            writer.flush();
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return true;
    }
}
```

配置依赖

要使Processor在编译中被运行，需要引入google的autoservice包，打开compiler module的build.gradle文件，加入

```
apply plugin: 'java-library'

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.google.auto.service:auto-service:1.0-rc3' //引入auto-service包
}

sourceCompatibility = "1.7"
targetCompatibility = "1.7"
```

引用compiler模块的项目还需要加入annotationProcessor的引用。

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:23.4.0'
    testCompile 'junit:junit:4.12'
    annotationProcessor project(':compiler') #表示处理compiler下的Processor
    compile project(path: ':compiler')
}
```