action的实现方式

1、定义个pojo（简单的java对象）类

好处：自定义一个普通java即可，不具有侵入型

|  |
| --- |
| **package** com.qimh.action;  **public** **class** PojoAction {  **public** String execute(){  System.*out*.println("pojo action.....");  **return** "success";  }  } |

2、实现action接口

好处：使的我们编写的代码更加规范

|  |
| --- |
| **package** com.qimh.action;  **import** com.opensymphony.xwork2.Action;  **public** **class** InterfaceAction **implements** Action {  **public** String execute() **throws** Exception {  // **TODO** Auto-generated method stub  System.*out*.println("InterfaceAction ......");  **return** Action.*SUCCESS*;  }  } |

3、继承ActionSupper类

好处：可以继承一下ActionSupport的一些功能，如：验证

官方推荐使用

|  |
| --- |
| **package** com.qimh.action;  **import** com.opensymphony.xwork2.ActionSupport;  **public** **class** ExtendsAction **extends** ActionSupport {    @Override  **public** String execute() **throws** Exception {  // **TODO** Auto-generated method stub  System.*out*.println("extends action....");  **return** ActionSupport.*SUCCESS*;  }  } |