# 表单验证组件调研

JQuery Validation是目前市面上做的比较长久的且比较被大家接受的表单验证组件。易用性扩展性都比较高，这次表单验证组建的调研也就仅仅对JQuery Validation（下面简称JV）的做了深入的研究。

与大部分的表单验证组件一样，JV的表单验证分为规则绑定和执行验证，规则绑定的流程如下：

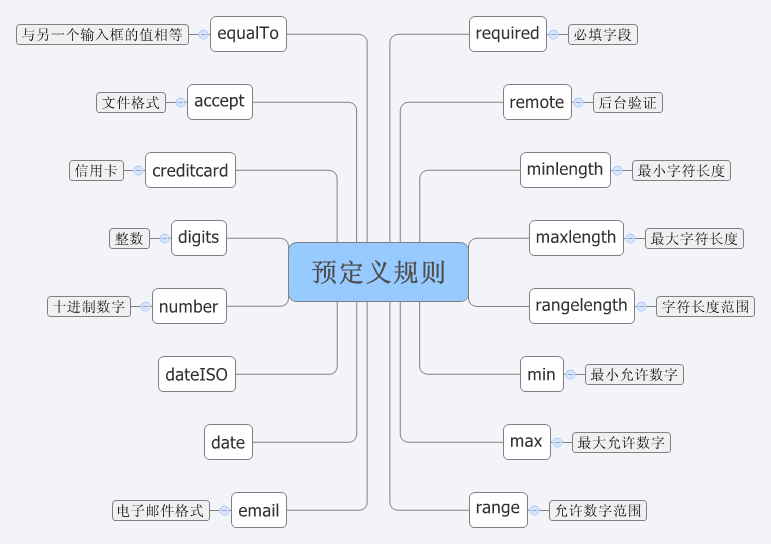
**首先预定义一些验证规则（rules），每个验证规则都有一套验证方法（Method）和提示文本（Message） 》 将这些规则通过一些方式绑定的具体的表单控件 》 将执行验证函数绑定到表单提交事件，三个步骤看上去很简单。**

执行验证的过程：

**获取所有需要验证的表单控件元素 》 获取绑定到每个表单控件的验证规则 》 执行规则中的验证方法，验证失败的话就显示验证文本。**

一如既往得简单。这就是JV表单验证组建的两个主流程了。

**首先来看看JV预定义了哪些验证规则（图）：**



常用的都有了，当然为了符合国情，自定义验证规则是必然的功能，可以用addMethod（name，method，message）绑定性的验证规则到默认配置。这些预定义规则中，有几个需要特殊说明下。

required：

除了可以设置默认必填外，还可以设置依赖必填，当一个css选择符为true或者一个函数返回true时才要求必填，大大提高了灵活度。

remote：

后台验证，成功是要求后台必须返回字符串：“true“，失败时返回失败信息

maxlength：

最大值设置为：-1；2147483647；524288时无效，具体原因没有找到相关文档说明

**JV一共有四种将验证规则绑定到表单组件的形式（图）**

****

通过class来绑定时比较常见的手法，一写不需要传值的规则可以选择这种方式，如：email；url；number等。

当需要给验证方式（method）传值时就需要选择自定义属性的方式了，比如：max；maxlength等。前面提到的requried，如果需要选择符依赖必填，就可以选择使用自定义属性的形式了。这完全基于JQuery强大的css选择器所赐。

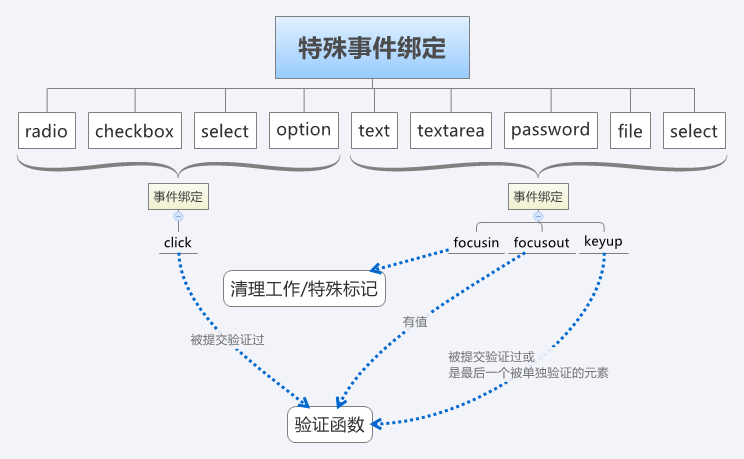
静态规则绑定就是在构造验证类时，使用参数的形式传入。这种方式更灵活，也更强大。如required的函数依赖必填验证就需要通过静态绑定规则到表单控件。

最后一种通过元素据的形式绑定验证规则需要JQuery的Metadata插件支持才行。

**绑定提交事件，直接使用JQuery的submit()方法**

除了提交的时候会进行验证，JV还绑定了一些特殊事件到表单控件，以方便这些事件触发时调用控件的验证函数进行验证。

：text；:password；:file；select； textarea 这5种形式的表单控件绑定了focusout； keyup事件。:radio, :checkbox, select, option这4种形式的表单控件绑定了click事件。当然，他们不是任何时候触发都会调用当前控件的验证方法的。focusout的条件比较宽松一点，主要触发focusout事件时当前控件有值就会调用验证函数。click事件被触发了时需要当前元素已经没提交验证过才会调用验证函数。keyup事件与click事件类似，当前元素被提交过一次后，在这个元素上触发keyup事件时就会调用验证，keyup事件的验证触发还有一个条件是，当前元素是最后一个验证过的元素时，主要用于直接调用element()方法验证过的元素。：text；:password；:file；select； textarea这5种表单控件还绑定了一个事件：onfocusin；触发这个事件的时候并不会调用任何验证方法，只是做了一些额外的清理和标记工作。说了这么多，估计没表达清楚，上图：



**//////////////////////////////////////////////////////////////////////////////////**

**验证流程第一步：获取所有需要验证的元素**

这里有一些规则：提交按钮，重置按钮，被禁用的元素是被排除在外的。指定了的忽略元素也被排除在外。另外没有name的元素和没有绑定验证规则的元素也是被排除在外的。并且如果存在相同name的元素（radio；checkbox），则只获取第一个元素。

**获取字段验证规则**

获取规则跟绑定规则是紧密联系在一起的，将同时在四种绑定形式上获取验证规则。多途径获取同一类型的数据，必然存在重复的问题。所以这里有一个优先级问题。优先级从低到高分别是：metadata规则；class规则；自定义属性规则；静态规则。优先级高的规则将覆盖优先级低的同名规则。前面讲到required规则是支持依赖必填验证的，其实JV的所有规则都在支持依赖验证的。在静态绑定规则时，将规则参数写包含depends属性的对象即可，实际的参数通过该对象的param属性传入即可。此时，依赖选择符或函数赋值给depends属性即可达到依赖验证的目的。甚是强大！！！

rules:{

age:{

max:{

depends:function(){

return $(“.test”).attr(“checked”) ;

},

param:6

},

min:1,

required:true

}

}

//当.test被选中时，才执行age控件的值最大为6的验证

**调用绑定到表单控件的验证方法执行验证**

调用验证方法的过程比较简单，直接调用绑定的验证方法即可。不过现实错误信息的过程还是值得一提的。扩展性比较高。

首先错误信息是支持简单的数组模板替换的，如大小范围错误信息可以这样定义： "请输入大于 {0} 且小于 {1}的数字。"。

可以通过配置参数showErrors自定义错误处理函数。可以通过配置参数wrapper追加一层DOM结构到错误信息外包裹。可以通过配置参数errorClass和validClass验证通过和验证不通过的附加class。可以通过配置参数success，给由错误变成功的提示元素追加class。如果success是一个函数，则直接执行这个函数。

至此，JV的两个流程都走完了。深入研究后发现JV确实很强大，考虑的地方也很多，kissy validation完全可以借鉴JV验证模式来开发。最后附上较全的JV配置参数列表：

