1.Struts2是在WebWork2基础上发展而来

2.struts2的优点

a.在设计上Struts2没有像Struts1那样与ServletAPI和struts API有着 紧密的耦合，Struts2的应用可以不依赖于ServletAPI和struts API. Struts2属于无侵入式设计

b.Struts2提供了拦截器，可以进行AOP编程，实现权限拦截等功能。

c.Struts2提供了类型转换器，可以把特殊的请求参数转换成需要类型。

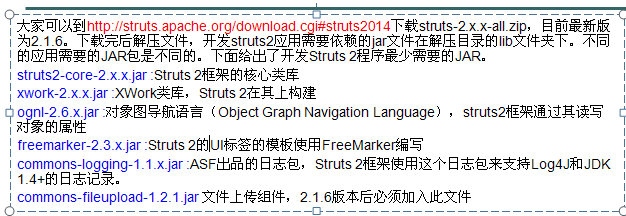
d.Struts2提供支持多种表现技术，如jsp，freemarker，velocity等。

e.Struts2的输入校验可以指定方法进行校验，解决了Struts2长久篇

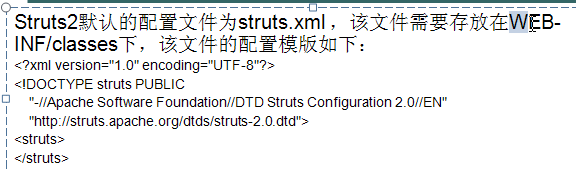
f.提供了全局范围，包范围和Action范围的国际化资源文件管理实现

3.搭建Struts2开发环境

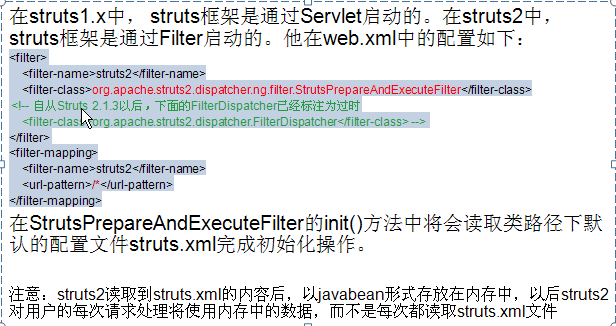
a.找到开发Struts2应用需要使用的jar包



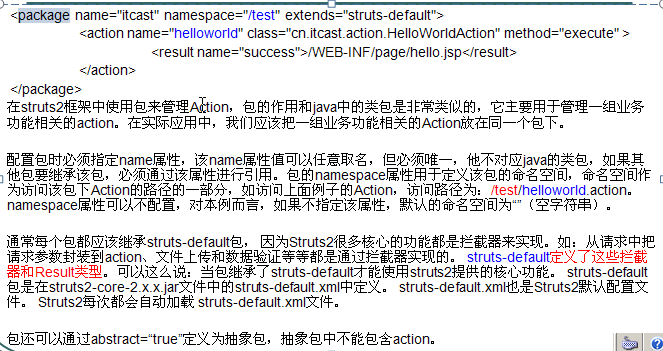
b.编写Struts2的配置文件



c.在web.xml中加入Struts2MVC框架启动配置



4.Struts2配置中包的配置



包中命名空间的优点，可以简化书写

访问：ip:端口/命名空间/actioin名

注：不要忘记继承struts-default

5.添加struts2提示

Exlipse--Windows—preference—xml—xml catalog—add

找到struts\_xxx.dtd文件添加进去，在struts的jar包中有，选择url，下面填写

<http://struts.apache.org/dtds/struts-2.1.7.dtd。在struts.xml>文件中有这句话

6.Action名称搜索顺序



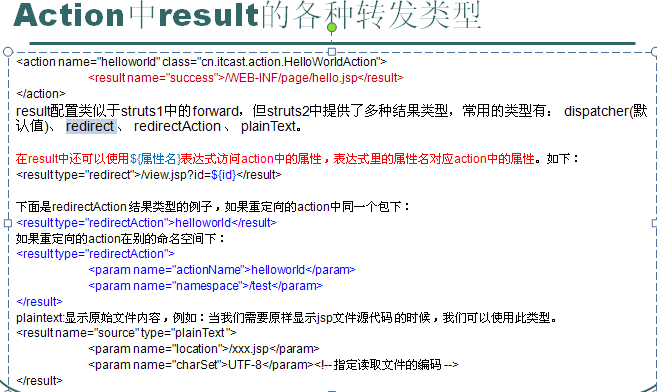
7.未指定内容的默认值

如果没有为action指定class 默认为ActionSupport

如果没有为action指定method 默认为action中的execute()方法

如果没有为action指定result的name 默认值为success

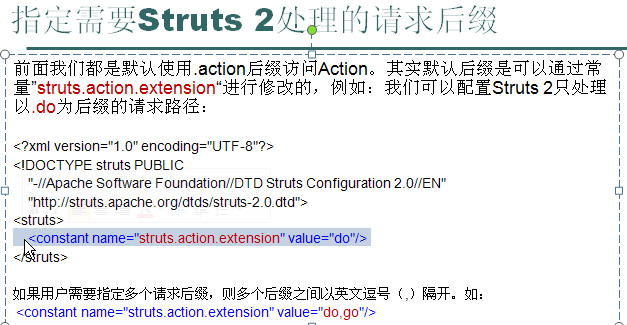
8.Action中result的各种转发类型



9.为action的属性注入值



10.指定Struts2的请求后缀



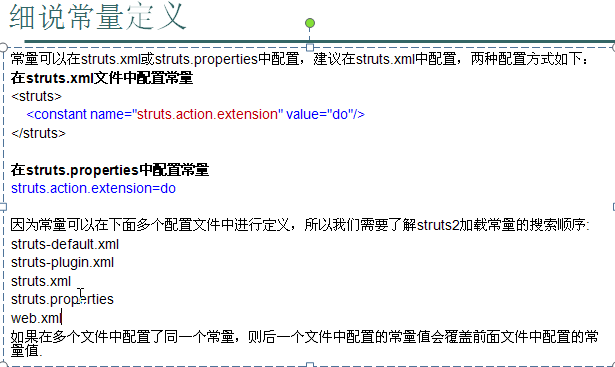
如：<struts>

<constant name=*"struts.action.extension"*

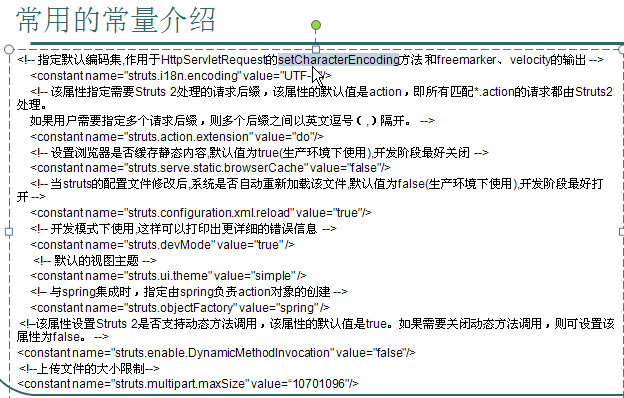
value=*"do,go"*></constant></struts>

注：修改后不加后缀也不能访问

11.常量定义

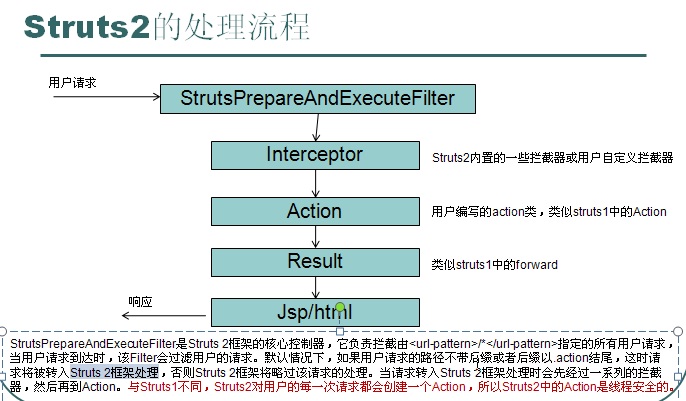


12.常用的常量



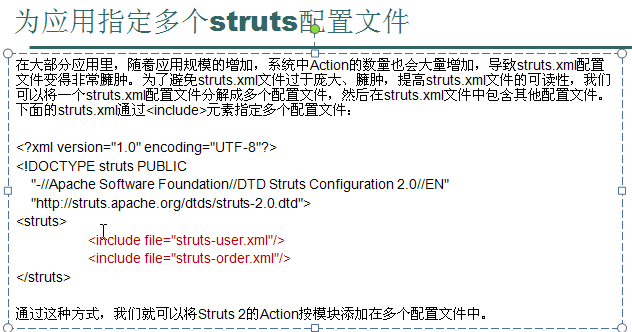
上传文件大小为总大小

13.struts2处理请求流程



Struts1中的action为单例的，Struts2中对用户的每一次请求都会创建一个action，线程安全的

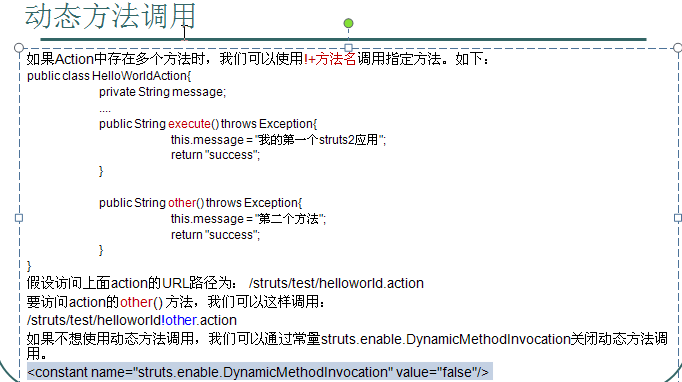
14.struts为应用配置多个配置文件



引入配置文件: <include file=”文件名”/>

15.动态方法调用

之一：不推荐，常禁用



之二：推荐，通配符



根据用户输入的值来调用方法

16.接收请求参数，必需定义属性提供set方法

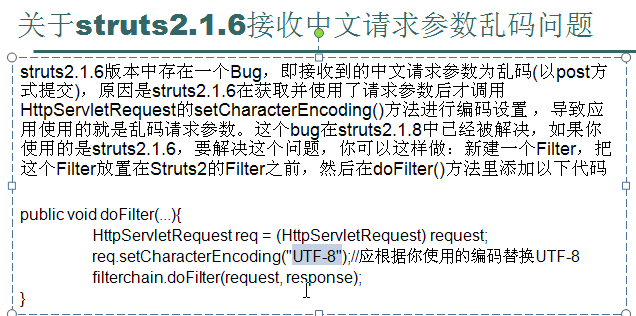


基本类型，Integer id , String name;

复合类型，Personal personal.id,personal.name,传递和获取都要加person

接收请求参数，必需定义set方法

17.struts2.1.6中文乱码解决



18.类型转换器

局部：只能对action转换，全局

默认接收的日期格式为yyyy-mm-dd 如果是yyyymmdd则会报错

局部：



自定义需要继承DefaultTypeConverter类，重写convertValue方法

注册，添加到xxx.properties文件中

全局：

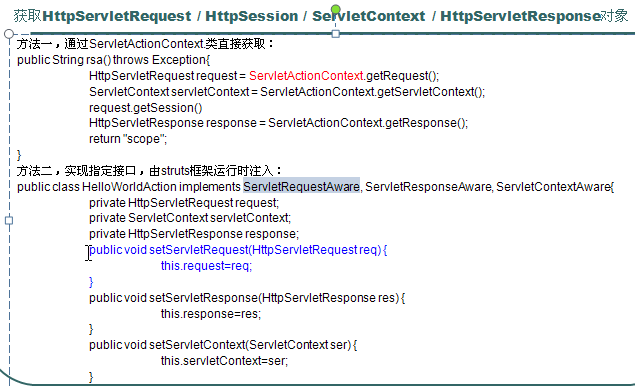


相对于局部这里文件名和key都要变

19. 访问或添加/request/session/application



20.得到HttpServletRequest/HttpSession/ServletContext/HttpServletResponse对象



方法一简单，建议请用。

21.文件上传



例：

public class HelloWorldAction {

private File image;//获取文件

private String imageFileName;//获取文件名，命名为文件上传名+FileName格式

public String getImageFileName() {

return imageFileName;

}

public void setImageFileName(String imageFileName) {

this.imageFileName = imageFileName;

}

public File getImage() {

return image;

}

public void setImage(File image) {

this.image = image;

}

public String execute() throws Exception{

String realpath = ServletActionContext.getServletContext().getRealPath("/images");//获取真实路径

if(image!=null){

File savefile = new File(new File(realpath), imageFileName);

if(!savefile.getParentFile().exists()) savefile.getParentFile().mkdirs();

FileUtils.copyFile(image, savefile);// FileUtils，在commens-io-xxx.jar包中，是一个工具类用于复制文件

ActionContext.getContext().put("message", "上传成功");

}

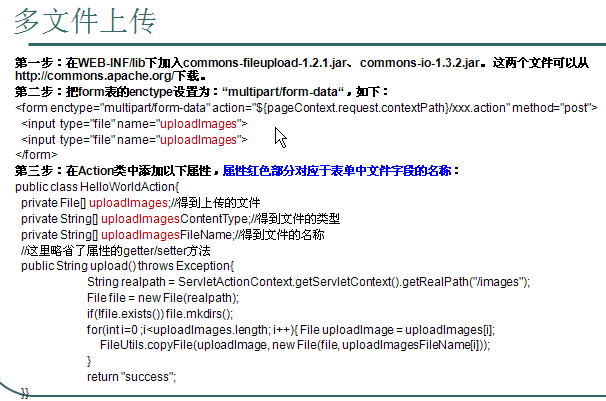
return "success";

}

}

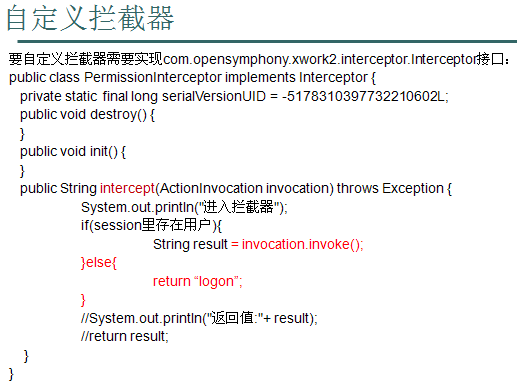
设置文件大小：在struts配置文件中<constant name="struts.multipart.maxSize" value="10701096"/>字节

22.多个文件上传

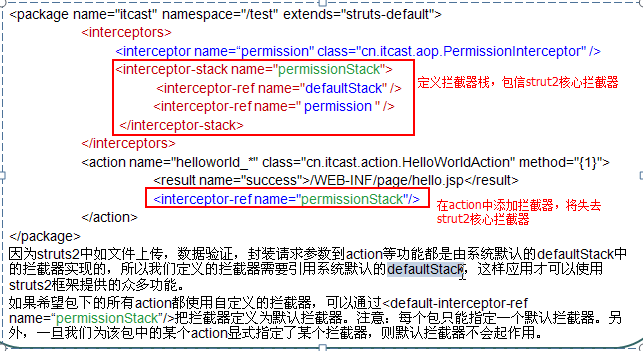


在处理的时候把属性的类型定义为数组或集合类型即可。

23.自定义拦截器

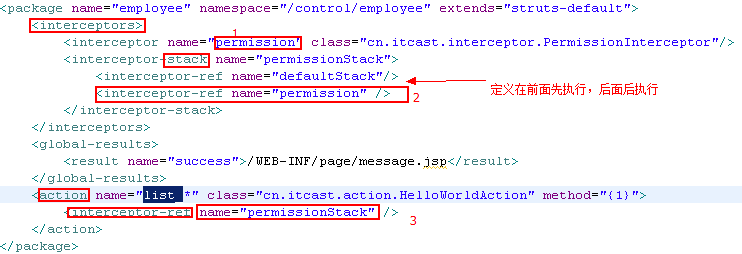


配置：



注：如果在action中添加拦截器，就会失去strut2的核心拦截器，定义拦截器栈解决

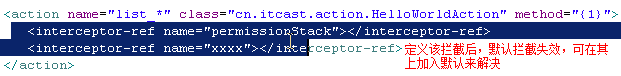
例：



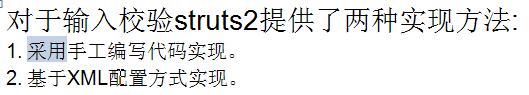
如果想让拦截器在整个包中都有效，在包中加入



再定义私有拦截：

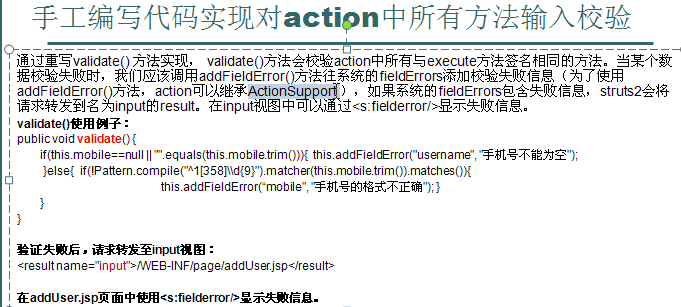


24.struts2输入校验，所有方法和指定方法，不管使用什么校验Action类都需要继承ActionSupport类



Validate在action类中输写

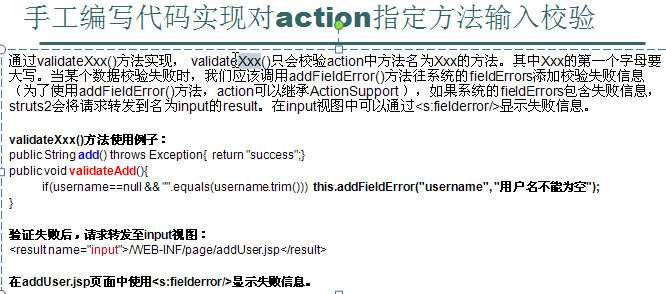
手工-所有：





校验错误会跳转到，结果为input的页面中

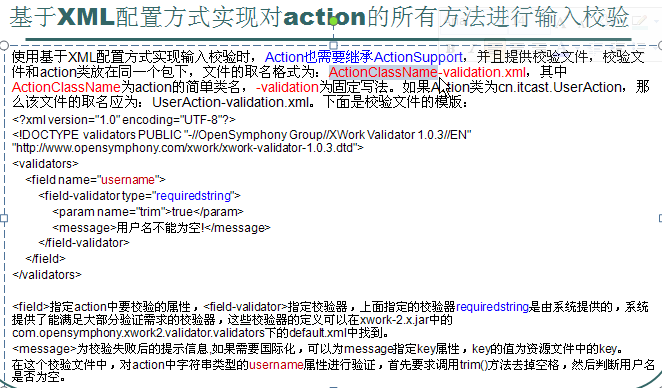
手工-指定：



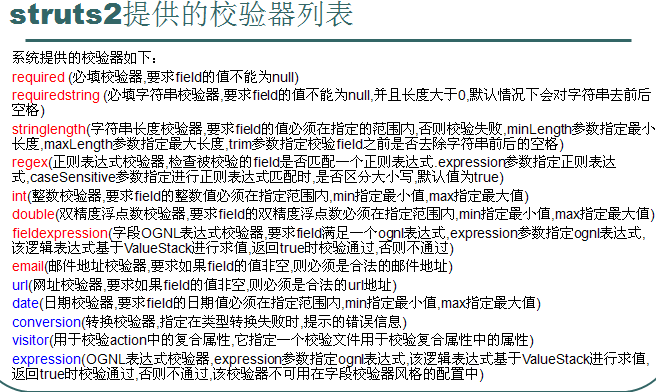
ValidateXxx(); Xxx第一个字母为大写



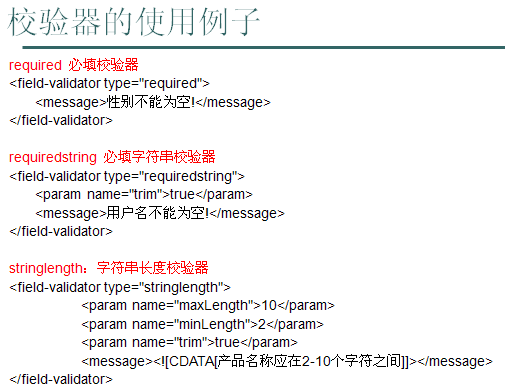
Xml配置—所有

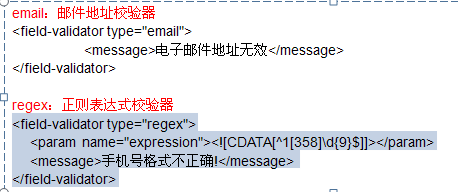


系统提供校验器：

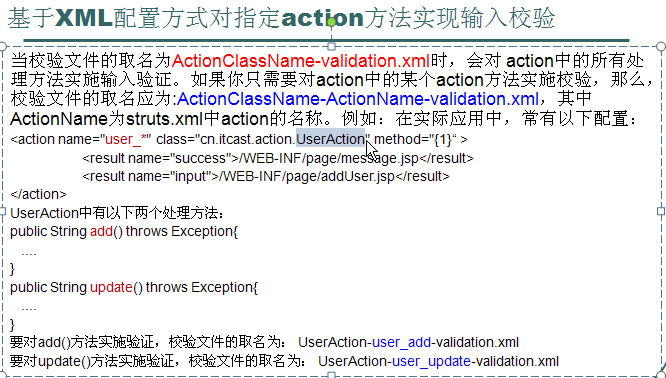


校验器使用例子：

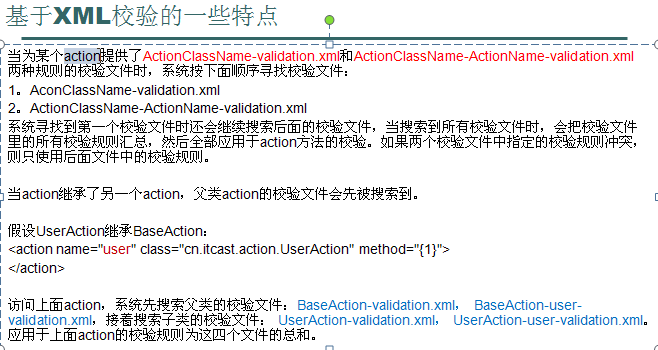




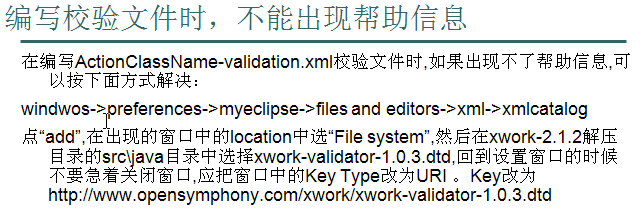
Xml配置—指定action



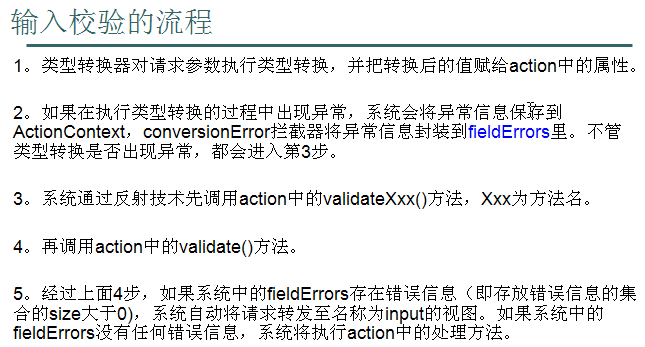
基于xml校验的特点：



校验文件无提示解决：



25.输入校验流程



进入到input视图有两种情况：1.类型转换失败，2.校验失败

26.国际化

资源文件

全局：

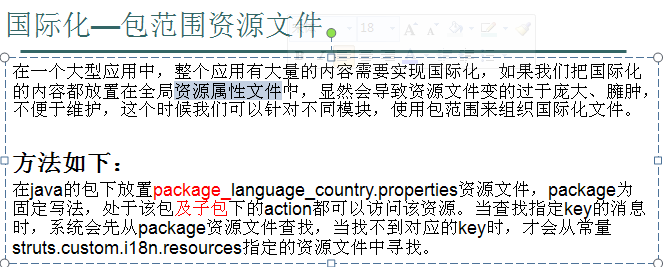




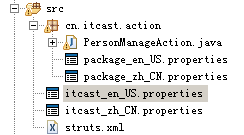
例：



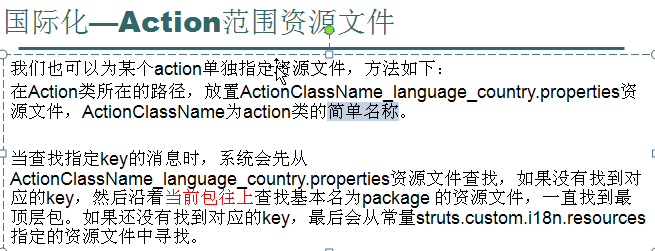
包范围：



全局，包范围目录结构：



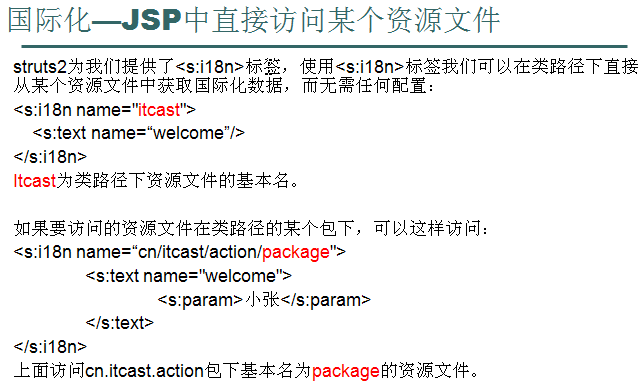
Action范围：



各种范围的国际化，只是文件位置和文件名不一样

如果都配置了，先寻找Action，再找包，再找全局

不用配置直接访问方式：不配置是指不全规范命名，也不用在struts.xml文件中加入国际化常量

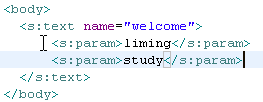


配置i18n 的name属性与文件对应即可

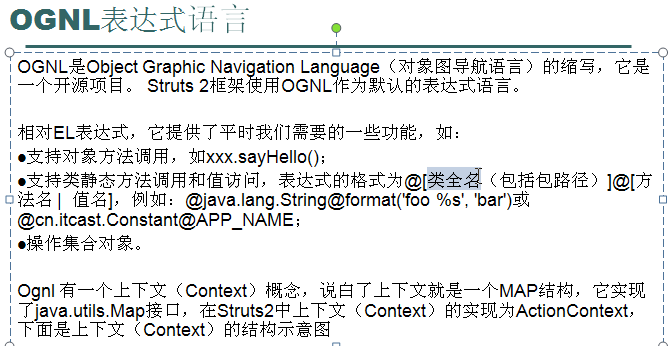
27.输出带点位符的信息，全局



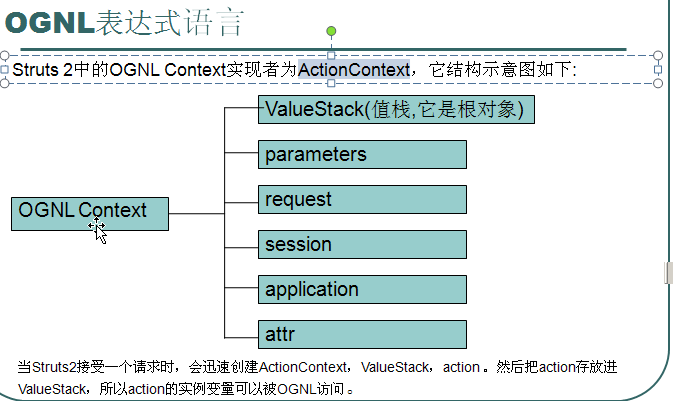
例：jsp



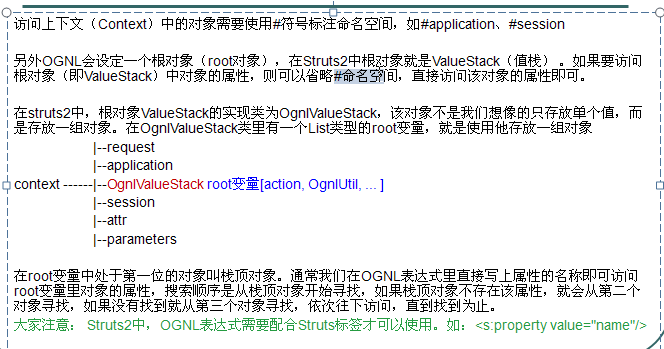
28.ognl表达式



Ognl结构图：



访问：有一个例外，根对象，ValueStack

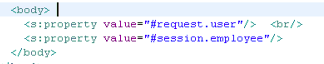


根对象，可以有两种访问方式：

1. ,根对象(ValueStack)中

2.el表达式：${message}

其它对象访问：



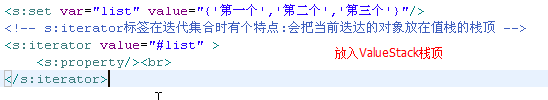
各对象意义：



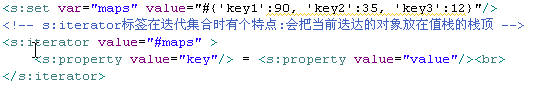
使用ognl创建list，map对象



访问list：

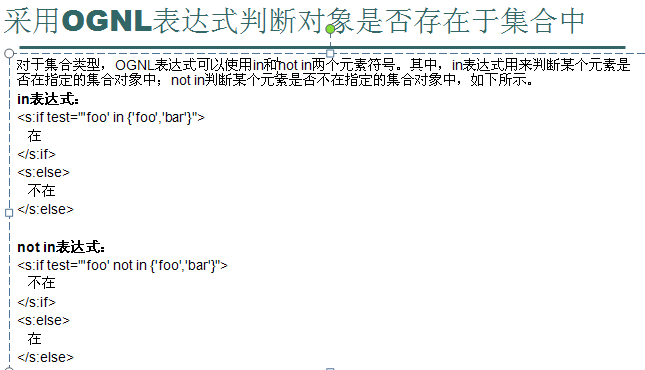


访问map:

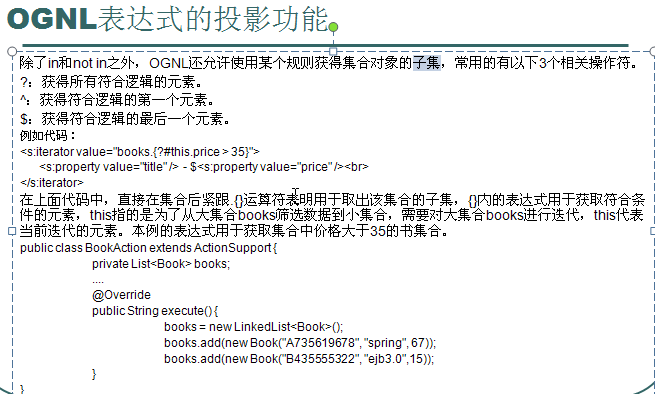


输出key和value

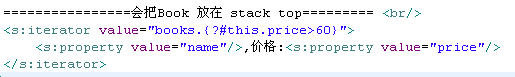
Ognl判断对象是否在集合中：



Ognl投影功能：可以理解为筛选，一般很少用，会在业务层就过滤掉

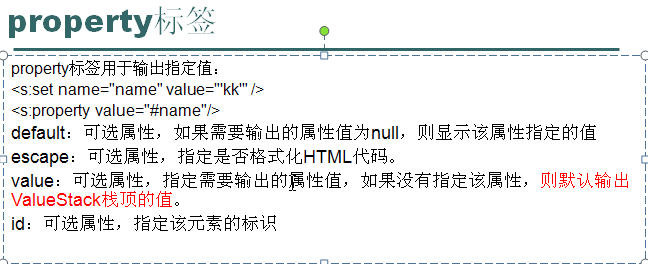


例：

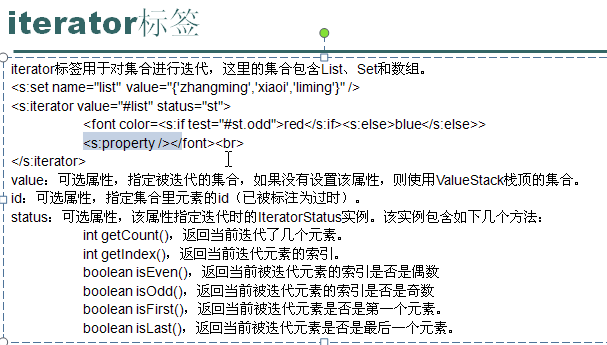


29.Struts2常用标签

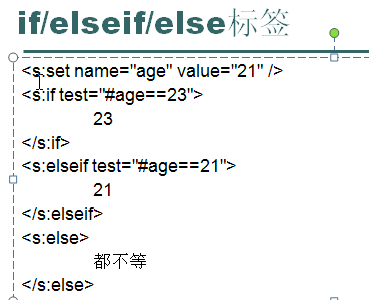
Property标签使用：



Iterator标签使用：



If/else标签使用



url标签



<s:Param中value默认会当ognl表达式处理

<s:url中默认会当字符串处理，加%{}后当ognl使用

表单标签：

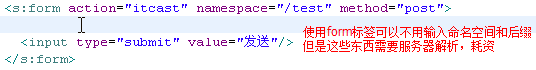






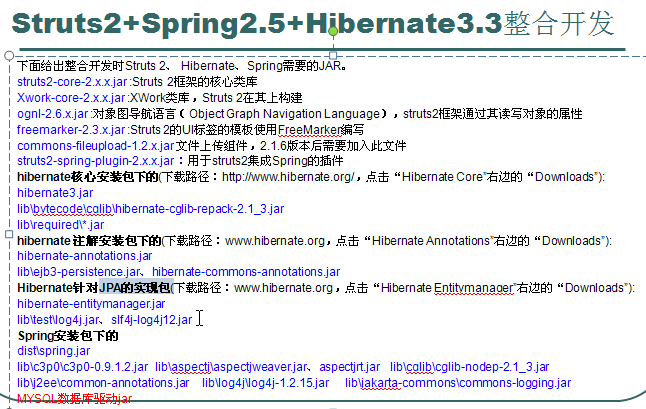
30.





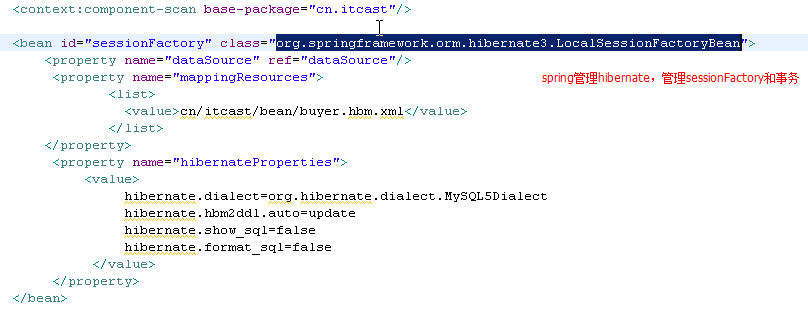
31.ssh整合

Spring整合hibernate



Spring管理hibernate，由spring提供

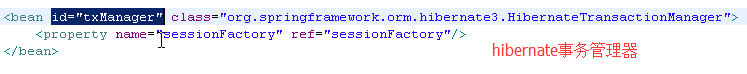
Spring配置文件配置



配置数据源，c3p0



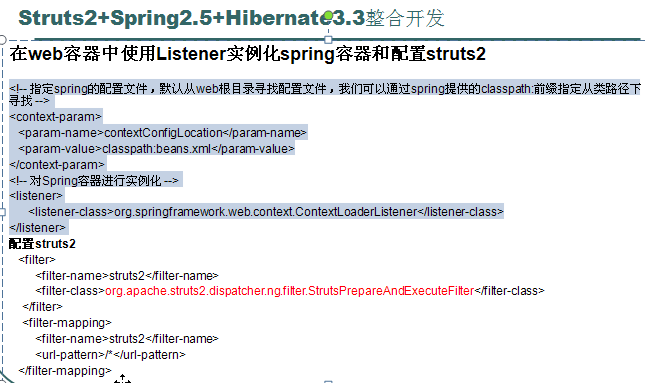
连接url中的&号要转换成xml认识的$amp;&为xml特殊字符



使用基于注解配置事务：



再整合struts2



在web应用中实例化spring容器通过listener

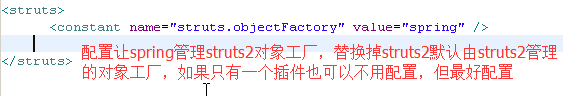
Web.xml配置



配置struts2，通过filter

Web.xml配置





Struts2配置



测试两种输出方式：

