**一、使用jquery进行表单验证**

使用了html辅助方法的页面，同样可以使用jQuery进行表单验证

引入jquery-添加验证方法-调用验证方法

实例

<script>

$(function(){

//当表单提交时，绑定checkForm方法

$("form").bind("submit",checkForm);

});

function checkForm(){

if(checkUser()&&checkPwd())

return true;

return false;

}

function checkUser(){

…

return false;

}

function checkPwd(){

…

return false;

}

//页面中调用方法new {onblur="checkUser()"}

<script>

@using (Html.BeginForm("index","home",FromMethod.Post)){

<span>用户名</span>@Html.TextBox("uname","",new {onblur="checkUser()"})

}

//注意，给辅助方法写属性需要使用new

实例

Html.BeginForm("index","home",FromMethod.Post,new{id="",name="",onblur="",@class=""})

class前需要使用@

**二、使用JQUERY Validate插件验证**

先依次导入

jquery-3.4.1.min.js

jquery.validate.js

query.metadata.js

jquery.validate.messages\_cn.js 用以支持中文信息

<script src="~/Scripts/jquery-1.8.3.min.js"></script>

<script src="~/Scripts/jquery.validate.js"></script>

<script src="~/Scripts/jquery.metadata.js"></script>

<script src="~/Scripts/jquery.validate.messages\_cn.js"></script>

实际操作，页面中写法

@section scripts{

@Scripts.Render("~/bundles/jqueryval")

<script>

$("#formstudent").validate({ //注意不是ta,是te结尾

rules: {

login\_name: { required: true },

login\_pwd: { required: true,rangelength:[3,12] }

},

messages:{

login\_name: { required: "账号不能为空"},

login\_pwd: { required: "密码不能为空",rangelength:"长度3-12位" }

}

})

</script>

}

此种验证会在页面元素后自动显示错误信息，不需要显示的调用。

**三、模型注解验证**

模型中添加注解

在模型字段前，通过注释，添加验证

[Required(ErrorMessage="错误信息")]

[StringLength(50,ErrorMessage="错误信息")]

[StringLength(12,MinimumLength =3, ErrorMessage = "密码长度3-12位")]

[Range(1,50,ErrorMessage="错误信息")]

[Range(typeof(decimal),1,50,ErrorMessage="错误信息")]

[Compare("表单字段",ErrorMessage="错误信息")]

[RegularExpression("pattern",ErrorMessage="错误信息")] 匹配正则表达式

**设置别名[Display(Name="")]**

//模型注解可以使用服务器端或者脚本验证

1、服务器端控制器视图动作中

if(ModelState.IsValid){

ModelState.AddModelError("","用户名密码不匹配")；

}

视图中

@Html.ValidationSummary()方法集中显示错误提示信息

@Html.ValidationMessage("id") 显示相应字段的错误提示信息