[**CKEditor的安装与基本使用（JSP）**](http://blog.mcncc.com/7875.html)

**一、下载CKEditor**

1. 直接下载地址，当前最新版本为3.5：

<http://download.cksource.com/CKEditor/CKEditor/CKEditor%203.5/ckeditor_3.5.zip>

2. 或者可直接到官方网站下载：

<http://ckeditor.com/download>

**二、安装CKEditor**

解压ckeditor\_3.5.zip，得到ckeditor文件夹，将ckeditor整个文件夹复制到项目工程的根目录下

**三、验证CKEditor是否安装成功**

部署运行项目，访问: http://localhost/项目名称/ckeditor/\_samples/index.html

则可出现 **“CKEditor Samples List”**的Demo页面，说明CKeditor已安装成功

**四、应用CKEditor**

1. 导入JS文件

<script type=”text/javascript” src=”http://chrissie.javaeye.com/blog/<%=request.getContextPath()%>/ckeditor/ckeditor.js“></script>

2. 创建和使用CKEditor

<textarea class=”ckeditor” cols=”80″ id=”content” name=”content” rows=”10″> CKEditor Test……(此处的内容会在编辑器中显示)</textarea><script type=”text/javascript”>//<![CDATA[ CKEDITOR.replace('content',{toolbar:'Full', skin : 'kama'});//]]></script>

说明：

1) textarea属性值 name=”content“: 名字可随意定义，但必须与下面的**CKEDITOR.replace(‘content‘);**匹配

2) **CKEDITOR.replace(‘content‘);**是最基本的写法，表示使用CKEditor的JavaScript API创建的编辑器实例，

替换上面的**textarea**。

若要为CKEditor增加一些属性设置，则其属性必须写在{}花括号内，

如上{toolbar:‘Full’, skin : ‘kama’}，

具体的属性网上很多文章均可搜索到，这里不累述了。

3) 上面的textarea后面的“**CKEDITOR.replace(‘content‘);”**脚本也可以写在<head></head>标签内

<script type=”text/javascript”> window.onload = function(){ CKEDITOR.replace(‘content’); }</script>

**五、获取编辑器里的数据**

有时在提交表单时，需要判断下编辑器里的内容是否为空，这时可使用CKEditor的JavaScript API：

**CKEDITOR.instances.content.getData()**

function test() { var editor\_data = http://chrissie.javaeye.com/blog/CKEDITOR.instances.content.getData(); if(editor\_data==null || editor\_data==”"){ alert(“请填写内容！”); return false; }}

注意：如果你在编辑器里打了一堆的空格，内容是不为空的，因为编辑器会把你打的空格都转换成&nbsp;

### CKEditor添加自定义按钮实例：

自定义工具栏按钮:

我们可以自定义ckeditor工具栏要显示的按钮，工具栏按钮定义可以参考[这里](http://docs.cksource.com/CKEditor_3.x/Developers_Guide/Toolbar)。

现在我们需要向工具栏添加一个自定义功能的按钮。ckeditor工具栏中的每个按钮都是作为插件定义在**ckeditor\plugins\** 目录中。我们在**ckeditor\plugins\**中创建一个新文件夹linkbutton。在linkbutton文件夹内，我们创建一个plugin.js文件，它的代码如下：

CKEDITOR.plugins.add('linkbutton', {

init: function(a){

var b = a.addCommand('linkbutton', {

//Section 1 : 按下自定义按钮时执行的代码

exec:function(editor){

alert("这是自定义按钮");

}

});

//Section 2 : 创建自定义按钮、绑定方法

a.ui.addButton('linkbutton', {

label: '弹出页面',

command: 'linkbutton',

icon: this.path + 'logo\_ckeditor.png'

});

}

});

在上面的代码中我们指定自定义按钮的图标logo\_ckeditor.png，logo\_ckeditor.png放在linkbutton文件夹中。

接下来我们需要注册linkbutton插件。<http://ajithmanmadhan.wordpress.com/2009/12/16/customizing-ckeditor-and-adding-a-new-toolbar-button/>原文中的给出方法是在ckeditor.js中注册，这会使以后升级新版本遇到困难。提倡使用下面的方法在config.js中注册自定义插件：

CKEDITOR.editorConfig = function( config )  
{  
    // Define changes to default configuration here. For example:  
    // config.language = 'fr';  
    // config.skin = 'office2003';  
    config.extraPlugins="linkbutton"  
};

最后，在ckeditor中显示自定义按钮linkbutton，代码如下：

<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />  
<title>自定义按钮</title>  
<script type="text/javascript" src="ckeditor.js"></script>  
<style type="text/css">   
body{font-size:12px;line-height:1.8;font-family:'微软雅黑';}  
</style>  
</head>  
   
<body>  
<textarea id="ckeditor"></textarea>  
<script type="text/javascript">  
    CKEDITOR.replace( 'ckeditor',{  
        toolbar :  
            [  
                [ 'Bold', 'Italic', '-', 'NumberedList', 'BulletedList', '-','Link', 'Unlink','-','linkbutton' ]  
            ],  
        extraPlugins:"linkbutton"    //注册linkbutton,也可在config.js中注册  
    });  
</script>  
</body>  
</html>

注意：自定义插件的名称必须在任何地方都要保持一致。

用户可以在config.js里自定义工具栏：

  config.language = 'zh-cn';// 界面语言，默认为 'en'  
    config.width = 400; // 设置宽高  
    config.height = 400;// 设置高度  
    config.skin = 'v2';// 编辑器样式，有三种：'kama'（默认）、'office2003'、'v2'  
    config.uiColor = '#FFF'; // 背景颜色

CKEDITOR.editorConfig = function( config )

{

config.toolbar = 'MyToolbar';//把默认工具栏改为‘MyToolbar’

config.toolbar\_MyToolbar =

[

['NewPage','Preview'],

['Cut','Copy','Paste','PasteText','PasteFromWord','-','Scayt'],

['Undo','Redo','-','Find','Replace','-','SelectAll','RemoveFormat'],

['Image','Flash','Table','HorizontalRule','Smiley','SpecialChar','PageBreak'],

'/',

['Styles','Format'],

['Bold','Italic','Strike'],

['NumberedList','BulletedList','-','Outdent','Indent','Blockquote'],

['Link','Unlink','Anchor'],

['Maximize','-','About']

];

};

或者你也可以定义多个Toolbar，然后在不同的地方使用不同的toolbar属性的设置：

CKEDITOR.replace( 'editor1',

{

toolbar : 'MyToolbar'

});

CKEDITOR.replace( 'editor2',

{

toolbar : 'Basic',//自定义工具栏按钮

uiColor: '#14B8C4' //自定义工具栏背景色

});

当然，你也可以在调用CKEditor的地方直接定义新的工具栏

CKEDITOR.replace( 'editor1',

{

toolbar :

[

['Styles', 'Format'],

['Bold', 'Italic', '-', 'NumberedList', 'BulletedList', '-', 'Link', '-', 'About']

]

});