

## 1-5 BO数据验证

### 数据验证

#### 1. 验证Bean：

```
public class ValBean {

    /**
     * Bean Validation 中内置的 constraint
     * @NotNull 被注释的元素必须为 null
     * @NotNull 被注释的元素必须不为 null
     * @AssertTrue 被注释的元素必须为 true
     * @AssertFalse 被注释的元素必须为 false
     * @Min(value) 被注释的元素必须是一个数字，其值必须大于等于指定的最小值
     * @Max(value) 被注释的元素必须是一个数字，其值必须小于等于指定的最大值
     * @DecimalMin(value) 被注释的元素必须是一个数字，其值必须大于等于指定的最小值
     * @DecimalMax(value) 被注释的元素必须是一个数字，其值必须小于等于指定的最大值
     * @Size(max=, min=) 被注释的元素的大小必须在指定的范围内
     * @Digits (integer, fraction) 被注释的元素必须是一个数字，其值必须在可接受的范围内
     * @Past 被注释的元素必须是一个过去的日期
     * @Future 被注释的元素必须是一个将来的日期
     * @Pattern(regex=, flag=) 被注释的元素必须符合指定的正则表达式
     * Hibernate Validator 附加的 constraint
     * @NotBlank(message =) 验证字符串非null，且长度必须大于0
     * @Email 被注释的元素必须是电子邮箱地址
     * @Length(min=,max=) 被注释的字符串的大小必须在指定的范围内
     * @NotEmpty 被注释的字符串的必须非空
     * @Range(min=,max=,message=) 被注释的元素必须在合适的范围内
     */
    private Long id;

    @Max(value=20, message="{val.age.message}")
    private Integer age;

    @NotBlank(message="{username.not.null}")
    @Length(max=6, min=3, message="{username.length}")
    private String username;

    @NotBlank(message="{pwd.not.null}")
    @Pattern(regexp="/^(?![0-9]+$)(?![a-zA-Z]+$)[0-9A-Za-z]{6,10}$/", message="密码必须是6~10位数字和字母的组合")
    private String password;

    @Pattern(regexp="^((13[0-9])|(15[^4,\\D])|(18[0,5-9]))\\d{8}$", message="手机号格式不正确")
    private String phone;

    @Email(message="{email.format.error}")
    private String email;
}
```

#### 2. 验证Controller：

```
@Controller
@RequestMapping(value = "/val")
public class ValidateController {

    @RequestMapping(value = "/val", method=RequestMethod.POST)
    @ResponseBody
```