

# 校内注册系统结构

## 1 概述

校内注册系统日方向在 58 万，所以本身对从性能上来讲，对机器的压力并不是很大。但是注册系统本身业务来讲更新是比较频繁，并且改动从注册流程上来讲可能是脱胎换骨。所以整个系统结构围绕着易于维护，易于更新简建立。

注册系统可以分为：注册 web 服务，和注册远程调用服务。其中注册 web 服务是为实际用户通过浏览器的 (<http://reg.xiaonei.com/>)。注册远程调用服务是为内部机器远程调用使用的 (<http://services.reg.xiaonei.com/> 只能内网访问)。

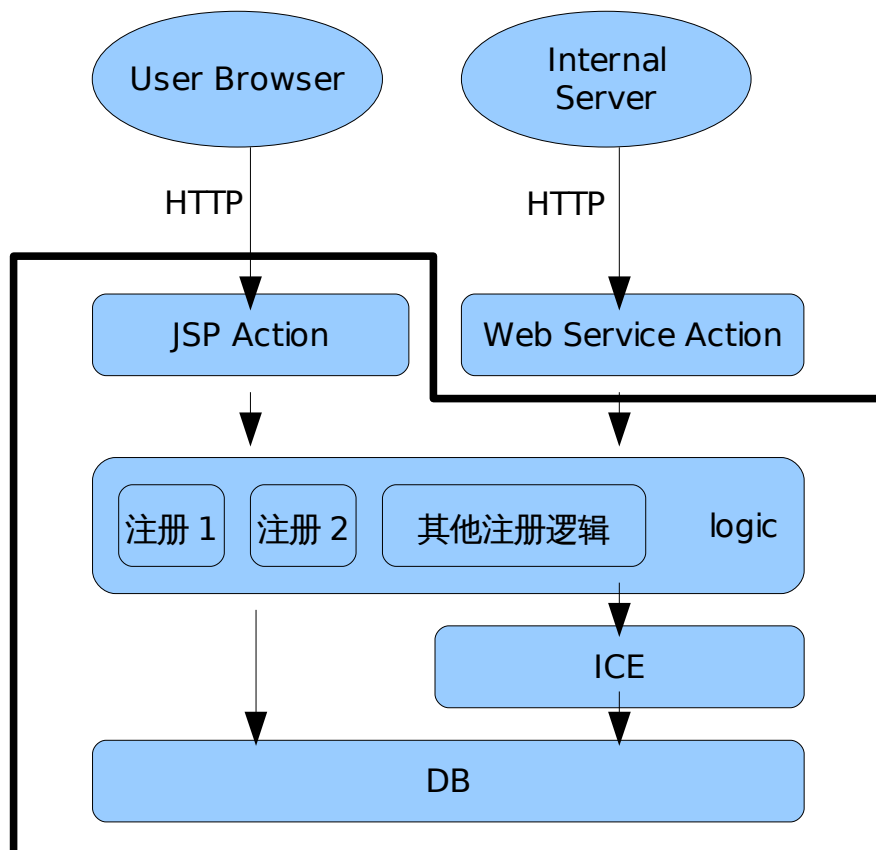
主要遵循 DRY 和 KISS 原则，建立注册系统。

## 2 目的

本文档说明的主要目的是，为了说明校内现在的注册系统结构，方便其他人员了解校内注册系统。

## 3 系统结构

注册系统的概要结构图如下：



Drawing 1: 校内注册系统结构概图

如上图所示，注册在使用统一套底层流程的情况下为外界提供两套调用接口。一套为普通用户使用，另外一套内部其他服务使用，包括：审核服务、手机 web 回掉注册服务。

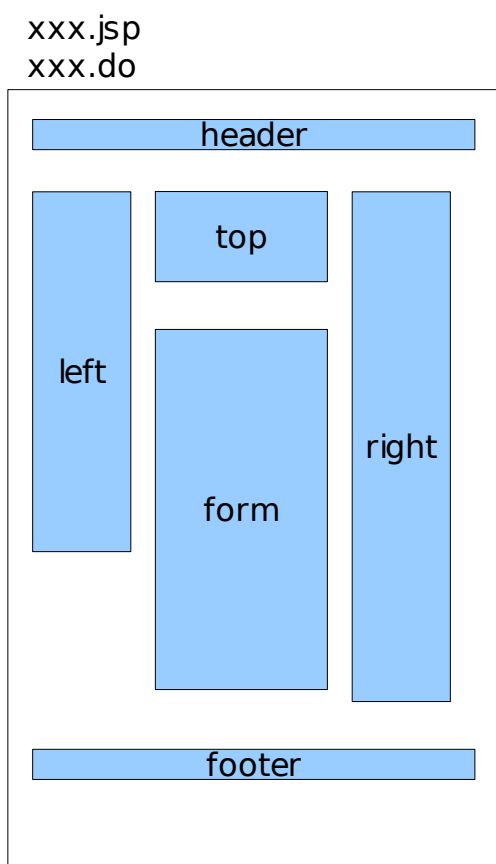
在 logic 层上事先封装好一些常用流程，在外部使用工厂方法调用。

## 4 子模块描述

注册系统作为一个整体，是由以下小子模块组成的：view,form,validation,analyze-url,init-page-act,submit-act,regFactory, audit,activate。

### 4.1 view(jsp)

注册页面统一布局,如下图所示



统一页面框架方便嵌套页面，扩展性好。以后需要时可以为产品方便的制作一个后台完全有产品人员调整页面

```
<div class="full-page flash-invite">
  <!-- begin:left -->
    <%@ include file="" %>
  <!-- begin:left -->
  <!-- begin:middle -->
  <div class="form-register">
    <div class="top_info clearfix">
      <div class="noinv_message">
      </div>
    </div>
  </div>
```

```

<!-- regForm -->
    <%@ include file="/inc/errors.inc" %>
    <%@ include file="" %>
<!-- regForm -->
</div>
<!-- begin:middle -->
<!-- begin:right -->
    <%@ include file="" %>
<!-- end:right-->
</div>

```

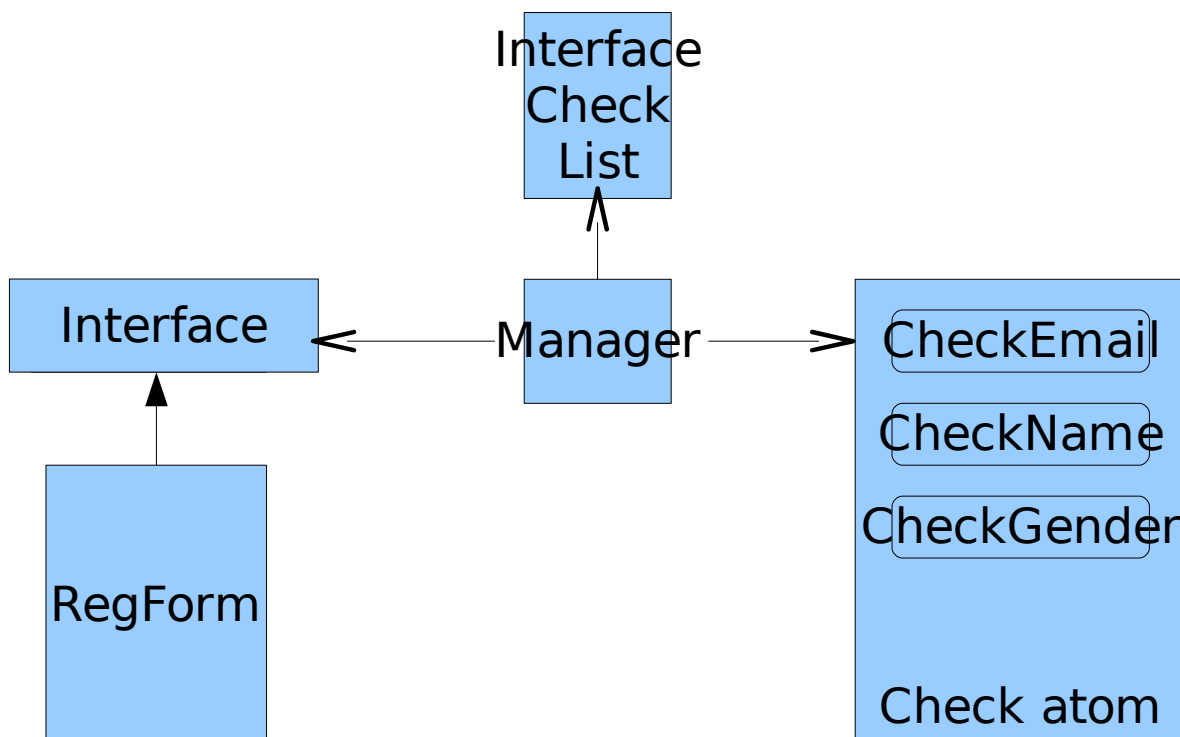
## 4.2 form validation

后台表单验证框架为了灵活考虑，没有使用 struts 结构。通过检查 form 的实现接口类型，数据校验列表动态产生。

所有的 form 都要继承一个 BaseRegForm，并且继承若干接口，这些接口中有单个对应元素的 Setter 和 Getter 方法。BaseRegForm 中是一个大而全的属性集合，覆盖到了所有页面。

通过一个 RegCheckManager 比较 form 接口，动态校验数据。

如下图所示：



com.xiaonei.reg.usertrace.logic.UserTraceLogic 这样好处是简化了新的表单类型的数据校验，只需根据页面上出现的数据建立一个新的 ActionForm 让它去实现对应的接口，数据校验的工作就不用 2 次开发了。

```

public class RegCommForm extends BaseRegForm implements IStage, IAccType,
    IRegEmail, INicknameId, IPwd, IName, IGender, IHomecity,
    IUniversity,
    IMiddleschool {
    ... ..
}

```

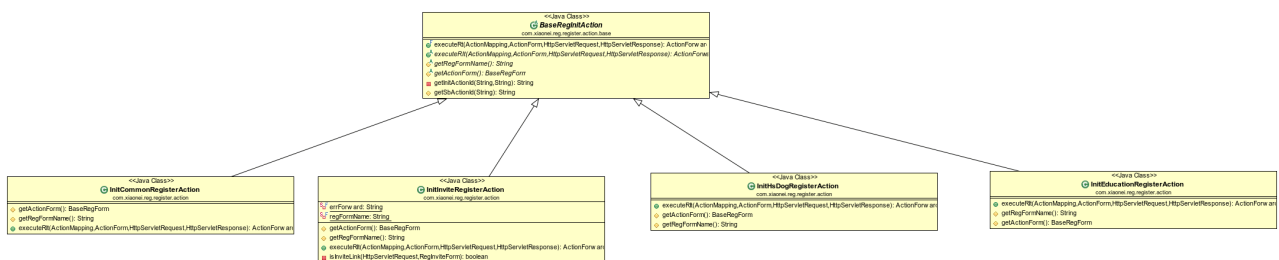
## 4.3 analyze-url

根据 url 参数，分析出注册连接类型指向正确的页面。这个功能是在调链系统之前实现的，现在有了调链系统，这个功能作为兼容老连接，并且以后会逐步淘汰

## 4.4 init-page-act

初始化注册页面和提交注册表单 action 分离，做到 one action,one thing，简化逻辑。并且针对 init page action 和 submit action 分别抽象出各自的共通业务，减少 2 次开发成本，和 bug 数量。

BaseInitAction 中完成了退出在线用户，记录访问页面 log。在子类里面标明需要跳转的页面就可以完成就基本的页面初始化了。



```
protected abstract String getRegFormName();
```

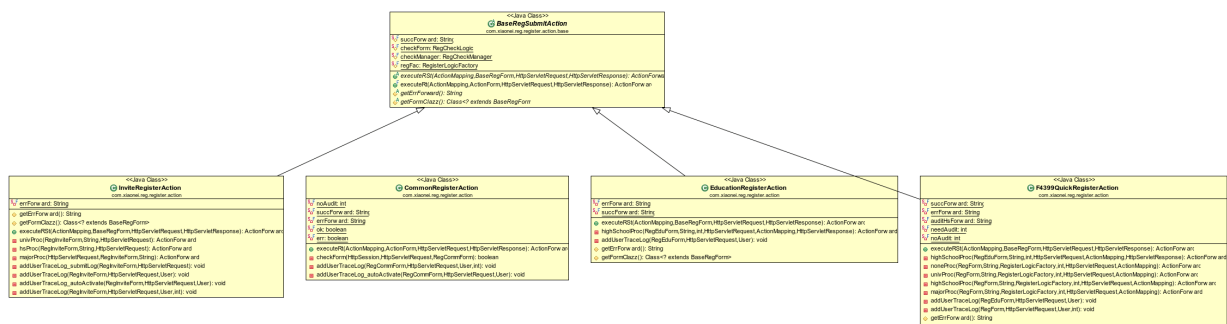
```
protected abstract BaseRegForm getActionForm();
```

这两个方法是子类需要实现的，指定对应的ActionForm

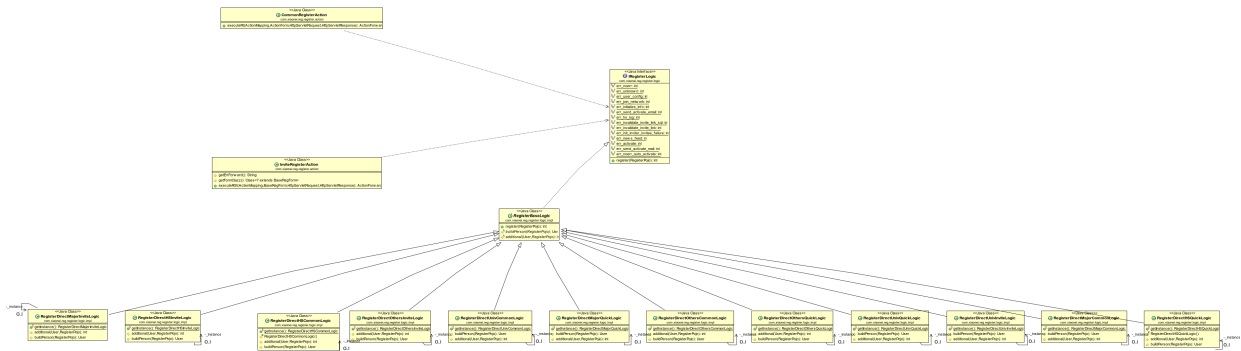
## 4.5 submit-act

在 baseRegSubmitAction 中完成了 validation 方法的调用，和页面默认的跳转

子类只需要实现对应方法的调用。



## 4.6 RegFactory



上图最下面一行事  $n$  种注册流程，实现了统一的接口，通过 `factory` 选取，每种流程是由若干 `atom logic` 组合而成。

## 4.7 Audit

注册服务与审核服务通信使用数据库和 http 协议，注册时信息保存到审核库中。审核结束后审核结果使用密文通过 Http 协议发送给注册服务，注册服务做相应操作，注册用户。

## 4.8 mobile

手机注册，如果用户填写完注册信息后，会发送用户一下行短信。用户可以吧短信的中验证码，回写到 web 页面上。验证成功后成功注册。

用户也可以根据页面上的提示进行上行短信注册，填写完验证信息后上行短信到 9318xxx，手机段同样调用注册的服务 `servivces.reg.xiaonei.com`，注册。

## 4.9 Activate

激活用户可以有以下方式，用户点击邮件中激活连接激活，符合自动激活条件的用户系统自动激活，用户发送邮件到 [admin@xiaonei.com](mailto:admin@xiaonei.com) 激活，管理员后台激活。为了保证代码统一，激活代码只有一份，然后提供出来若干类型接口供外部调用。

连接激活，与自动激活调用内部接口。邮件激活和管理员激活调用 http 接口访问 `servives.reg.xiaonei.com`。

激活除了修改用户标志位，还要做一些附加业务，如，加好友，加邀请奖励等。