EasyCheck Form Validation **Documentation**

QuickEasy Steps :

1、Import EasyCheck.css，add EasyCheck.js right after jQuery

```
<link rel="stylesheet" type="text/css" href="css/EasyCheck.css"/>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/EasyCheck.min.js"></script>
<script type="text/javascript">
    //Optional Parameters
    EasyCheck.blurChk=true;      // Validate when out of focus stated, false means it's disabled, default value is true
    EasyCheck.keyupChk=true;    // Validate when the key up, false means it's disabled, default value is true
    EasyCheck.loadChk=true;     // After loading html, Checking the validation expression is immediately on or not otherwise
    only validating on the time of submit the form ( if set the value is false , blurChk and keyupChk are disabled ) ,
    default value is true
</script>
```

EasyCheck.css Instruction : In EasyCheck.css,it defines/has four css class style to describe validation failure or success which include error-info, ok-info,error-input and ok-input as below , and also those features can always be rewritten when needed.

You can modify '.easycheck_okInput' to define textfiled default style(ok-input);modify '.easycheck_errorInput' to define textfiled error-input style. Or in page redefine css class style to replace default style.

```
/* CSS Document */
/*div css template for validation faliure*/
.easycheck_errorInfo {
    padding: 2px 8px;
    padding-left: 8px;
    height: 23px;
    font-family: segoe ui, Arial, Helvetica, sans-serif;
    margin-left: 10px;
    font-size: 15px;
    line-height: 23px;
    background-color: #FF6600;
    color: #fff;
    display: inline;
    font-weight: bold;
}
/*div css template for validation success*/
.easycheck_okInfo {
    padding: 3px 8px;
    height: 23px;
    font-family: segoe ui, Arial, Helvetica, sans-serif;
    margin-left: 10px;
    font-size: 15px;
    line-height: 23px;
```

```
    background-color: #7AC935;
    color: #fff;
    display: inline;
    font-weight: bold;
}
/* text_field css template for validation faliure*/
.easycheck_errorInput {
    height: 23px;
    font-size: 14px;
    text-shadow: 0 0 0 rgba(0, 0, 0, 0.3);
    border: 1px solid #DD080A;
    text-indent: inherit;
    font-weight: normal;
    line-height: 23px;
    background: #fff;
    font-family: "segoe ui", sans-serif, Verdana, Arial, Helvetica, Tahoma;
    margin-right: 10px;
    width: 325px;
    margin: 0px 0 0 10px;
    letter-spacing: 1px;
}
/*text_field css template for validation success*/
.easycheck_okInput {
    height: 23px;
    font-size: 14px;
    text-shadow: 0 0 0 rgba(0, 0, 0, 0.3);
    border: 1px solid #cfcfc9;
    text-indent: inherit;
    font-weight: normal;
    line-height: 23px;
    background: #fff;
    font-family: "segoe ui", sans-serif, Verdana, Arial, Helvetica, Tahoma;
    margin-right: 10px;
    width: 325px;
    margin: 0px 0 0 10px;
    letter-spacing: 1px;
}
```

2、Using EasyCheck validator on html

- Class validator :

required <input type="text" name="name" class=" required" />

email <input type="text" name="name" class=" email" />

url <input type="text" name="name" class=" url" />

number <input type="text" name="name" class=" number" />

integer integer <input type="text" name="name" class=" integer" />

if need to use multiple validators at same time, put " " to separate them in between :

required and email <input type="text" name="name" class=" required email" />

- attribute validator :

equalTo: the value must equals to which element's id is ElementId

<input type="password" name="name" equalTo="ElementId"/>

equallength	The length of the value	<input type="password" name="name" equallength ="4"/>
maxlength	The max length of the value	<input type="text" name="name" maxlength="20"/>
minlength	The min length of the value	<input type="text" name="name" minlength="6"/>
max	The max value	<input type="text" name="name" max="20"/>
min	The min value	<input type="text" name="name" min="2"/>
extension	validate input value end with	<input type="text" name="name" extension=".jpg,.gif"/>
reg	customized regular expression	<input type="text" name="name" reg="[A-Z]*"/>
vc	Using Ajax requests the URL specified by vc to check the validate code ,if URL return true value which means success , otherwise failure which is false.	<input type="text" vc="chkvc.jsp" name="vc" />

Server-defined ProcessDemo :

```
<%
        String vc = request.getParameter("vc");  // get input info by verify  text field name
        String res = "false";
        if (vc != null && vc.equals(session.getAttribute("randomNumber"))) {
            res = "true";
        }
        out.print(res);  if output is true which means validation success, otherwise false means failed
%>
```

Explanation: By default, for the reason to avoid unnecessary request from Server,validate vc is only fired by submitting forms, but not on keys up or out of focus those two events. See below :

```
EasyCheck.easyCheckIgnore["vc"]=true; //check Explanation on top.
```

EasyCheckIgnor parameter can set up Ignorance validator for keyUp or onBlur event ,also can easily change its value to false or commented out when needed , which means it will check by the time keyUp or onBlur event is fired.

And also behind the codes, add minlength and maxlength in, it will has the length range :

<input type="password" value="" name="urepwd" size="20" class="txt required" equalto="upwd" maxlength="12" minlength="6"/>

if add min and max in ,there will have the size range :

```
<input   type="password" value=""   name="urepwd" size="20"   class="txt required"   max="20" max="40"/>
```

3、Validate by Submitting Forms

Validate by submitting forms, attach specified id properties and onsubmit="return easyCheckForm(this)" event for forms

```
<form action="login.action" onsubmit="return easyCheckForm(this)" id="regForm">
```

4、To prevent multiple form submit from client side

To avoid multiple form submit, there is a default function can works In EasyCheck.

Sometimes when the server response is slow, it will easily happened click multiple times on submit button from client side, make dupe submission to server, to prevent this situation, EasyCheck has a default function to disable the submit button after click this button to submit form,But in some circumstances ,we still can cancel this function by import EasyCheck.js first then set EasyCheck.easyCheckSubmitDisable value is false.

```
EasyCheck.easyCheckSubmitDisable=false;  //cancel the disable function
```

Special issues for Firefox :

Because when loading data on Firefox browser, its cache has memory ability,if click go back button on browser after submit data,the submit button will still display disabled. So add attribute autocomplete="off" on submit button can fix this problem.

Attribute autocomplete : It is for disable the memory function for default state of browser.Text field on Taobao, Baidu has the same attribute. Although autocomplete attribute is not standard , but this function has been used on IE5 , after, lots of web browsers and html5 also accept.

Besides, if dose not want to change html page using this way to prevent that issue , EasyCheck also can use JS to implement the same way :

```
//method1 : Set id Array for enabled button after click go back on Firefox, accept   multiple array
EasyCheck.removeDisableBtn=['submitId'];
//method2 : Set formId array for enabled button after click go back page on Firefox , accept multiple form id , in those forms all submit buttons will be enabled after click go back page on Firefox
EasyCheck.removeDisableForm=['formId'];
//method3 : Forced setting all submit buttons values to enabled, its default value is false
EasyCheck.removeDisable=true;   // ( this parameter will enabled all the disabled submit
```

buttons on all forms, if there dose not have any disabled submit button by default, this setup is the most efficient way, also set removeDisable is true in EasyCheck.js is the same.)

All those methods can be used together at same time.

5、forbid_validation faliure textfield css

ecss = "no" can forbid validation faliure textfield css——.easycheck_errorInput (EasyCheck CSS) 。

```
<textarea name=" content" class="required" style="width: 400px;height: 100px;border: 1px solid #D4D0C8;" ecss="no"></textarea>
```

other , can use global parameter EasyCheck.ecss to forbid all validation faliure textfield css in the page :

```
EasyCheck.ecss="no";
```

6、Advanced : customized the position of message

(By default easyCheck will display the error message after the text field but if want to display those message on the custom position, see below :

- Find a needed position on page, draw a div for error message , set a id , the id format is error_ElementName (error_element's name)
- [info attribute which is optional , after , this will be the message's prefixion]

then EasyCheck's message will display in this position you create.

)

example :

```
<tr>
        <td align="left"    width="300px">
            <label class="lbl"><div style="color:#FF0000; display:inline">*</div>Login Email</label>
            <div   id="error_uemail"    info=" Login Email "></div>
        /td>
</tr>
<tr>
        <td align="left"><input    type="text"    name="uemail" value="" class="txt required email" size="20"    /> </td>
</tr>
```

7、Advanced : custom message content

After import EasyCheck.js , using expression as below, it can change the message content :

```
EasyCheck.msg['required']=" is required";
```

EasyCheck. Msg List has some example of default msg names and their values, check below:

required	Is required
email	Invalid email
url	Invalid url
number	Invalid number
integer	Invalid integer
equalto	Didn't match input
equallength	length has to be {0}
minlength][maxlength	Please use between {0} and {1} characters
minlength	Use at least {0} characters
maxlength	Must have at most {0} characters
min][max	Value is between{0}and{1}
min	The minimum value of {0}
max	The maximum value of {0}
regexp	Invalid value
extension	Invalid extension only {0}
vc	Didn't match the word verification

8、Advanced : Manually clear the error message

For example, when validate form in the pop up layer, clear the msg before this layer.

To invoke EasyCheck.clearAllError(); to execute.

9、Advanced : custom a new validator--Add a new validator on html

For example, html needs a validator for checking the user is exist or not,see below:

```
//2、Defined the validate function(o,e), check the user name is exist or not
function checkExists(o,e){
	/*

	custom a new plugins function,invokeEasyCheck. addChkMethod(o,e,chkCode,msg)  for
implement new plugins registration to system
	o	Trigger the event element
	e	Event Object
	chkCode  implements callback function, return true or false is the validation result
	msg	validator message
	*/
	 return EasyCheck.addChkMethod(o,e,
		function(o){ //o, Object
		    var val=$(o).val();
		   --WRITE YOUR CODE,return the result  is true or false --
		    var res=false; //result
```

```
                    dwr.engine.setAsync(false); //Disable DWR asynchronous AJAX
                    UserInfoDWR.checkEmail(val,function(d){
                       res=d;
                    });
                    return res;
                },
                "--VALIDATOR MESSAGE-- This name is already exist");
}

//Register Class Validator (validator name, validator function)
EasyCheck.chkList.push(new EasyCheck.ChkRule("exists",checkExists));

//Register Attribute Validator (validator name, validator function)
EasyCheck.chkList.push(new EasyCheck.ChkRule("exists",checkExists));
```

10、Advanced : Extended EasyCheck validation framework, new validator

Two easy steps for changing EasyCheck's validation framework , add new validator !

1、custom new validation functions

```
/*
 custom new validation functions , to invoke EasyCheck. addChkMethod(o,e,chkCode,msg) for
 implement new validator plugins register to system
    o  Trigger the event element
   e        Event Object
   chkCode  implements callback function, return true or false is the validation result
   msg      alert message
  Attention to blue part:
*/
function checkNew(o,e){
    return EasyCheck. addChkMethod(o,e,
          function(o){
              --WRITE YOUR CODE,return the result    is true or false --
             //verified the implementation,return true or false : true(validation success) ;
                              false(validation failure) , display the error message
             /
         },
         " the error msg when return false");
}
```

2、Register a custom validation function

```
/*
    register validator Object(validatorName,validatorFunction[,isAttributeValidator])
    when is a attribute validator ,set isAttributeValidator=true
    Get EasyCheck.chkList array from EasyCheck ，in this array, depending on
     the validation function's class, add new validation object
```

3、if this validator only needs to be verified when submit form （such as validate code dose not need to be verified when onBlur or keyUp event fired），then see below to register ,blue part is validators' names on the event onblur and keyup fired）

- EasyCheck.easyCheckIgnore : validatorName specified validator stop working， Only check when submit forms。
  EasyCheck.easyCheckIgnore["validatorName"]=true;
- EasyCheck.easyCheckBlurIgnore : validatorName specified validator stop working when onblur event fired。
  EasyCheck.easyCheckBlurIgnore["validatorName "]=true;
- EasyCheck.easyCheckKeyupIgnore : validatorName specified validator stop working when onkeyup event fired。
  EasyCheck.easyCheckKeyupIgnore["validatorName "]=true;
- EasyCheck.easyCheckEleIgnore : elementName specified Element stop validator when onkeyup and onblur event fired。
  EasyCheck.easyCheckEleIgnore["elementName"]=true;

11、Advanced : Maintenance and management

For the globalized reason to easy manage and maintain the error message, we can defined them to EasyCheck. Msg list as below:

```
msg:{
        "required":"Is required",
        "email":"Invalid email",
        "url":" Invalid url",
        "number":" Invalid number",
        "integer":" Invalid integer",
        "equalto":"Didn't match input",
        "equallength":"Length    has to be {0}",
```

```
        "minlength][maxlength":"Please use between {0} and {1} characters",
        "minlength":"Use at least {0} characters",
        "maxlength":"Must have at most {0} characters",
        "min][max":"Value is between{0} and {1}",
        "min":"The minimum value of {0}",
        "max":"The maximum value of {0}",
        "regexp":"Invalid value",
         "extension":"Invalid extension,only {0}",
         "vc":"Didn't match the word verification" ,
        "messageName":"message content，you can use {0}，{1}……parameter"
    }
```

- Register messages ，using EasyCheck.msg[“messageName”]to get message

As :

EasyCheck.msg["required"]

```
function checkNew(o,e){
    return EasyCheck. addChkMethod(o,e,
          function(o){
              //return true or false : true:success ; false:failure and also alert display the msg
             //return $.trim(val)!="";
        },EasyCheck.msg["required"]);
```

- if msg has parameter (for example "length is between {0}and {1}") , then invoke EasyCheck.formatMsg(“Message content”，“paramter1” ，……) to format msg evaluation check as follow：

```
EasyCheck.formatMsg(EasyCheck.msg["minlength][maxlength"],$(o).attr("minlength"),$(o).attr("maxlength
"))
```