**Plugin 插件开发教程**

## 一、配置SDK

打开IDE后，选择 **File | Project Structure**  
在配置窗口中点击 New... 新增SDK，选择 IntelliJ Platform Plugin SDK  
（如果第一次打开IDE，选择右下角 **Configure | Project Defaults | Project Structure** 打开配置窗口）

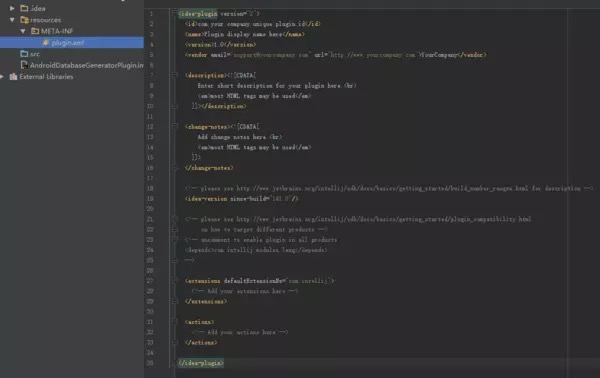
tes

## 二、新建Plugin Project

在打开的IntelliJ IDEA 中选择 **File | New | Project**，  
左侧菜单选择IntelliJ Platform Plugin项目，SDK选择刚才新配置的，下一步  
其中 Groovy、Python等啥其他附加的内容无需勾选  
最后输入项目名称确定，完成创建

## 三、配置插件项目

创建好的Plugin项目中，会默认生成一个 **plugin.xml** 的配置文件  
 其中可以修改该插件项目的相关配置信息，比如插件名称、插件版本号等，如下图



各标签意义具体为：

* **[ id ]** 插件id，类似于Android项目的包名，不能和其他插件项目重复，所以推荐使用com.xxx.xxx的格式
* **[ name ]** 插件名称，别人在官方插件库搜索你的插件时使用的名称
* **[ version ]** 插件版本号
* **[ vendor ]** 插件发布者信息，可以添加邮箱链接
* **[ description ]** 插件描述信息，在这里可以介绍你的插件内容，支持HTML标签
* **[ change-notes ]** 插件版本变更日志，支持HTML标签
* **[ idea-version ]** 对IntelliJ IDEA软件支持本插件版本号  
  分为since-build最低版本 和 until-build最高版本，两个属性可以任选一或者同时使用  
  官网有详细介绍 [www.jetbrains.org/intellij/sd…](https://link.juejin.im?target=http%3A%2F%2Fwww.jetbrains.org%2Fintellij%2Fsdk%2Fdocs%2Fbasics%2Fgetting_started%2Fbuild_number_ranges.html" \t "_blank)  
  大体规则为 **since-build <= 支持版本 < until-build**
* **[ extensions ]** 自定义扩展，暂时用不到
* **[ actions ]** 具体的插件动作，后面会介绍

配置中填写的信息，会在别人搜索你插件的时候展现介绍  
在 **File | Settings | Plugins** 中查看已有插件，或者再点击**Browse respositories**中浏览插件中心里其他人发布的插件  
当选中一个插件时，右侧就可以看到plugin.xml中配置的信息了  
同样，这里也可以进行插件的删除和添加安装操作，每次操作都需要重启软件后才能生效

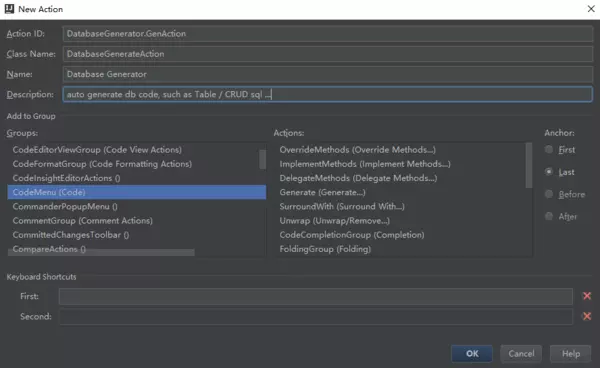
## 四、开始编写插件

插件是依附于IDE的，为其提供一些小功能，比如Android Studio中，  
**Code | Generate | Getter and Setter** 这个自动生成get和set方法的，其实就算是个插件

当然，这些都是Android Studio中默认自带的，我们可以根据需要开发新的插件项目创建好时有一个src文件夹，可以在其中**File | New ...**新建文件，  
主要有这么几种针对插件的特殊文件类型

* **GUI Form**：表单界面
* **Dialog**：对话框
* **XXXComponent**：作用域类，其中又分为**Application、Project、Module**分为作用于不同范围
* **Action**：处理插件交互行为的类

其中最主要的是Action类  
之前例子中提到了，生成getter setter方法的功能是在菜单栏中的Code | Generate | Getter and Setter选择的  
那么我们自己创建的插件选项在哪里呢？属于哪个菜单呢？这里在创建Action类的时候就可以指定操作入口，Action的创建的对话框如下  
http://www.jetbrains.org/intellij/sdk/docs/tutorials/action\_system/grouping\_action.html

Í

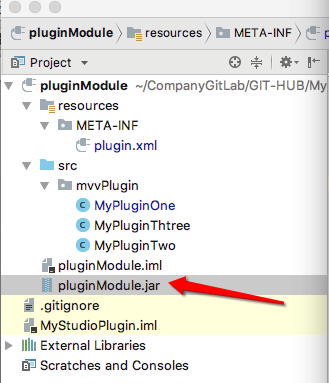
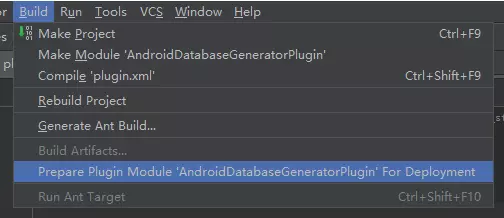
<http://www.jetbrains.org/intellij/sdk/docs/tutorials/action_system/grouping_action.html>

* 这里的信息我们都可以再次修改，但是class name修改时要注意和类文件名匹配
* 可以修改添加多个，即在不同菜单中都可以选择使用
* 同一个group下多个action的id不能重复

<https://translate.google.com/translate?hl=zh-CN&sl=en&u=https://www.jetbrains.org/intellij/sdk/docs/tutorials/action_system/grouping_action.html&prev=search>

## 五、打包

（这里我们先把整个流程介绍完，更复杂的插件开发会下一篇介绍）  
比较简单，直接在顶部主菜单栏中选择 **Build | Prepare Plugin Module XXX For Deployment**即可  
**打包前要注意把 plugin.xml 配置文件中的相关信息填写完整**



# 插件SDK中的常用对象介绍

## Virtual File

虚拟文件类。  
可以当做Java开发中的File对象理解，概念比较类似

#### 获取方法

* 通过Action获取: **event.getData(PlatformDataKeys.VIRTUAL\_FILE).**  
  这个也是之前教程（一）中的获取方法
* 通过本地文件路径获取: **LocalFileSystem.getInstance().findFileByIoFile()**
* 通过PSI file获取: **psiFile.getVirtualFile()**
* 通过document获取: **FileDocumentManager.getInstance().getFile()**

#### 用处

传统的文件操作方法这个对象都支持，比如获取文件内容，重命名，移动，删除等

## PSI File

PSI系统下的文件类。

#### 获取方法

* 通过Action获取: **e.getData(LangDataKeys.PSI\_FILE).**
* 通过VirtualFile获取: **PsiManager.getInstance(project).findFile()**
* 通过document获取: **PsiDocumentManager.getInstance(project).getPsiFile()**
* 通过文件中的Element元素获取: **psiElement.getContainingFile()**  
  如果要通过名字获取，请使用 **FilenameIndex.getFilesByName(project, name, scope)**

#### 用处

作为PSI系统中的一个元素，可以使用PSI Element的各种具体方法

项目实战

阐述项目组件开发思想