**bootstrapvalidator之API学习**

2016年05月31日 03:33:45 [哪知道](https://me.csdn.net/nazhidao) 阅读数：47349 标签： [bootstrap](http://so.csdn.net/so/search/s.do?q=bootstrap&t=blog)[bootstrapvalidator](http://so.csdn.net/so/search/s.do?q=bootstrapvalidator&t=blog) 更多

个人分类： [前端开发](https://blog.csdn.net/nazhidao/article/category/6271414)

最近项目用到了bootstrap框架，其中前端用的校验，采用的是bootstrapvalidator插件，也是非常强大的一款插件。我这里用的是0.5.2版本。  
下面记录一下使用中学习到的相关API，不定期更新。  
  
  
1. 获取validator对象或实例  
一般使用校验是直接调用$(form).bootstrapValidator(options)来初始化validator。初始化后有两种方式获取validator对象或实例，可以用来调用其对象的方法，比如调用resetForm来重置表单。有两种方式获取：  
1）

1. *// Get plugin instance*
2. var bootstrapValidator = $(form).data('bootstrapValidator');
3. *// and then call method*
4. bootstrapValidator.methodName(parameters)

这种方式获取的是BootstrapValidator的实例，可以直接调用其方法。  
2）

$(form).bootstrapValidator(methodName, parameters);

这种方式获取的是代表当前form的jquery对象，调用方式是将方法名和参数分别传入到bootstrapValidator方法中，可以链式调用。  
两种方式的使用分别如下：

1. *// The first way*
2. $(form)
3. .data('bootstrapValidator')
4. .updateStatus('birthday', 'NOT\_VALIDATED')
5. .validateField('birthday');
7. *// The second one*
8. $(form)
9. .bootstrapValidator('updateStatus', 'birthday', 'NOT\_VALIDATED')
10. .bootstrapValidator('validateField', 'birthday');

2. defaultSubmit()  
使用默认的提交方式提交表单，调用此方法BootstrapValidator将不执行任何的校验。一般需要时可以放在validator校验的submitHandler属性里调用。  
使用：

1. $('#defaultForm').bootstrapValidator({
2. fields: {
3. username: {
4. message: 'The username is not valid',
5. validators: {
6. notEmpty: {
7. message: 'The username is required and can\'t be empty'
8. }
9. }
10. }
11. }，
12. submitHandler: function(validator, form, submitButton) {
13. *// a)*
14. *// Use Ajax to submit form data*
15. *//$.post(form.attr('action'), form.serialize(), function(result) {*
16. *// ... process the result ...*
17. *//}, 'json');*
19. *//b)*
20. *// Do your task*
21. *// ...*
22. *// Submit the form*
23. validator.defaultSubmit();
24. }
25. });

3. disableSubmitButtons(boolean)  
启用或禁用提交按钮。BootstrapValidator里默认的提交按钮是所有表单内的type属性值为submit的按钮，即[type="submit"]。  
使用：  
当登录失败时，重新启用提交按钮。

1. $('#loginForm').bootstrapValidator({
2. message: 'This value is not valid',
3. feedbackIcons: {
4. valid: 'glyphicon glyphicon-ok',
5. invalid: 'glyphicon glyphicon-remove',
6. validating: 'glyphicon glyphicon-refresh'
7. },
8. submitHandler: function(validator, form, submitButton) {
9. $.post(form.attr('action'), form.serialize(), function(result) {
10. *// The result is a JSON formatted by your back-end*
11. *// I assume the format is as following:*
12. *// {*
13. *// valid: true, // false if the account is not found*
14. *// username: 'Username', // null if the account is not found*
15. *// }*
16. if (result.valid == true || result.valid == 'true') {
17. *// You can reload the current location*
18. window.location.reload();
20. *// Or use Javascript to update your page, such as showing the account name*
21. *// $('#welcome').html('Hello ' + result.username);*
22. } else {
23. *// The account is not found*
24. *// Show the errors*
25. $('#errors').html('The account is not found').removeClass('hide');
27. *// Enable the submit buttons*
28. $('#loginForm').bootstrapValidator('disableSubmitButtons', false);
29. }
30. }, 'json');
31. },
32. fields: {
33. username: {
34. validators: {
35. notEmpty: {
36. message: 'The username is required'
37. }
38. }
39. },
40. password: {
41. validators: {
42. notEmpty: {
43. message: 'The password is required'
44. }
45. }
46. }
47. }
48. });

4. enableFieldValidators(field, enabled)  
启用或禁用指定字段的所有校验。这里我的实

验结果是如果禁用了校验，好像对应的字段输入（文本框、下拉等）也会变为禁用。  
使用：  
当密码框不为空时，开启密码框和确认密码框的校验：

1. *// Enable the password/confirm password validators if the password is not empty*
2. $('#signupForm').find('[name="password"]').on('keyup', function() {
3. var isEmpty = $(this).val() == '';
4. $('#signupForm').bootstrapValidator('enableFieldValidators', 'password', !isEmpty)
5. .bootstrapValidator('enableFieldValidators', 'confirm\_password', !isEmpty);
6. if ($(this).val().length == 1) {
7. $('#signupForm').bootstrapValidator('validateField', 'password')
8. .bootstrapValidator('validateField', 'confirm\_password');
9. }
10. });

5. getFieldElements(field)根据指定的name获取指定的元素，返回值是null或一个jquery对象数组。

 6. isValid()返回当前需要验证的所有字段是否都合法。调用此方法前需先调用validate来进行验证，validate方法可用在需要点击按钮或者链接而非提交对表单进行校验的时候。使用：点击某按钮时，提示所有字段是否通过校验。

1. $("#isAllValid").on("click", function(){
2. alert($("#defaultForm").data('bootstrapValidator').isValid());
3. });

7. resetForm(Boolean)  
重置表单中设置过校验的内容，将隐藏所有错误提示和图标。  
使用：

1. $("#isAllValid").on("click", function(){
2. alert($("#defaultForm").data('bootstrapValidator').isValid());
3. if(!$("#defaultForm").data('bootstrapValidator').isValid()) {
4. $("#defaultForm").data('bootstrapValidator').resetForm();
5. }
6. });

8. updateElementStatus($field, status, validatorName)  
更新元素状态。status的值有：NOT\_VALIDATED, VALIDATING, INVALID or VALID。  
  
  
9. updateStatus(field, status, validatorName)  
更新指定的字段状态。BootstrapValidator默认在校验某个字段合法后不再重新校验，当调用其他插件或者方法可能会改变字段值时，需要重新对该字段进行校验。  
使用：

点击按钮对文本框进行赋值，并对其重新校验：

1. $('#defaultForm').bootstrapValidator({
2. fields: {
3. username: {
4. message: 'The username is not valid',
5. validators: {
6. notEmpty: {
7. message: 'The username is required and can\'t be empty'
8. }
9. }
10. }，
11. stringLength: {
12. min: 6,
13. max: 30,
14. message: 'The username must be more than 6 and less than 30 characters long'
15. }
16. }
17. });
19. $("#setname").on("click", function(){
20. $("input[name=username]").val('san');
21. var bootstrapValidator = $("#defaultForm").data('bootstrapValidator');
22. bootstrapValidator.updateStatus('username', 'NOT\_VALIDATED').validateField('username'); *//错误提示信息变了*
23. });

10. validate()  
手动对表单进行校验，validate方法可用在需要点击按钮或者链接而非提交对表单进行校验的时候。  
由第一条可知，调用方式同样有两种：

1. $(form).bootstrapValidator(options).bootstrapValidator('validate');
3. *// or*
4. $(form).bootstrapValidator(options);
5. $(form).data('bootstrapValidator').validate();

11. validateField(field)  
对指定的字段进行校验。