软件项目文档规范

重庆英卡电子有限公司

一、前端验证

1. 导入 js 框架

```
<script src="https://cdn.cqwo.com/js/jquery-1.7.1.min.js"></script>
<script src="https://cdn.cqwo.com/js/jquery.validate.min.js"></script>
<script src="https://cdn.cqwo.com/js/jquery.validate.unobtrusive.min.js"></script>
```

2. 验证规则

(1) 开启验证

```
data-val="true"
```

(2) 数据必须填项目

```
data-val-required="XXX 为必须项目"
```

(3) 取值范围

```
data-val-range-max="100" data-val-range-min="0"
data-val-range="从最小值到最大为 10000"
```

(4) 验证类型

```
data-val-number="字必须输入个数字。"
data-val-email=”必须输入正确格式的电子邮件”
data-val-url=”必须输入正确格式的网址”
data-val-date=”必须输入正确格式的日期”
data-val-digits=”必须输入正整数”
data-val-number=”必须输入整数”
```

(5) 验证相同

```
data-val-equalto="密码和确认密码不匹配。"
data-val-equalto-other="password"
```

(6) 验证字符长度

```
data-val-length-min="6"
data-val-length-max="100"
data-val-length="密码必须至少包含 6 个字符。"
```

(7) 验证正则表达式

```
data-val-regex-pattern="(13[0-9]|15[012356789]|18[0236789]|14[57])[0-9]{8}"
正则的表达式
data-val-regex="手机格式不正确" 表示正则的验证方式，
```

(8) Ajax 验证

```
data-val-remote="已经被注册，请选择其他登录名称！"
data-val-remote-url="checkuserid.aspx"  验证验证 url
```

实例: <input type=”text” id=”uid” name =”uid” value=”” data-val="true" data-val-required="XXX 为必须项目" data-val-number="字必须输入个数字。" data-val-range-max="100" data-val-range-min="0" data-val-range="从最小值到最大为 10000"/>

3. 验证规则提示

```
<span class="field-validation-valid" data-valmsg-for="xx" data-valmsg-replace="true"></span>
```

data-valmsg-for 替换成自己的 id

二、后端验证

1. 需要导入的包

```
<!-- Hibernate Validator -->
<dependency>
   <groupId>org.hibernate</groupId>
   <artifactId>hibernate-validator</artifactId>
</dependency>
```

2. Controller 使用方法

```
//使用 ValidateModel 验证 model 模型
ValidationResult result=ValidateModel.validateEntity(model);
if (result.isNotErrors()){    //判断验证模型是否通过
      ....
}

return PromptView(result.toString());    //result.toString()为错误列表
```

3. Model 注解规则

验证注解	验证的数据类型	说明
@AssertFalse	Boolean,boolean	验证注解的元素值是 false
@AssertTrue	Boolean,boolean	验证注解的元素值是 true
@NotNull	任意类型	验证注解的元素值不是 null
@Null	任意类型	验证注解的元素值是 null
@MIN(value=值)	BigDecimal，BigInteger, byte,short, int, long，等任何 Number 或 CharSequence (存储的是数字）子类型	验证注解的元素值大于等于@Min 指定的 value 值
@MAX(value=值)	和@Min 要求一样	验证注解的元素值小于等于@Max 指定的 value 值
@DecimalMin(value=值)	和@Min 要求一样	验证注解的元素值大于等于@ DecimalMin 指定的 value 值
@DecimalMax(value=值)	和@Min 要求一样	验证注解的元素值小于等于@ DecimalMax 指定的 value 值
@Digits(integer=整数位数, fraction=小数位数)	和@Min 要求一样	验证注解的元素值的整数位数和小数位数上限
@Size(min=下限, max=上限)	字符串、Collection、Map、数组等	验证注解的元素值的在 min 和 max（包含）指定区间之内，如字符长度、集合大小
@Past	java.util.Date,java.util.Calendar;Joda Time 类库的日期类型	验证注解的元素值（日期类型）比当前时间早
@Future	与@Past 要求一样	验证注解的元素值（日期类型）比当前时间晚
@NotBlank	CharSequence 子类型	验证注解的元素值不为空（不为 null、去除首位空格后长度为 0），不同于@NotEmpty，@NotBlank 只应用于字符串且在比较时会去除字符串的首位空格
@Length(min=下限, max=上限)	CharSequence 子类型	验证注解的元素值长度在 min 和 max 区间内
@NotEmpty	CharSequence 子类型、Collection、Map、数组	验证注解的元素值不为 null 且不为空（字符串长度不为 0、集合大小不为 0）
@Range(min=最小值,	BigDecimal,BigInteger,CharSequ	验证注解的元素值在最小值和最大值之间

验证注解	验证的数据类型	说明
max=最大值)	ence, byte, short, int, long 等原子类型和包装类型	
@Email(regexp=正则表达式,flag=标志的模式)	CharSequence 子类型（如 String）	验证注解的元素值是 Email，也可以通过 regexp 和 flag 指定自定义的 email 格式
@Pattern(regexp=正则表达式,flag=标志的模式)	String，任何 CharSequence 的子类型	验证注解的元素值与指定的正则表达式匹配
@Valid	任何非原子类型	指定递归验证关联的对象；如用户对象中有个地址对象属性，如果想在验证用户对象时一起验证地址对象的话，在地址对象上加@Valid 注解即可级联验证

4. 实例

```
/**
 * 验证码
 */
@NotNull(message="验证码不能为空")
@Length(min=5, max=10, message="验证码长度必须在 5-10 之间")
private String verifyCode="";
```

三、省市下拉列表用法

1. 导入 js 框架

```
<script type="text/javascript" src="xxx/region.js"></script>
```

2. H5 引用

```
<select class="btn btn-default" id="provinceSelect">
    <option value="-1">请选择</option>
</select>
<select class="citySelect btn btn-default" id="citySelect">
    <option selected="selected" value="-1">请选择</option>
</select>
<select class="countySelect btn btn-default" id="regionId" name="regionId">
    <option selected="selected" value="-1">请选择</option>
</select>
```

regionId 可自行修改成自己的字段

3. 默认值

```
var provinceId = 500000; //省 id
var cityId = 500100; //市 id
var countyId = 500233; //县或区 id
```

四、图片（文件上传）上传

1. JAVA 后台代码 示例

```
-/**
 * 新闻图片上传
 *
 * @return
 */
public UploadFileInfo UploadNewsImage() {

	List<UploadFileInfo> uploadFileInfoList=fileUpload("news");
	if (uploadFileInfoList.size() >= 1) {
		return uploadFileInfoList.get(0);
	}
	return null;
}
```

红色标注部分为 upload 中子文件夹名称

2. 端 html 单文件上传

(1) Html 代码示例

```
<input class="form-control text" id="litpic" name="litpic" type="hidden" value="${Model.litpic}"/>
<span class="field-validation-valid" data-valmsg-for="litpic" data-valmsg-replace="true"></span>
<input class="file" type="file" data-max-file-count="1" id="litpicfile" name="litpicfile" multiple>
```

红色部分代表上传的图片标签

(2) JS 代码示例

```
<script type="text/javascript">
	$("#litpicfile").fileinput({
		language: "zh",
		dataType: "json",
		showUpload: false,
		showPreview: false,
		uploadUrl: "/admin/tool/upload.do?operation=uploadnewsimage",
		allowedFileExtensions: ["png", "jpg"]
	}).on("filebatchselected", function (event, files) {
		$(this).fileinput("upload");
	}).on("fileuploaded", function (event, data) {

		var result = eval("(" + data.response + ")");

		if (result) {
			var state = result.State;

			if (state == 1) {
				var imgResult = result.Content;
				if (imgResult === "-1") {
					alert("图片上传文件失败");
					return;
				} else if (imgResult === "-2") {
					alert("图片上传文件类型不正确,请上传 png 或者 jpg 图片");
					return;
				} else if (imgResult === "-3") {
					alert("图片上传文件过大,感谢亲的理解");
					return;
				}
				$("#litpic").val(result.Content);

				alert("新闻图片上传成功");
			} else {
				alert(result.Content);
			}
		}
	});
</script>
```

(3) 编辑器代码示例

```
<textarea class="form-control text" cols="20" data-val="true"
data-val-length="新闻长度不能大于 5000" data-val-length-max="5000"
id="body" name="body" placeholder="新闻详情" rows="2">${Model.body}</textarea>

<script type="text/javascript">
        window.UMEDITOR_CONFIG.imageUrl = "" +
"/admin/tool/upload.do?operation=uploadnewseditorimage";
        window.UMEDITOR_CONFIG.initialFrameHeight = "300";
        window.UMEDITOR_CONFIG.initialFrameWidth = "880";
        UM.getEditor('body');

</script>
```

3. 确认删除

onclick="return confirm('确认删除？')"

五、编辑器的使用(UEDITER)

1. 引入 JS 和 css 样式

```
<script type="text/javascript" src="/components/ueditor/ueditor.config.js"></script>
<script type="text/javascript" src="/components/ueditor/ueditor.all.min.js"></script>
<!--建议手动加在语言，避免在 ie 下有时因为加载语言失败导致编辑器加载失败-->
<!--这里加载的语言文件会覆盖你在配置项目里添加的语言类型，比如你在配置项目里配置的是英文，这里加载的中文，那最后就是中文-->
<script type="text/javascript" src="/components/ueditor/lang/zh-cn/zh-cn.js"></script>
```

2. 初始化插件

```
<textarea id="editor" name=”body” type="text/plain" style="width:800px;height:500px;">我喜欢你
</textarea>
<script type="text/javascript">
    window.UEDITOR_CONFIG.serverUrl = "/admin/tool/ueditor"; //后台编辑器配置
    //实例化编辑器
    //建议使用工厂方法 getEditor 创建和引用编辑器实例，如果在某个闭包下引用该编辑器，直接调用 UE.getEditor('editor')就能拿到相关的实例
    var ue = UE.getEditor('editor');
```

3. 前端文件上传配置

```
    UE.Editor.prototype._bkGetActionUrl = UE.Editor.prototype.getActionUrl;
    UE.Editor.prototype.getActionUrl = function (action) {
        if (action === 'uploadimage' || action === 'uploadscrawl' || action === 'uploadimage') {
            return '/admin/tool/upload?operation=uploadnewseditorimage';
        } else {
            return this._bkGetActionUrl.call(this, action);
        }
    }</script>
```

4. 后端文件上传配置

```
/**
 * 图片上传
 *
 * @param operation
 * @return
 */
@RequestMapping(value = "tool/upload")
@ResponseBody
public String upload(@RequestParam(defaultValue = "") String operation,
                     MultipartHttpServletRequest request) {

    Map<String, MultipartFile> multipartFileMap = request.getFileMap();

    if (multipartFileMap.size() <= 0) {
        return "-1";
    }

    //新闻
    if ("uploadnewseditorimage".equals(operation))//上传广告图片
    {
        List<UploadFileInfo> litpic = uploads.fileUpload("news", multipartFileMap);
        if (litpic.size() >= 1) {
            Map<String, String> params = new HashMap<>();
            params.put("state", "SUCCESS");
            params.put("url", litpic.get(0).getFileUrl());
            params.put("title", "生态热线");
            params.put("original", getUEState(litpic.get(0).getFileUrl()));
            return JSON.toJSONString(params);

        }
     }

    return “-1”;
}
```