

# 基于Ajax的动态JavaScript文件加载模型

熊海东<sup>1</sup>, 陈亚军<sup>2</sup>, 潘刚<sup>3</sup>

(1.西华师范大学 计算机学院, 四川 南充 637002; 2.西华师范大学 实验中心, 四川 南充 637002;  
3.四川文理学院 计算机学院, 四川 达州 635000)

**【摘要】**在基于B/S模式的系统中,浏览器端程序是以动态脚本语言JavaScript为主。浏览器的标签在不刷新的情况下,在初始化时加载用户可能用到的所有JavaScript文件。而对于浏览器端不同权限的用户、同一用户在不同时刻所需要用到的JavaScript文件是不一样的。本文提出一种动态加载javascript文件的模型,只传递浏览器端必须用到的JavaScript文件,杜绝了一些安全隐患,减少网络流量,保护了服务端的JavaScript代码。

**【关键词】**B/S模式;异步Ajax请求;JavaScript;eval;动态加载

**【中图分类号】**TP393.092 **【文献标识码】**A **【文章编号】**1673-1891(2011)03-0052-03

## 引言

伴随着互联网的迅猛发展,计算机应用程序也发生了巨大变化。传统C/S模式已经显得非常局限了,B/S模式以其跨地域、跨平台、跨浏览器、变更容易等特点成为主流。而B/S模式中的B端也就是浏览器端是与用户直接交互的部分。浏览器端程序的事实标准是JavaScript文件,是由服务端发送过来的。当用户打开浏览器,通过Web向S端也就是服务端发起应用请求,服务端响应浏览器的请求向浏览器端传递JavaScript文件,浏览器端刷新浏览器标签并解释运行JavaScript文件。通常简单的Web应用就只刷新一次浏览器标签,只需一次加载所有可能用到的JavaScript文件。而对于复杂的Web应用如果一次加载所有的JavaScript文件的话,会造成浏览器端加载JavaScript文件时间过长,而且用户可能只是用到其中一部分,这就造成很大的浪费,一方面浪费用户的时间,另一方面浪费网络和服务端资源。所以浏览器端运行时按需要动态加载JavaScript文件很有必要。

动态加载JavaScript文件的方法总体上有2种:一种是基于运行时增加Html的src节点;另一种是利用JavaScript内置eval函数,将文件内容以字符串形式插入到当前的JavaScript运行环境。第一种方式,增加的JavaScript文件以明文的形式存在于浏览器端,用户使用开发工具就可以轻易获得。第二种方式,新增的JavaScript文件以变量的形式存在于浏览器端,这种方法可以实现隐藏JavaScript文件,避免JavaScript代码被窃取。

本文在第二种JavaScript文件加载基础上,结合Ajax异步请求方式,针对ExtJS框架提出了一种JavaScript文件动态加载模型,并被证明是行之有效的。

## 1 模块化JavaScript文件动态加载模型

要做到JavaScript文件按需加载,首先必须确定浏览器端的用户在进行某项操作时最少需要哪些JavaScript文件。大量的应用表明,按照业务逻辑将JavaScript文件划分成最小粒度比较合适,而业务逻辑通常又是同浏览器端界面一致,即一个业务逻辑对应一个界面。

在使用ExtJS2.0框架的界面里,一个界面的根节点通常是一个Panel类,一个Panel类又是由一个或多个JavaScript文件构成。所以可以将这样的Panel类注册为动态类,也就是将类名构成的字符串与类作HashMap关联。然后将该类与该类实现所需的JavaScript文件路径做HashMap关联。这样最终实现了通过类名对应的字符串把类和类实现所用到的JavaScript做了关联。

代码实现过程4如下:

```
dynamicClass={};
RegisterDynamicClass=function (className,
classType){
    dynamicClass[className]=classType;
}
RegisterDynamicClass("DailyPanel", DailyPanel)
dynamicFile={};
RegisterDynamicFile=function (className,
filelist){
    dynamicFile[className]=filelist;
}
RegisterDynamicFile("DailyPlan", [
    "plan/dailyPlan.js",
    "plan/showWindow.js"
]);
```

收稿日期:2011-08-07

作者简介:熊海东(1986-),男,湖北黄冈人,西华师范大学计算机学院硕士研究生,主要研究方向为方法Web应用开发。

以上代码作为浏览器端的基本 JavaScript 文件的形式存在。经过以上过程就将动态类注册到浏览器端的 JavaScript 执行环境中。经过这一步骤就完成了业务逻辑到 JavaScript 文件的映射。也是实现了将 JavaScript 文件按照业务逻辑模块化。

Ajax 技术最早由 Jesse James Garrett 在 2005 年 2 月提出,他在文章<<Ajax: A New Approach to Web Applications>>中,描述了利用 web2.0 标准实现了浏览器异步动态加载数据的技术。其特点是混合使用 XHTML、CSS、JavaScript、DOM、XMLHttpRequest 进行 Web 开发实现了异步数据加载,浏览器端动态显示。

基于 JavaScript 文件可以按照业务逻辑模块化的特点,结合 Ajax 技术的异步动态加载技术,从而提出了基于 Ajax 的动态 JavaScript 文件加载模型。模型示意图如下:

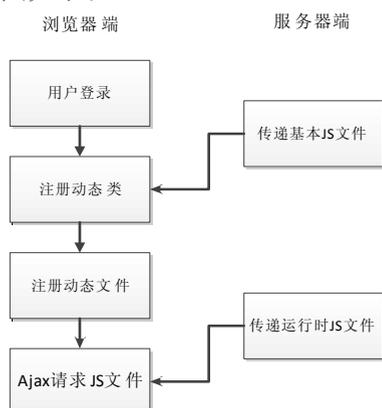


图1 模型示意图

从图1可以看到,系统整个过程都贯彻运行时动态加载思想。浏览器端始终只持有当前运行时必须的 JavaScript 文件。而实现则是通过 Ajax 异步请求,eval 函数插入 JavaScript 文件内容。

## 2 动态加载 JavaScript 文件

表1 未采用动态加载与采用动态加载访问情况对比

加载方式	JavaScript 文件数对比	流量对比(KB)	平均加载时间(S)	最大加载时间(S)
传统加载	39	1007.3	1.42	1.93
动态加载	23	798.4	1.39	2.05

可以看到采用动态加载全面领先,由于首次文件数量的减少,用户等待时间得以减少,而且该对比测试是在局域网环境中进行。若是在互联网环境中,时间对比的情况将更加明显,采用动态加载能明显提高用户体验。

### 3.2 代码可见性对比

某一段的浏览器端特效 JavaScript 代码 code,因为研发人员花了大量的时间和精力,希望浏览器端只是看到特效,而不希望竞争公司研发人员看见代

码。浏览器端用户在某次应用中初次使用某功能模块时,而该类被注册为动态类,也就是 Panel 类所需的 JavaScript 文件还在服务端。这时浏览器端就依据该 Panel 类对应的已经注册的 JavaScript 文件路径,向服务端发起 Ajax 异步请求,这样就能避免浏览器标签的刷新。服务端响应浏览器端的请求将被请求的 JavaScript 文件内容以字符串的形式传给浏览器端。浏览器端接收到服务的字符串后,保存为临时字符串变量,通过 JavaScript 内置全局函数 eval 将该变量插入到当前的 JavaScript 执行环境中。这样就完成了 JavaScript 文件的动态加载,最后实例化该类,形成 Panel 类对象,浏览器端的界面得到渲染更新。

代码实现如下:

```

loadDynamicModule=function(className){
    filelist=DynamicFile[className];
    for(var i=0;i<filelist.length;i++){
        Ajax.request({
            url:Path+filelist[i],
            success:function(response,param){
                eval(param.script);
            }
        });
    }
}
  
```

## 3 对比分析

### 3.1 网络流量及完成加载时间对比

由采用动态加载,浏览器端请求 JavaScript 文件的时间得以分散,系统性能会有一些提升。为此,做这样的测试,同一 Web 系统,在局域网环境下,使用 Firebug 对系统访问时的情况进行统计。未采用动态加载与采用动态加载访问情况对比如表1。

浏览器端查看页面的代码最佳方法莫过于使用 Web 开发工具 Firebug。同样的 JavaScript 代码,使用 Ajax 与不使用 Ajax 在 Firebug 下的可见性对比如表2。

未使用 Ajax,在 Firebug 可以以明文的形式看见特效的核心代码 code,而使用 Ajax 的话,通过开发工具只能看见变量 ajax。可见通过采用 Ajax 能实现隐藏 Web 前端代码,避免 JavaScript 代码被窃取,从而保护开发人员的劳动成果。

表2 使用 Ajax 与不使用 Ajax 在 Firebug 下的可见性对比

未使用 Ajax	使用 Ajax
/** *未采用 Ajax,核心前端代码, *以明文的形式存在 JavaScript 文件, *使用工具可查看 */ alert("Using Ajax");	/** *采用 Ajax,核心前端代码。 *code 是 JavaScript 程序段, *是从服务器动态获取的字符串, *但 code 以变量的形式存在于内存里 *code==alert("Using Ajax"); */ eval(code);

#### 4 结束语

使用本文提出的 JavaScript 文件动态加载模型,对原有的 Web 系统进行代码重构,改善了 Web 系统性能,

提高了质量。但也带来了一些新的问题,大量的使用 eval 函数,会影响浏览器端的性能,JavaScript 代码会变得难以调试。相信随着研究的深入,这些问题会得到解决。

#### 注释及参考文献:

- [1]Nicholas C.Zakas.Professional JavaScript for Web Developers[M].北京:人民邮电出版社,2008:56-62.
- [2]彭仁夔.Ext JS 源码分析与开发实例宝典[M].北京:电子工业出版社,2010:37-45.
- [3]李刚.轻量级 Java EE 企业应用实战-Struts2+Spring+Hibernate 整合开发[M].北京:电子工业出版社,2008:327-350.
- [4]Eric Freeman, Elisabeth Freeman, Kathy Sierra, Bert Bates, Head First 设计模式[M].北京:中国电力出版社,2007:142-150.
- [5]杨俊,李艳梅.JavaScript 面向对象编程解析[J].办公自动化,2010(8).
- [6]刘东.基于 Ext 的富客户端审批系统的设计与实现[D].大连理工大学,2010.
- [7]夏超.基于 Ajax 的 WebGis 系统设计与实现[D].华中科技大学,2010.
- [8]陈玉林.Ajax 在 Web 开发中的轻量级应用[D].湖南科技学院学报,2007.

## Ajax Based Dynamic Loading JavaScript File Model

XIONG Hai-dong<sup>1</sup>, CHEN Ya-jun<sup>2</sup>, PAN Gang<sup>3</sup>

(1.Computer Institute, China West Normal university, Nanchong, Sichuan 637002;  
 2.Experimental Center China West Normal university, Nanchong, Sichuan 637002;  
 3.Computer Institute Sichuan University of Arts and Science, DaZhou, Sichuan 635000)

**Abstract:** In the B/S model websystem, the browser is running with JavaScript language. Browser loads all the JavaScript files only at the first time if the browser doesn't refresh. The JavaScript files which need to be used are not the same for the users with different permissions and the same user at different times. In this paper a dynamic loading JavaScript file model is presented, passing needed JavaScript files only, reducing network traffic, protecting the JavaScript code.

**Key words:** B/S model; Ajax request; JavaScript; Eval; Dynamic load