2017.4.13

模板模块 校验模块 事件绑定模块

# 事件绑定 BindEvent

浏览器端JS 以事件驱动，因此，事件绑定在浏览器端是非常重要的功能，为了方便为基础组件绑定事件与指定回调函数。设计完成了事件绑定模块。

## 事件绑定模块原理

### 为了正确的进行事件绑定，在配置页面时需要配置 bind\* =“callbackName，triggerName” 和 listen =‘{“triggerName“：”callbackName“}’

### 在对自定义标签进行解析时，会解析bind\* 并将其转义为bindlist=“eventType-callbackName-triggerName, … ,…”转义后，将bindlist放置在正确的位置，正确的位置为 拥有hasbind=“true”的标签 ，如未找到hasbind属性的标签，则为当前解析标签。

### listen 为事件订阅配置，此配置项会配置在解析好的组件最外层。

### 所有解析过程完毕后，进行具体的事件绑定

## 事件绑定的具体过程。

### 标签检测自身是否含有bindlist属性，如含有，则将其放到解析后的dom最外层div上。

### 检测自身是否含有listen属性，若含有，则将其放到解析后的dom最外层上。

### 检测是否含有bind属性，若含有，则将其转义为bindlist属性并赋值给bindList变量。

### 检测解析后的dom元素子孙中是否含有hasbind=“true”的标签，若含有，则返回此边傲倩，若不含有，则返回最外层dom元素本身。

### 将第iii步的bindList与 第iv步返回元素中的bindlist属性合并。

### 重复i ii iii iv v步 直到所有标签解析完毕。

### 所有标签解析完毕后，遍历所有dom元素，找到含有listen属性的标签好含有bindlist属性的标签，进行具体的事件订阅与事件绑定。

### 事件绑定步骤完毕。

## 属性配置表格

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 值 | 类型 | 描述 |
| bind\* | “callbackName，triggerName” | string | 具体的事件配置，第一个配置项为回调函数名，第二个配置项为帆布消息名 |
| listen | ‘{“triggerName”：“callbackName”}’ | object | 事件订阅的配置，第一个为订阅的消息名，第二个位回调函数名 |

## bind\* 对照表格

|  |  |
| --- | --- |
| bind\* | 浏览器事件名 |
| bindcl | click |
| binddb | dbclick |
| bindfs | focus |
| bindfsi | focusin |
| bindfso | focusout |
| bindbl | blur |
| bindcg | change |
| bindku | keyup |
| bindkd | keydown |
| bindkp | keypress |

## API表格

|  |  |
| --- | --- |
| API | 功能 |
| BindEvent . bindEvent( $dom ) | 通过读取$dom上的bindlist属性为组件进行事件绑定 |
| BindEvent . setListener ( $dom ) | 通过读取$dom上的listen属性为组件进行事件订阅 |
| BindEvent . parseOptions ( $tag , $dom ) | 解析标签时的入口，此API的逻辑即事件配置解析时的逻辑。 |

## 注意事项

### 容器组件 禁止绑定事件 ，因为性能的考虑。

### 最好是在基础组件上绑定事件。

## 回调函数列表

### 为了更方便使程序猿们在进行事件订阅和事件绑定时，注重在业务逻辑，因此，将事件绑定列表 和 回调函数列表抽出。

### 绑定的事件列表 存放在 api目录下的common-event.js 中

### 订阅的回调函数列表 存放在api目录下的common-listen.js中

### 在需要固化 绑定事件的回调函数时，可打开common-event.js 在此列表中添加。如： 在evenList中添加函数， 回调函数中的 triggerName 形参，为发布消息名，可有可无。

### 

### 在需要固化 订阅的回调函数时，可打开 common-listen.js 在此列表中添加回调，如：在listenList中添加函数，params 为发布者带来的参数。

### 

### 当回调函数很特殊，不需要进行固化时，可使用框架提供的API来进行添加，添加函数 需要在绑定之前。API如下：

|  |  |
| --- | --- |
| API | 功能 |
| EventHandler .addHandler (eventName , function(triggerName){}) | 为 绑定事件列表 添加函数 |
| EventListenr .addListenr (handlerName, fn) | 为 回调函数列表 添加函数 |

### 事件绑定的回调函数中，this为触发此事件的Dom元素

### 订阅的回调函数中，this为订阅消息的Dom元素

# 校验模块 CheckHelper

校验模块，目的是为表单中的项目进行输入校验，复杂的校验，也可使用事件机制来进行。先仅支持对input teatarea的校验。

## 校验模块原理

### 为了正确的进行事件绑定，在配置页面时需要配置，如 regexp = “required，number-必须输入数字”typestyle=“after-blue”

### 在对自定义标签进行解析时，会解析regexp 并将其放置在正确的位置，正确的位置为 拥有hascheck=“true”的标签 ，如未找到haschec属性的标签，则为当前解析标签。

### 所有解析过程完毕后，进行具体的校验绑定

## 校验模块具体步骤

### 检测标签上是否含有regexp属性，若含有，则进行处理并进行第ii步，如没有，退出此标签的校验解析。

### 查找解析后的dom子孙中是否含有hascheck 属性的标签，如有，则返回此标签，若没有，则返回最外层的dom本身。

### 将第i步得到的regexp属性 覆盖到第ii步返回的dom上。

### 重复 i ii iii 的步骤 直到解析完毕

### 解析完毕后 查找页面中所有含有regexp 属性的标签。

### 获得的标签若为表单元素，则为此标签进行校验绑定，若不是，则虚招此标签下的所有表单元素标签进行绑定

### tipstyle的配置为校验提示样式，校验绑定时，会替换掉默认的校验提示的class属性。

### 校验步骤绑定完毕。

## 属性配置表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 值 | 类型 | 描述 |
| regexp | “required-校验提示” | string | 校验规则，第一个为校验规则，第二个为校验提示，当没有提示时，会启用默认提示。 |
| tipstyle | after-blue | string | 校验提示的class名，会覆盖校验提示的class |

## 已有的校验规则与默认提示

|  |  |  |
| --- | --- | --- |
| 规则名 | 默认提示 | 描述 |
| required | 必输项 | 必填类型的校验函数 |
| init | 请输入正整数 | 正整数类型的校验函数 |
| number | 输入不合法 | 正整数类型的正则 |
| lowerLetter | 输入不合法 | 消息字母的正则 |

## 校验时的步骤

### 检索校验函数列表中 是否含有匹配的函数，如有，则使用此函数进行校验，否则，进行第ii步。

### 校验正则表达式列表中是否含有匹配的内容，如有，则使用此正则来进行校验，否则，进行第iii步。

### 将传入的校验规则转为正则表达式进行校验。

## API表格

|  |  |
| --- | --- |
| API | 功能 |
| CheckHelper.bindCheck ( $dom ) | 为组件绑定校验，会搜索dom中的表单元素进行校验 |
| CheckHelper.checkForm ( $form ) | 为form表单进行表单校验 |
| CheckHelper.setTheme ( themeName ) | 为校验模块变更主题 |
| CheckHelper.hideAllTip ( $form ) | 隐藏所有的校验结果，用于重置表单 |
| CheckHelper.parseOtions($tag,$dom) | 标签解析时进行校验配置解析的入口。 |
| CheckHelper.checkDom($dom); | 手动触发校验的API，$dom为拥有校验的input元素。 |

## 校验规则列表 ValidateRules

### 为了更方便的维护列表，因此，将校验规则单独抽出为 common-rule.js

### 在校验规则列表中，如若要固化校验规则函数，则看如下操作：在checkList中添加函数，result为校验结果，regTip为校验提示，$dom为具体的校验DOM元素。

### 

### 在校验规则列表中，如若固化正则表达式，则可如下操作：在regExpList中添加需要的正则表达式。

### 

### 此列表中 还可以维护 拥有检验的元素 好 校验时绑定的事件 如下：

### 

### 当校验的规则很特殊，不需要固化时，可使用API添加

|  |  |
| --- | --- |
| API | 功能 |
| ValidateRules . addCheckRegExp(regExpName, regExp) | 为规则列表增加正则 |
| ValidateRules .addCheckHandler(handlerName, handler) | 为规则列表增加校验函数 |

# 模板模块 TemplateHelper

模板模块，为了更好的维护模板而存在，通过配置项来生成模板并缓存，通过读取缓存的模板和数据来渲染出页面。

## 模板模块的原理

### 模板中会固化模板生成工厂，当需要生成模板时，提供生成模板的配置项与模板工厂名，和生成模板的唯一标识。

### 通过工厂和配置生成的模板会被以传入的标识为ID缓存在模板缓存列表中。

### 当需要模板来渲染数据时，会用ID来从模板缓存中寻找需要的模板。

## API表格

|  |  |
| --- | --- |
| API | 功能 |
| TemplateHelper.render ( id, data [, EngineName]) | 标识为id的EngineName模板通过data数据来生成html，EngineName 为空时，默认为mustache模板 |
| TemplateHelper.parseOptions(type, id, options [, EngineName] ) | 通过options来生成标识为id的EngineName模板。EngineName 为空时，默认为mustache模板 |

## 模板缓存列表 TemplateList

### 模板缓存列表中，存放着固化的模板生成工厂，如下：

### 

### 需要为 mustache模板引擎添加模板时，只需在mustache对象中加入乡音的函数即可，render函数为模板渲染函数。

### 当模板特殊化，或不想在此列表中添加模板时，可是用提供的API来添加，API如下：

|  |  |
| --- | --- |
| API | 功能 |
| TemplateList .addTemFactory（type, fn [, EngineName]） | 为列表添加工厂函数，EngineName 为空时，默认为mustache模板 |