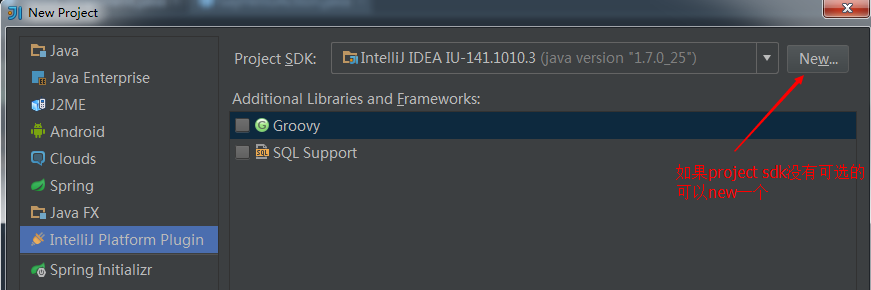
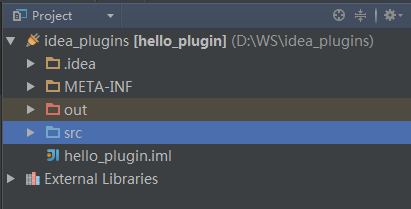
[初试IDEA插件开发](http://blog.csdn.net/mislead/article/details/46385653)

这里我们写一个自己的插件，仅仅是简单的弹出一个对话框。

#### 1、新建一个插件项目

* File->new->project->intellij platform plugin
* 选择好project sdk：如果project sdk为空，只需要new一个，选择你的idea安装目录即可。
* 点击下一步，选择好安装路径完成新建。   
  
* 建好的目录结构   
  

#### 2、实现弹出对话框的功能

* 新建一个Application Component：在src目录上右键->new->Application Component。
* 添加一个方法sayHello()。其他需要实现的接口方法暂且不管。具体代码（注意import的包名）：

import com.intellij.openapi.components.ApplicationComponent;

import com.intellij.openapi.ui.Messages;

import org.jetbrains.annotations.NotNull;

/\*\*

\* Created by mislead on 2015/6/4.

\*/public class HelloComponent implements ApplicationComponent {

public HelloComponent() {

}

public void initComponent() {

// TODO: insert component initialization logic here

}

public void disposeComponent() {

// TODO: insert component disposal logic here

}

@NotNull

public String getComponentName() {

return "HelloComponent";

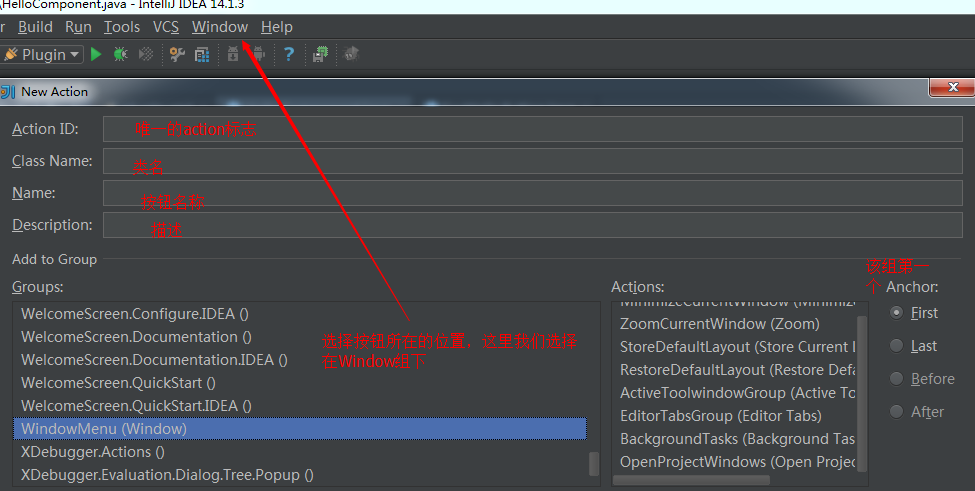
}

public void sayHello() {

Messages.showMessageDialog("hello, plugin", "hello", Messages.getInformationIcon());

}

}

* 新建一个Action：   
  
* Action具体代码：

import com.intellij.openapi.actionSystem.AnAction;

import com.intellij.openapi.actionSystem.AnActionEvent;

import com.intellij.openapi.application.Application;

import com.intellij.openapi.application.ApplicationManager;

/\*\*

\* Created by mislead on 2015/6/4.

\*/public class SayHelloAction extends AnAction {

public void actionPerformed(AnActionEvent e) {

Application application = ApplicationManager.getApplication();

HelloComponent component = application.getComponent(HelloComponent.class);

component.sayHello();

}

}

* 完成之后运行或者调试IMG_259：运行会重新运行一个IDEA 的实例程序，第一次可能会花点时间，会重新提示你进行设置和注册。
* 插件效果：   
  