# SSM开发框架编码规范

## 前言

### 介绍

本文档内容描述SSM(Struts+Spring+Mybatis)开发框架的基本结构和编码指导规范。在统一团队成员代码风格一致性上，编码规范起到了至关重要的作用。在开发过程中有很多地方受益于编码规范。遵循编码规范可使小组构建一致、高效的程序。许多实例证明，如果在项目初期简历一套完善的编码规范，在开发过程中对代码扩展，及重构都有好处。

### 目标

* 让开发人员快速理解SSM开发框架，以及相关技术
* 保持团队成员代码风格的一致性
* 提高代码质量与可读性，从而改善开发人员之间的沟通效果

### 适用范围

本文档规范适用于金鑫公司内部所有以SSM作为开发框架的项目。

## 软件技术环境

|  |  |  |
| --- | --- | --- |
| 软件技术名称 | 版本 | 备注 |
| JAVA | jdk1.6.0\_10 |  |
| Tomcat | apache-tomcat-6.0.35 |  |
| Struts | 2.0.11 |  |
| Spring | 3.0.5 |  |
| Mybatis | 3.0.5 |  |

## 框架总体结构

SSM框架总体结构如下图所示：



## 程序文件介绍

### 公共核心程序包

公共核心程序存放在core包下：com.jinxinol.core.\*。在核心包中提供了以下内容：常用工具类、WEB层通用支持类、Service层通用支持类、数据访问层通用支持类、WebService通用支持类。

#### 常用工具包

常用工具包(com.jinxinol.core.util.\*)提供了在编码过程中频繁使用的工具类：

* **字符串工具类(StringUtils)**：定义了针对字符串的若干静态方法
* **对象工具类(ObjectUtils)**：定义了用于处理一不安对象的若干静态方法
* **日期工具类(DateUtils)**：定义了针对日期的若干静态方法
* **JSON生成器(JSONGenerator)**：提供了将java对象转换为JSON格式字符串的静态方法
* **加密支持工具类**：加密支持工具类存放在com.jinxinol.core.util.encription包下，在encription包中有Base64编码解码工具类(Base64Utils)、对称DES加密工具类(DESUtils)、MD5摘要工具类、消息摘要工具类(MessageDigestUtils)。
* **HTTP请求/相应处理工具**：提供对Http请求和相应的处理

在开始编码前，开发人员需要了解核心包中提供的工具类。为了保持代码的整洁度和一致性，开发人员应该优先统一使用核心包中提供的工具支持类。在WEB应用编码过程中，会经常对字符串和日期对象进行判断或处理，此时开发人员应该首先查看核心包中的StringUtils和DateUtils是否已经提供了相应的方法。

#### WEB层通用支持类

在SSM框架核心包中定义了WEB层通用支持类：

com.jinxinol.core.common.BaseAction

BaseAction继承了ActionSupport，并实现了ServletRequestAware和ServletResponseAware接口。BaseAction的类图结构如下：



* ActionSupport：来自Struts2提供的Action支持类；
* ServletRequestAware：通过实现该接口，将HttpServletRequest注入到Action实例对象中；
* ServletResponseAware：通过实现该接口，将HttpServletResponse注入到Action实例对象中；
* 支持JSON结果对象返回：支持Ajax请求JSON数据格式返回；

使用SSM框架开发WEB应用时，所有WEB Action 都需要继承BaseAction。

#### 实体抽象类

在SSM框架中定了实体抽象类：

com.jinxinol.core.common.Entity

抽象实体类Entity实现了可序列化接口java.io.Serializable，Entity类图结构如下所示：



Entity具有两个用于支持分页查询的属性：pageSize和pageNo。pageSize每页显示的数据记录数，默认值为10。pageNo当前的页码，默认值为0。

使用SSM框架开发WEB应用时，所有实体类都需要继承抽象实体类Entity。

#### Service层通用支持类

在SSM框架核心包中定义了Service通用支持类：

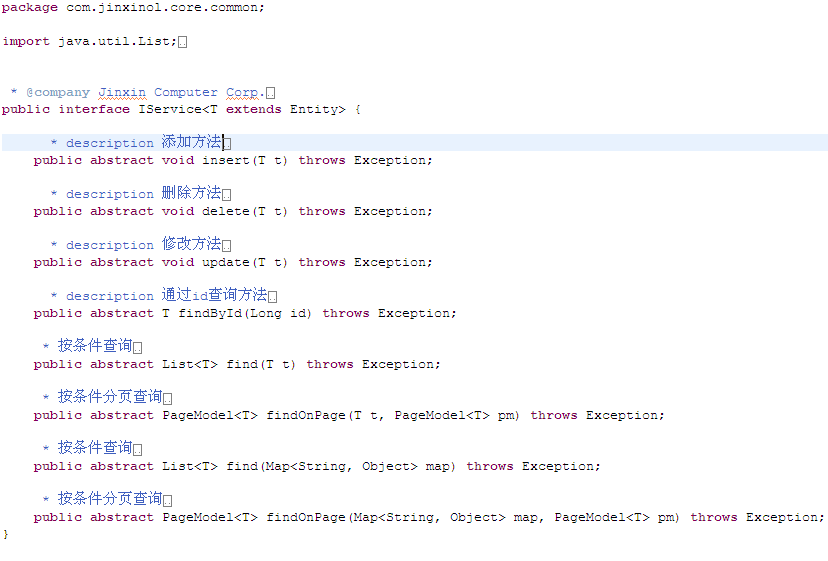
com.jinxinol.core.common.IService

com.jinxinol.core.common. AbstractService

通用业务服务接口IService定义了通用的业务层方法：对象的添加、对象的删除、对象的更新、对象的查询。下图是IService和AbstractService的实现关系：



定义IService时加入了泛型支持<T extends Entity>，如下所示：



在IService定义的查询方式3类：

* 根据实体ID查找实体对象：T findById(Long id)
* 根据传入查询条件查询满足条件的结果集：
  + - * 传入实体查询条件：List<T> find(T t)
      * 传入map查询条件：List<T> find(Map<String, Object> map)
* 根据传入查询条件进行分页查询
  + - * 传入实体查询条件：List<T> findOnPage(T t, PageModel<T> pm)
      * 传入map查询条件：List<T> findOnPage(Map<String, Object> map, PageModel<T> pm)

使用SSM框架开发WEB应用时，所有业务类都需要继承AbstractService抽象类。

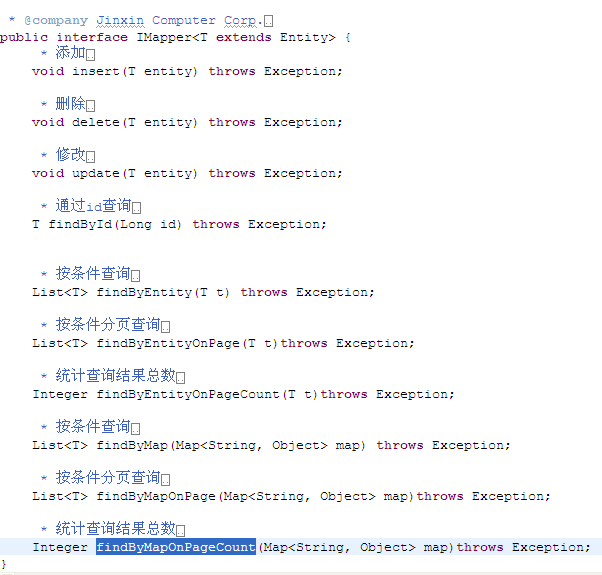
#### 数据访问层通用支持类

在SSM框架核心包中定义了数据访问通用支持类：

com.jinxinol.core.common.IMapper

通用数据访问接口IMapper定义了通用的DAO方法：对象的添加、对象的删除、对象的更新、对象的查询。

定义IMapper时加入了泛型支持



使用SSM框架开发WEB应用时，所有数据访问接口都需要继承IMapper接口。

#### WebService

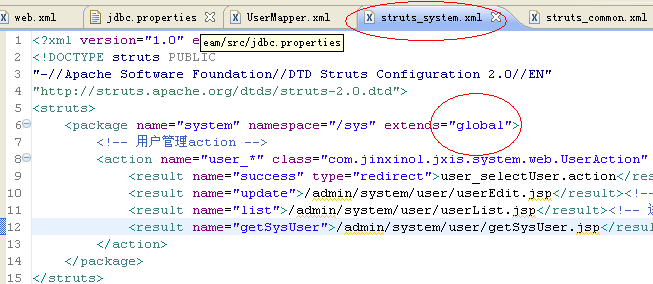
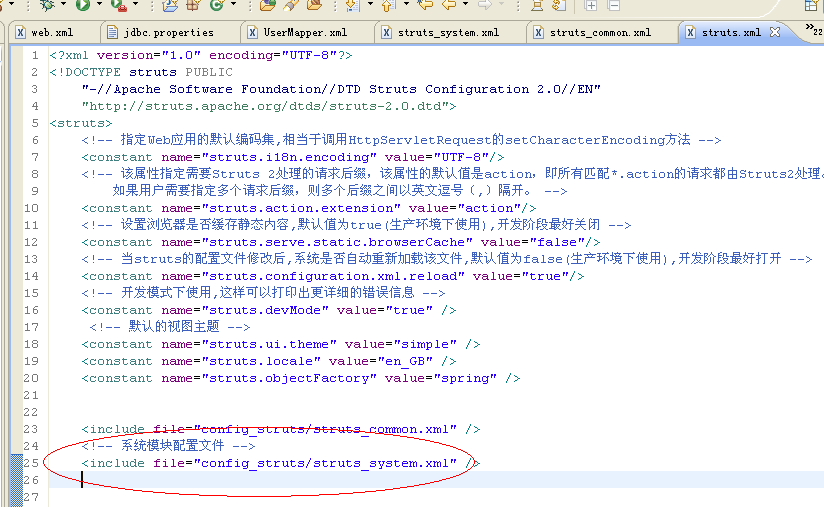
## 配置文件介绍

### 5.1 struts.xml

struts.xml配置文件是Struts框架的核心配置文件，该文件主要负责管理Struts 2框架的业务控制器Action，该文件位于开发环境中的src目录下。

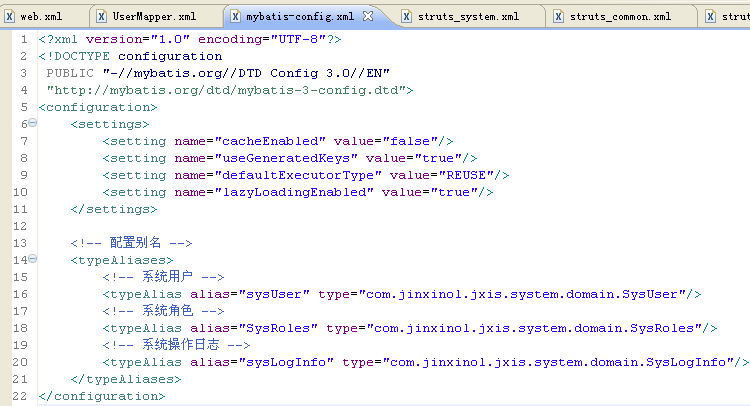
为了便于开发及结构清晰，在项目中的各个模块或者功能都会单独建立一个配置文件，然后引入到struts.xml主配置文件中，目录位于src. config\_strutss，每个配置文件都会继承global。

如下图所示：



### 5.2 mybatis-config.xml

mybatis-config.xml是Mybatis的主配置文件之一，主要作用域Mapper的sql返回映射及mybatis的基础配置等。

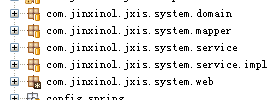


## 开发实例

### 6.1 基础准备

#### 6.1.1 建立自己功能模块的包结构

s



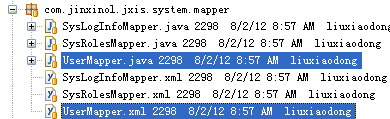
#### 6.1.2 根据数据结构建立DOMAIN类

如果需要分页需继承Entity类。如下图：s

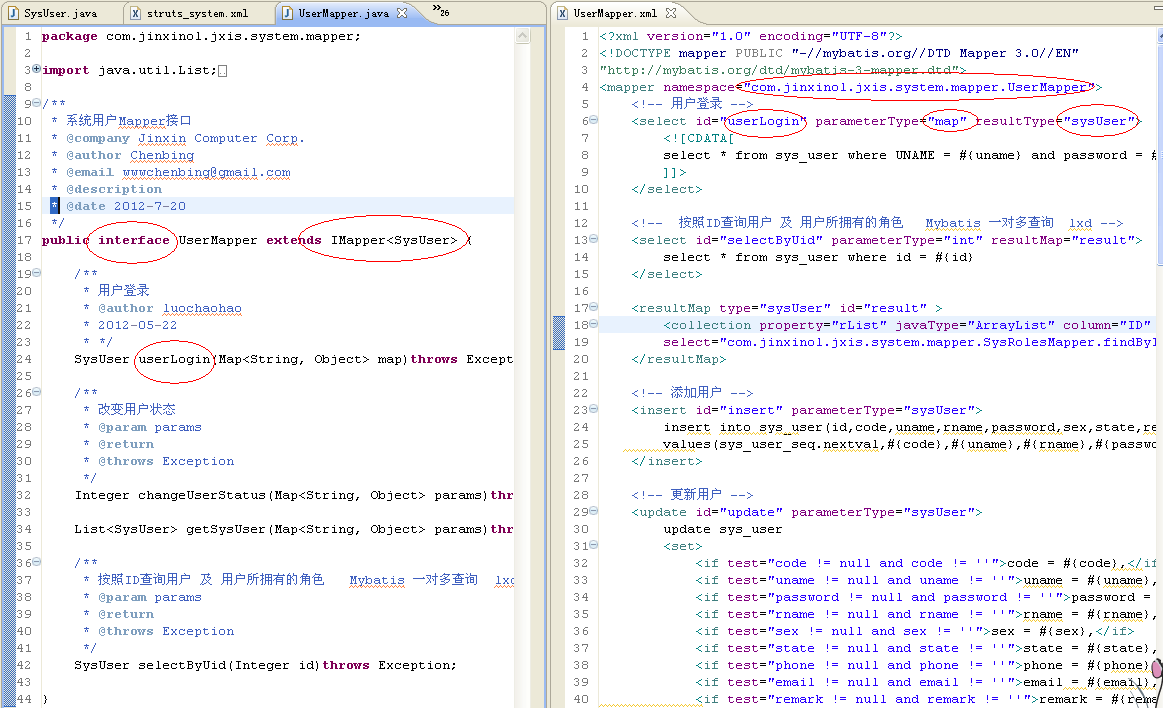


#### 6.1.3 根据业务建立数据访问层

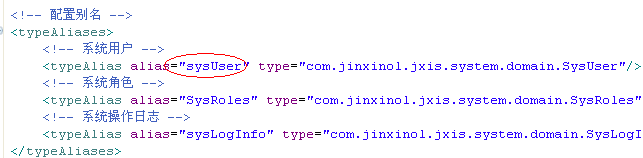
数据访问层对应两个文件：Mapper.java、Mapper.xml。



编写规范如下：

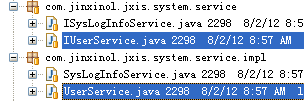


这里会进行配置Mybatis\_config.xml。

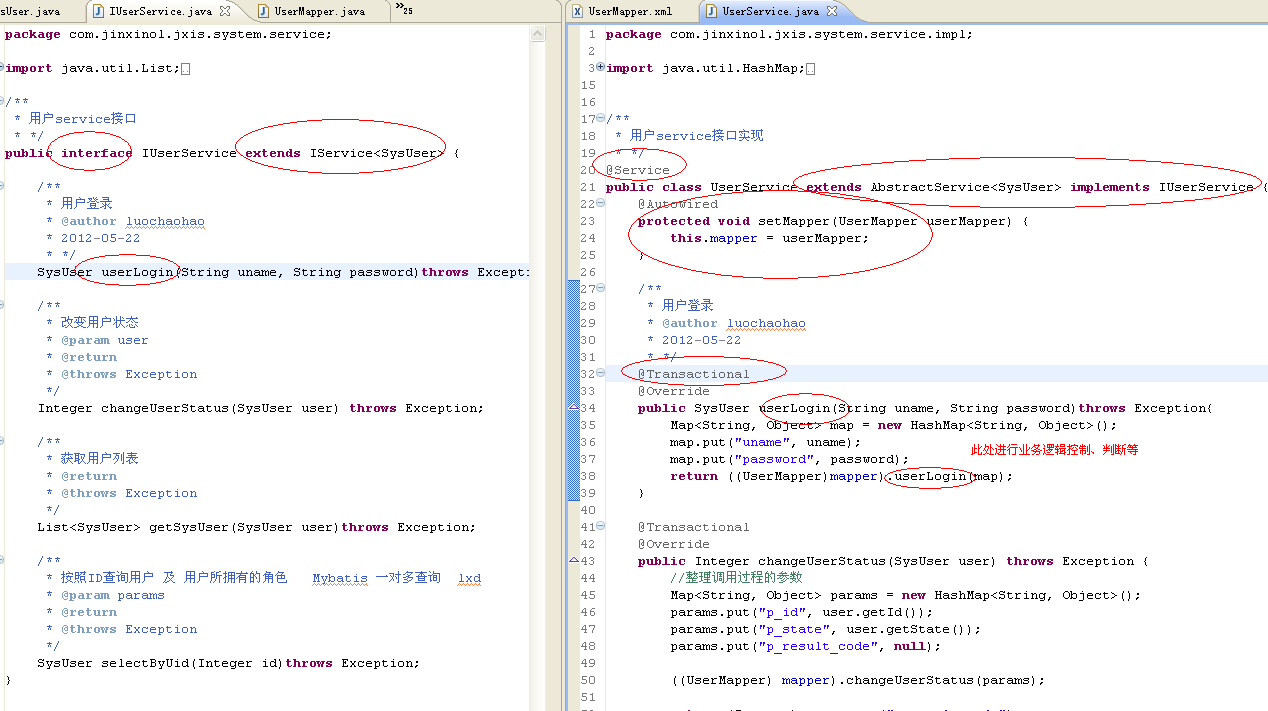


#### 6.1.4 根据业务建立业务逻辑层

业务逻辑层对应两个文件：IService.java、Service.java。



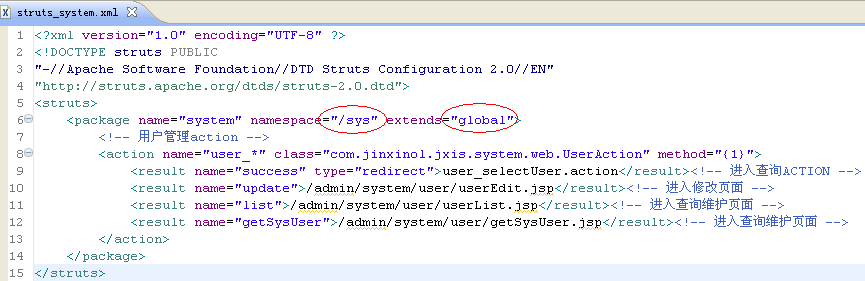
编写规范如下：



#### 6.1.5 建立Struts的Action控制层

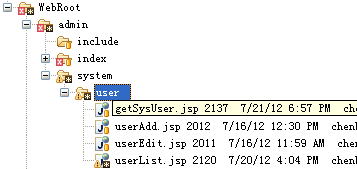


Action涉及到Struts.xml文件的配置：



#### 6.1.6 前台View展示层

可以包含javascript、jsp等文件，目录结构如下：



### 6.2 增删改查数据

整体配置参考基础准备，主要业务是增删改查用户信息。

1. 前台页面jsp页面，form表单或请求路径的action为sys/user\_{Action方法名}.action，请求UserAction.java的{Action方法名}方法。

s

1. UserAction控制层负责接收页面传来的数据，然后传入UserService的{Action方法名}方法进行业务逻辑处理。

s

1. UserService逻辑控制层需要注入UserMapper数据处理类，UserService负责调用UserMapper提供的方法实现用户功能。
2. UserMapper数据访问层主要是mybatis会根据包名和方法名结合sping自动调用映射相应的SQL语句来执行数据库操作。
3. 上述执行完成后，会回到Action继续执行，struts返s回“字符串”，根据struts\_system.xml配置文件找到<result name="字符串">，跳转到相应的页面或者Actiossn结束操作。

### 6.3 分页查询数据

整体配置参考基础准备，主要业务是根据相应信息查询用户信息列表分页。

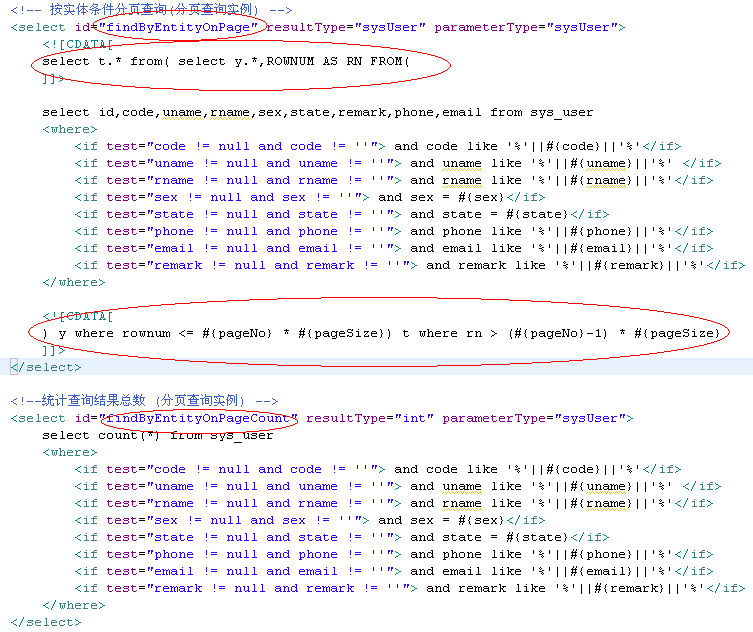
1、JSP页面请求。



2、ACTION控制层：



3、Mapper.xml配置文件的sql如下：



4、View视图层

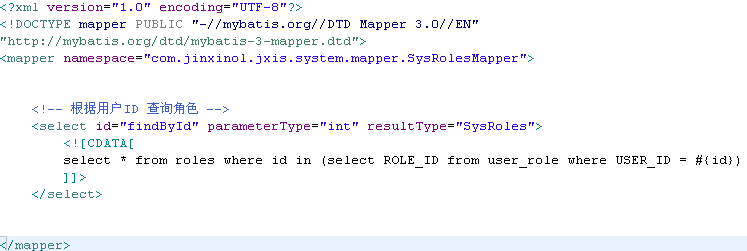


### 6.4 Mybatis一对多关联查询

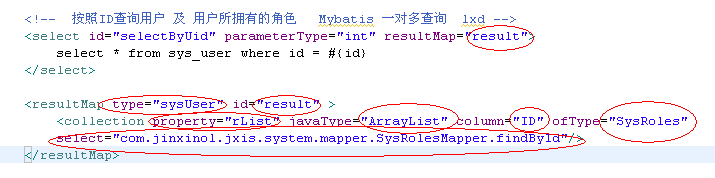
整体配置参考基础准备，主要业务是根据用户ID查询用户信息及所关联所拥有的角色信息。

1、总体跟前面的配置操作一样，主要是xml的配置文件不一样，一个xml文件调用另一个xml配置文件的sql语句。

2、首先配置用户角色的DOMAIN、MAPPER、mybatis\_config.xml等相关文件。



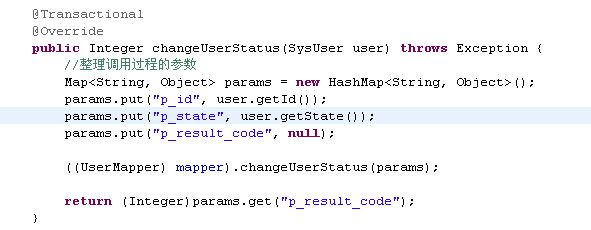
3、UserMapper.xml配置文件如下：



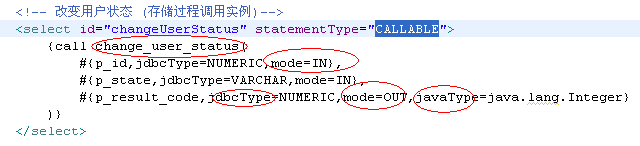
### 6.5 调用存储过程返回单个结果

整体配置参考基础准备，主要业务是调用存储过程改变用户的状态，传入参数2个（用户ID、用户状态），输出参数一个。

1. 总体配置操作跟前面一样，主要是xml的sql语句写法不一样，写法是调用存储过程的写法。
2. Service的写法：



1. Mapper.xml的写法：



调用存储过程statementType=“CALLABLE”

{call 存储过程名（

#{参数1，jdbcType=类型，mode=参数类型},

#{参数2，jdbcType=类型，mode=参数类型}

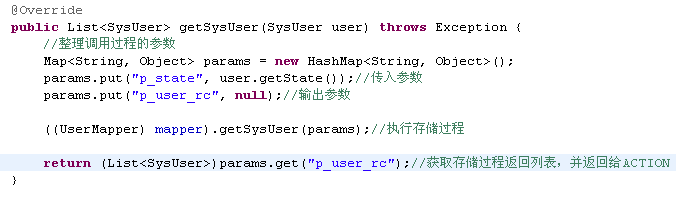
）

}

### 6.6 调用存储过程返回结果集

整体配置参考基础准备，主要业务是调用存储过程查询用户的一个结果集列表。

1. 总体配置操作跟前面一样，主要是xml的sql语句写法不一样，写法是调用存储过程的写法。
2. Service的写法：



1. Mapper.xml的写法

