

gineGrid-通用ajax类 (2.0版本) 使用说明.

程序创建于2007年2月，目的是给各位使用AJAX的朋友提供一个方便。同时相互交流经验。您可以任意散布、传播和复制程序代码，并且可以直接免费使用在任意用途。但是出于对作者的尊重，请保留代码中的注释信息，无论是客户端还是服务端，注释信息都不会影响程序的执行效率。并且在您觉得有更好的方案可以改进代码质量的时候，请发一份给我，由我来进行修改，然后统一发布更新。

2007-10-28 梁逸晨

我的信箱 kvspas@gmail.com QQ:2090913 MSN: ddsksworks@hotmail.com

我的网站地址：<http://www.kvspas.com> (所有作品都会在上面作最新更新)

博客: <http://blog.kvspas.com> (网站内容的同步镜像复制)

本次版本变动主要更新的部分：

在第一版的基础上，扩展了代理的功能

服务端代理上：

- 1 增加了POST代理。改善了读取效率。
- 2 改善了编码判断的稳定性，早期版本一定需要远程目标页面是HTML文档，现在只要是文本文件，都可以判断其编码类型。使乱码的出现机率更小。

客户端代码上：

应很多朋友的反映，在函数的命名上很难理解，所以我增加了六个新的接口，

cacheText和cacheXML用于访问缓存文本和缓存XML,这两个功能可以为站点节省很多带宽

ajaxText对应早期的eAT

ajaxXML对应早期的eAX

loadText对应早期的ajaxT

loadXML对应早期的ajaxX

第一版本中的ASP代理文件停止升级支持，以后只统一使用ASP.NET。也不会有开发PHP或者JSP版本的计划。

早期的四个函数仍然可以使用，但是不推荐使用，因为大项目中需要可读性清晰的接口

gineGrid-通用ajax类 (2.0版本)的下载地址：http://www.cnblogs.com/Files/kvspas/ginegrid_stl2.7z

您获得的压缩文件包总共包含有7个文件:

以下四个文件是在正式使用时不需要的。

1 test.html (测试用的主文件)

2 post.asp (测试POST数据用的服务端文件)

3 tt.txt (测试用的文本文件)

4 gineGrid_2_0.pdf PDF版本的说明文件。

下面2个文件是在正式使用时必需的。

5 ajax.aspx (跨域访问数据用的服务端代理文件)

6 js文件夹中的ginegrid_stl.js (核心类文件)

7 bin文件夹：把里面的dll文件拷贝到你网站根目录的bin目录下面。

下面是使用说明：

关于运行测试文件：

首先请确保您将所需的文件都保存在WEB服务器上，并且可以通过URL地址访问到它们。您必须通过标准的URL地址访问它们，才能够使程序得到正确运行，

例如 <http://localhost/test.html> <http://127.0.0.1/test.html> <http://www.abc.com/test.html>

您不能使用本机资源路径的方式来打开它们，否则程序不会正确执行。

例如 <d:\test.html> <d:\test\test.html> 这两种方法都是错误的。

如何在您的项目中使用它们：

首先,加载代码: ginegrid_stl.js. (一般情况下您不需要关心它是怎么执行的。您只要加载它即可)

代码默认创建了net对象，直接只用即可，你也可以创建自己的对象：

```
var s = new gineGrid();
```

然后即可使用六种基本功能：同步获取文本（loadText）、同步获取XML对象（loadXML）、异步获取文本（ajaxText）、异步获取XML对象（ajaxXML）、读取缓存本地文本（cacheText）、读取缓存XML（cacheXML）。

使用办法非常简单：

同步函数（url地址，可选的POST参数）

异步对象（回调函数，url地址，可选的POST参数）

同步获取文本：s.loadText("文本数据的地址",[可选的post参数])

例：var txt = s.loadText("tt.asp");//GET alert(txt);

var pxt = s.loadText("tt.asp","p1=12&p2=345");//POST
alert(pxt);

同步获取XML对象：s.loadXML("XML数据的地址",[可选的post参数])

例：var dom = s.loadXML("xml.asp");//GET

//dom即是一个XML Document 对象。
//获取对象后的操作，省略。

var pdom = s.loadXML("xml.asp","p1=12&p2=345");//POST

//dom即是一个XML Document 对象。
//获取对象后的操作，省略。

异步模式关系到一个结果变量：ajaxGLOBAL,所有获取的结果都保存在这个变量中。

异步获取文本：s.ajaxText(回调函数,"文本数据的地址",[可选的post参数]) 例：

GET方式：

```
ajaxText(  
  
function(){alert-ajaxGLOBAL);} //回调函数  
  
,"  
  
"tt.asp" //文本数据地址。  
  
);
```

POST方式： ajaxText(

```
function(){  
  
    alert-ajaxGLOBAL);} //回调函数  
  
,"  
  
"tt.asp" //文本数据地址。  
  
,"  
  
"p1=12&p2=345"  
  
);
```

异步获取XML对象：s.ajaxXML(回调函数,"XML数据的地址"[,可选的post参数]) 例：

GET方式：

```
ajaxXML(  
function(){  
    alert-ajaxGLOBAL;  
} ,  
xml.asp" //XML数据地址  
);
```

POST方式：

```
ajaxXML(  
function(){alert-ajaxGLOBAL);}
```

```
,
xml.asp" //XML数据地址。
,
p1=12&p2=345"
);
```

访问缓存文件：

cacheText和cacheXML用于访问缓存文本和缓存XML,这两个功能可以为站点节省很多带宽

例如你有一个长期不会更新的文件，你不希望客户端仍然每次都重新请求这个文件，你就可以这么做：

```
net.cache( //文本

function(){alert ajaxGLOBAL;},

"/test.css"

);
```

或者：

```
net.cache( //XML

function(){alert ajaxGLOBAL;},

"/test.xml"

);
```

访问跨域数据的时候，通过ajax.aspx来代理，encode参数标明目标页面的编码类型，url参数是目标文件的地址，例如：

```
net.ajaxText(
function(alert ajaxGLOBAL),
"ajax.aspx?encode=65001&url=http://www.abcd.com"
);
```

最后简单回顾：六个函数的用法其实都是一样的，您可以这么简要记忆：

- 1 同步函数 (url地址 , 可选的POST参数)
- 2 异步函数 (回调函数 , url地址 , 可选的POST参数)
- 3 缓存函数 (url地址)

参考文档:

[关于Ajax](#) 来自IBM的详细解释.

[基于Ajax的应用程序架构汇总](#) 来自蓝色理想

如果您有更好的建议和方案 , 欢迎与我交流 :

2007 - 10 -28 kvspas@gmail.com QQ : 2090913 MSN : ddsksworks@hotmail.com 梁逸晨