# 实验十：应用AJAX

## 实验目的

掌握AJAX相关知识及其实用方法。

## 实验内容

1. 掌握Ajax.ActionLink的使用方法
2. 掌握Ajax.BeginForm的使用方法
3. 掌握$.ajax的使用方法
4. 掌握jQuery.form.js的使用方法
5. 掌握JQuery Validate的简单使用
6. JSon数据的展示

## 实验步骤

1. **Ajax.ActionLink。**
2. 根据教材P160实现步骤完成。
3. **Ajax.BeginForm。**
4. 根据教材P163实验步骤完成。
5. **完成$.ajax方法的封装**
6. 采用$.ajax 完成P160页的例子，并为“About”action传递一个参数过去。
7. **jQuery.form.js的使用**
8. 改造P163例子
9. 在“ajaxController”中增加“leaveMessageAjax”Action，代码如下：

public JsonResult leaveMessageAjax(string userName, string message)

{

var file = Request.Files;

string result = string.Format("<div>用户：{0},留言内容:{1}</div>",

userName, message);

var json = new { message = result };

return Json(json);

}

1. 在“ajaxController”中定义一个indexJForm的action，在视图中添加如下的代码

@section Scripts{

<script type="text/javascript" src="/Scripts/jquery.form.js"></script>

<script type="text/javascript" src="/Scripts/lib.js"></script>

<script type="text/javascript">

//文档加载完毕后执行的代码

$(document).ready(function () {

//ajax提交form

$(".submitAjax").click(function () {

ajaxForm($("#leaveMessage"), function (e) {

alert(e.message);

$("#showMessage").append(e.message);

})

return false;

});

});

</script>

}

<div>============================</div>

<div>ajax.beginForm的使用</div>

<form id="leaveMessage" action="/ajax/leaveMessageAjax">

<div>用户名：<input type="text" name="userName" /></div>

<div>留言：<textarea name="message"></textarea> </div>

<div><input type="file" name="headerImg" /></div>

<div>

<input type="submit" class="submitAjax" value="提交" />

</div>

</form>

<div id="showMessage">

</div>

1. **ajaxStart,ajaxStop实现ajax请求的遮罩层。**
2. 在母版页中加入如下的标记代码

<div id="loading">

<img src="/Images/loading.gif" />

</div>

1. 在母版页head中加入如下的样式和js代码

<style type="text/css">

/\* 半透明的遮罩层 ajax请求用 \*/

#loading {

background: #000;

filter: alpha(opacity=30); /\* IE的透明度 \*/

opacity: 0.3; /\* 透明度 \*/

display: none;

position: fixed;

top: 0px; left: 0px;

width: 100%;

z-index: 10000; /\* 此处的图层要大于页面 \*/

display: none;

}

#loading img {

width: 124px;

height: 124px;

position: absolute;

top: 50%;

left: 50%;

margin-left: -62px;

margin-top: -62px;

}

</style>

$(document).ready(function () {

//ajax请求开始显示loading

$(document).ajaxStart(function (option) {

$("#loading").css("height", $(document).height());

$("#loading").show();

});

//ajax请求结束隐藏loading

$(document).ajaxStop(function (option) {

$("#loading").hide();

});

});

1. 把图片拷贝到images目录中
2. **完成图书批量删除功能**
3. 在简易书店的bookList视图中增加批量删除功能，在新增后追加以下代码：

<a class="batchDelete">批量删除</a>

1. 为图书展示每一行都增加复选框，注意表头和表体都要处理（此处例子给出的表头只增加了一个空单元格占位，可参考表体的方式自己增加全选操作）

表头： <th>

</th>

表体：<td>

<input type="checkbox" data-value="@item.BookId" class="rowCbx" />

</td>

1. 为批量删除增加点击响应，参考代码如下

<script type="text/javascript">

$(document).ready(function () {

$(".batchDelete").click(function () {

var selected = "";

$(".rowCbx").each(function () {

if ($(this).is(":checked"))

selected += $(this).attr("data-value") + ",";

})

$.ajax({

url: "/book/batchDelete",

type: "post",

data: { selected: selected },

dataType: "json",

error: function (e) {

alert("alert!ajax请求出错！");

},

success: function (e) {

alert(e.result);

}

});

return false;

})

})

</script>

1. 在控制器中增加响应批量删除的action，测试查看效果，检查勾选的数据是否成功传入action。

[HttpPost]

public ActionResult batchDelete(string selected)

{

var result = new { result = "testOK" };

return Json(result);

}

1. 增加业务类代码（db/bll/books.cs）

//批量删除

public static void batchDelete(List<string> bookList)

{

dbEntities dc = new dbEntities();

for (int iIndex = 0; iIndex < bookList.Count; ++iIndex)

{

if(bookList[iIndex] == "")

{

break;

}

int bookId = Convert.ToInt32(bookList[iIndex]);

Books entry = dc.Books.SingleOrDefault(a => a.BookId == bookId);

dc.Books.Remove(entry);

}

dc.SaveChanges();

}

1. 在action中完成调用并查看效果。尝试自己完成局部刷新的效果。
2. **JQuery.Validate.js的使用**
3. 把jquery.validate.unobtrusive.js和lib.js复制到Scripts目录中，把layer拷贝到contents目录中。
4. 在ajaxController中增加“indexValidate”Action，并在试图中添加如下代码，运行查看验证效果。

@section Scripts{

<script type="text/javascript" src="/Scripts/jquery.form.js"></script>

<script type="text/javascript" src="/Content/layer/layer.js"></script>

<script type="text/javascript" src="/Content/layer/layui.js"></script>

<script type="text/javascript" src="/Scripts/lib.js"></script>

}

<form id="leaveMessage" action="/ajax/leaveMessage">

<div>用户名：<input type="text" name="userName" class="required" /></div>

<div>留言：<textarea name="message"></textarea> </div>

<div>

<input type="submit" class="submitAjax" value="提交" />

</div>

</form>

1. **请求JSON并展示数据**
2. 在“ajaxController”中增加“jsonTest”Action，在试图中通过ajax请求第四章返回的json结果，并将Json对象内包含的信息输出到console界面中。
3. “jsonTest”Action的代码如下：

public ActionResult jsonTest()

{

return View();

}

1. jsonTest的视图代码如下：

@section Scripts{

<script type="text/javascript">

$(document).ready(function () {

$(".getJson").click(function () {

$.ajax({

url: "/result/json",

method: "post",

dataType: "html",

success: function (e) {

var jsonObj = $.parseJSON(e);

console.info(jsonObj.name);

console.info(jsonObj.parent.length);

for (i = 0; i < jsonObj.parent.length; i++)

{

console.info(jsonObj.parent[i].name);

}

console.info(jsonObj.parent1.name);

console.info(jsonObj);

}

});

return false;

});

});

</script>

}

<h2>JSon测试</h2>

<a class="getJson">获取json数据</a>