在Web应用程序中，***为了防止客户端传来的数据引发程序的异常***，常常需要对数据进行验证。输入验证分为***客户端验证***与***服务器端验证***。客户端验证主要通过JavaScript脚本进行，而服务器端验证则主要是通过Java代码进行验证。

为了保证数据的安全性，一般情况下，客户端验证与服务器端验证都是要进行的。

我们这里所讲的是SpringMVC在***服务端***是如何对数据进行验证的。

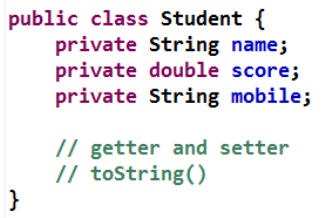
需求：要求用户输入的表单数据满足如下要求：

* 姓名：非空，且长度3-6个字符
* 成绩：0-100分
* 手机号：非空，且必须符合手机号格式

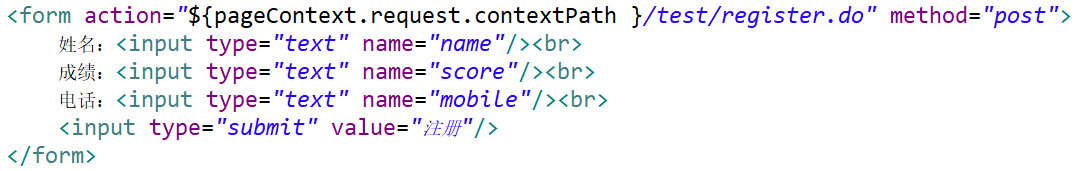
1. **搭建测试环境**
2. 导入Jar包

SpringMVC支持***JSR（Java Specification Requests，Java规范提案）303-Bean Validation数据验证规范***。而该规范的实现者很多，其中较常用的是Hibernate Validator。需要注意的是，Hibernate Validator是与Hibernate ORM并列的Hibernate的产品之一。这一点从Hibernate官网上所提供的资源形式可以看出它们之间的关系。

1. 定义实体



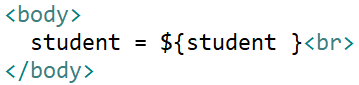
1. 定义index页面



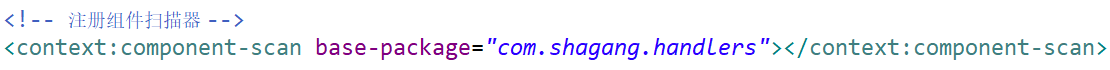
1. 定义Controller



1. 定义页面



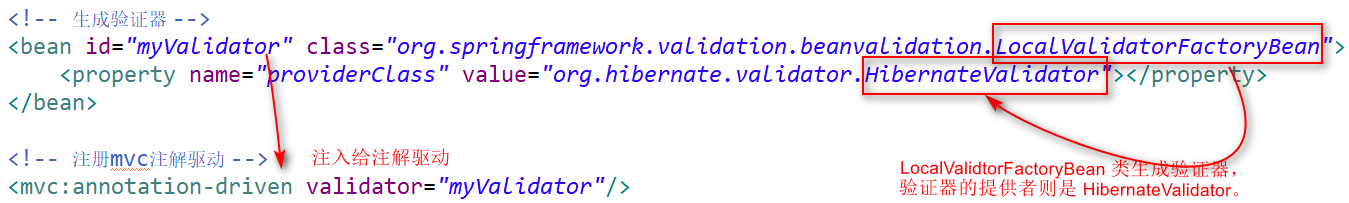
1. 定义SpringMVC配置文件



1. **实现数据验证**
2. 修改springmvc配置文件

验证器由SpringMVC框架的LocalValidtorFactoryBean类生成，而真正验证器的提供者则是HibernateValidator。

在SpringMVC配置文件中将验证器注册后，需要将其注入给注解驱动。



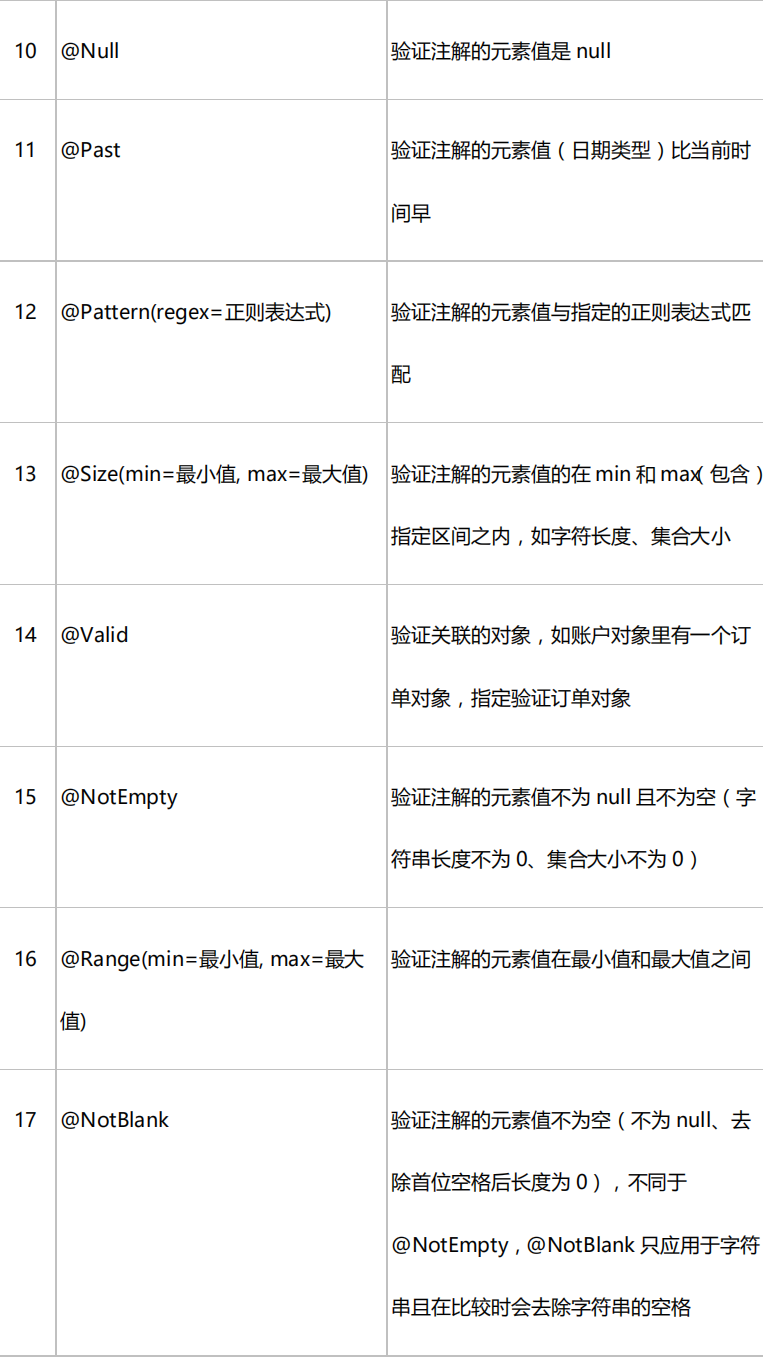
1. 在实体属性上添加验证注解

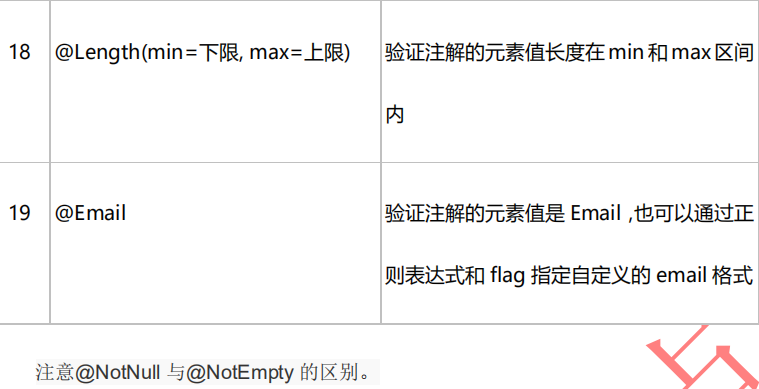
使用的验证器注解均为***javax.validation.constraints包中的类***。在注解的message属性中，可以使用***{属性名}***的方式来引用指定的注解的属性值。



Hibernate Validator中常用的验证注解介绍：







1. 修改Controller
2. 页面显示验证异常信息