

文章编号: 1005-8451 (2007) 05-0049-03

AJAX 异步提高 Web 应用效率

孟 莹, 杜林春, 楼新远

(西南交通大学 信息科学与技术学院, 成都 610031)

摘 要: 传统的 Web 技术的弱点是用户在向远程服务器提交请求直到请求返回到客户端的时间内必须处于等待状态, 降低 Web 应用的效率。Ajax 技术的出现有效的克服这个缺点。介绍 Ajax 的基本原理及其子技术, 并通过一个人才管理系统具体实践 Ajax 技术。

关键词: Ajax; XML; XMLHttpRequest 对象; JavaScript; 无刷新更新页面

中图分类号: TP39

文献标识码: A

Improving efficiency of Web application based on Ajax

MENG Ying, DU Lin-chun, LOU Xin-yuan

(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: Web applications had some weaknesses in transmitting datum. Users could do nothing while the server disposed datum. Ajax was such a technique that it could provide the efficiency of Web application. It was introduced the principle of Ajax and a Web application based on Ajax.

Key words: Ajax; XML; XMLHttpRequest object; JavaScript; updating Web

采用 B/S 结构的应用系统, 用户请求时, 强制用户进入提交、等待、重新显示交互结果的过程, 用户在等待服务器处理 HTTP 请求时, 页面处于等待的状态, 这是困扰大多数软件开发商的技术难题。

Ajax 的出现较好地解决了这个难题。Ajax 的核心是 JavaScript 对象 XMLHttpRequest, 它是一种支持异步请求的技术, 可以使用 JavaScript 代码向服务器发出请求并相应处理, 而不阻塞用户, 提高了 Web 应用程序的交互性。本文正是基于 Ajax 技术提出了支持远程数据库访问的 Web 应用系统。

1 Ajax 技术

1.1 Ajax 的组成

Ajax 是 Asynchronous JavaScript and XML 的缩写, 即“异步的 JavaScript 和 XML 处理”或“非同步 JavaScript 及 XML 技术”。Ajax 的组成如图 1 所示^[2]。

Ajax 是几项技术按照一定的方式组合在一起的一种协作方式。Ajax 涵盖:

(1) XMLHttpRequest: XMLHttpRequest 对象允许 Web 程序员从 Web 服务器以后台活动的方式获

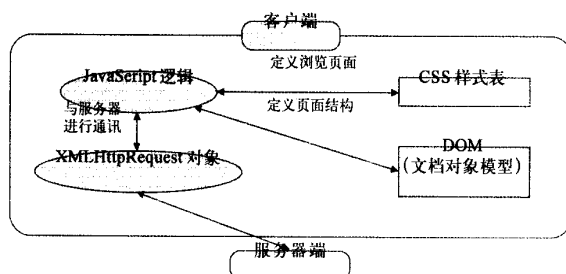


图 1 Ajax 的 4 个主要组件

取数据。它最大的用处是可以动态地更新网页的部分内容而不需要刷新整个页面;

(2) CSS: 层叠样式表, 在 Ajax 应用中, 用户界面的样式可以通过 CSS 独立修改;

(3) DOM: 文档对象模型, 是以一组可以使用 JavaScript 操作的可编程对象展现出 Web 页面的结构。通过使用 JavaScript 脚本修改 DOM, Ajax 应用程序可以在运行时改变用户界面, 或者高效地重绘页面中的某个部分;

(4) JavaScript: JavaScript 是一种粘合剂, 使 Ajax 应用的各部分集成在一起。在 Ajax 中 JavaScript 主要被用来传递用户界面上的数据到服务器并返回结果。本文中 JavaScript 代码主要完成以下几个基本的任务: a. 获取表单数据: JavaScript 代码从 HTML 表单中抽取数据并发送到服务器; b. 修改表单数

收稿日期: 2006-09-26

作者简介: 孟 莹, 在读硕士研究生; 杜林春, 在读硕士研究生。

据：从设置字段值到迅速替换图像；c.解析 HTML 和 XML：使用 JavaScript 代码操纵 DOM，处理 HTML 表单服务器返回的数据的结构。

1.2 Ajax 工作模型

一个 Ajax 应用系统通过在用户和服务器之间引进了一个用 JavaScript 编写的 Ajax 引擎，它的作用主要是消除使用时的断断续续。应用 Ajax 技术首先装载的是 Ajax 引擎。Ajax 引擎负责编译用户界面以及与服务端之间进行交互。它允许用户在应用软件之间的交互过程异步进行，独立于用户与网络服务器间的交流。用户的每一次操作将生成一个到 Ajax 引擎的调用^[3]。Ajax 工作模型如图 2 所示。

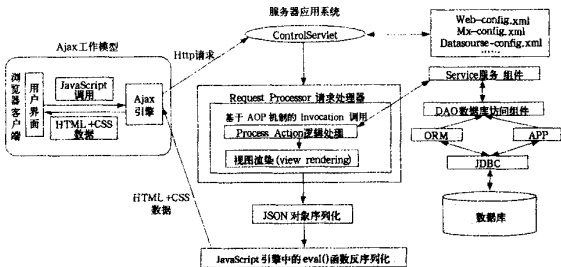


图 2 客户端发送请求处理过程

2 人才管理系统中的 Ajax 实现

2.1 功能简介

现以人才管理为例说明 Ajax 的实现原理。人才管理系统的主要功能包括对人员信息的增加、删除、修改和查询 4 种数据库操作，均在一个 Web 页面实现。Ajax 实质上也是遵循 Request/Server 模式，所以本设计的基本流程如图 2 所示。

2.2 具体实现

(1) 创建请求对象：创建请求对象，要求浏览器使用 XMLHttpRequest 或 ActiveXObject。这 2 个对象之间的主要区别在于使用它们的浏览器以及创建对象的方式。本文主要创建 XMLHttpRequest 对象，方法如下：

```
function getXmlHttpRequest() {
    try { return new XMLHttpRequest(); }
    catch (ex) { try { return new ActiveXObject
('Msxml2.XMLHTTP'); }
    catch (ex1) { try { return new ActiveXObject
('Microsoft.XMLHTTP'); }
    catch(ex1) { return new ActiveXObject
```

```
('Msxml2.XMLHTTP.4.0'); }
    }
    }
}
```

(2) 发送请求：创建了请求对象，即为向服务器发出请求作了准备。通过 XMLHttpRequest 对象与服务器进行对话的是 JavaScript 技术。Ajax 把 JavaScript 技术和 XMLHttpRequest 对象放在 Web 表单和服务器之间，当用户填写表单时，数据发送给一些 JavaScript 代码而不是直接发送给服务器；相反，JavaScript 代码捕获表单数据并向服务器发送请求，同时用户屏幕上的表单不会发生闪烁、消失或延迟，即 JavaScript 代码在幕后发送请求，用户甚至不知道请求的发出。请求是异步发送的，就是说 JavaScript 代码和用户不用等待服务器的响应，因此用户可以继续输入数据、滚动屏幕和使用应用程序。本文中，页面添加一个“查询人员信息”按钮，单击该按钮触发 doInvoke() 函数，具体实现如下：

```
<td><input name="query" value=" 查询人员信息" type="button" onclick="doInvoke(this.form, 'queryAllPage', populateSporterInfos);"></td>
```

调用 JavaScript 函数见下：

```
function doInvoke(form, method, callbackFunction) {
    invokeActionForm(form, method, callbackFunction);
}
```

此方法的 3 个参数分别为：a. form：提交的整个表单对象；b. method：FrameServer 框架中对应于处理该操作的具体实现方法；c. callbackFunction：回调函数，请求返回时调用的函数。

JavaScript 向后台应用程序发送请求的 invokeActionForm() 函数如下：

```
function invokeActionForm(form, method, callbackFunction) {
    var url = form.action;
    if (method) {
        url += "?ajax=true&method=" + method +
getFormValuesAsQueryString(form);
    }
    else {
        url += "?ajax=true&" + getFormValues-
AsQueryString(form).slice(1);
```

```

    }
    return fetchFromUrl(url, callbackFunction);
}

```

此方法的3个参数说明同 doInvoke() 函数。

invokeActionForm() 方法的作用是：将提交的表单中的数据发送到 FrameServer 框架的 action 组件中，同时 method 方法在服务器上被执行，处理用户的请求，callbackFunction 回调函数等待执行。当服务器处理完请求，XMLHttpRequest 对象返回给 JavaScript 代码，回调函数此时被触发。

```

function fetchFromUrl(url, callbackFunction)
{
    var xmlHttpRequest = getXmlHttpRequest();
    // 指定当 readyState 属性改变时的事件处理句柄，即为处理返回信息的函数
    xmlHttpRequest.onreadystatechange = function
    () {
        if (xmlHttpRequest.readyState == 4 &&
            callbackFunction != null) {
            callbackFunction(xmlHttpRequest);
        }
    }
    xmlHttpRequest.open("GET", url, true, "", ""); // true 代表异步方式
    xmlHttpRequest.send(""); // POST 方式通过该函数发送信息
    return xmlHttpRequest; // 返回 xmlHttpRequest 对象
}

```

(3) 应用程序处理请求：请求的处理过程应用了 FrameServer 框架。FrameServer 的功能实际上是一个 MVC 框架(FS_MVC)，其他的机制（如 IOC 容器，AOP 等）都围绕这个 MVC 实现。客户端首先向服务器发起请求，FS_MVC 截获用户请求，并且调用相应的请求处理模块进行处理，最后模型数据通过渲染器进行 Rendering（渲染）以后返回客户端。

a. Dao 组件是直接与数据库进行数据交互的组件；

b. Service 组件是对 Dao 组件的包装。一个 Dao 组件包括所有可能的数据库的增删改查的操作，给程序设计人员带来了许多不便而且代码混乱，Service 组件就像一个门面，将所有用到的 Dao 方法进行包装，给 Action 组件提供了一个方便友好的接口；

c. Action 模块就是捕获用户请求并且调用 Ser-

vice 组件完成业务逻辑功能。每一个 action 通过构造函数注射获取 service 组件，以获得 service 中包装的 Dao 数据访问组件；

d. 渲染：在 Action 组件中进行业务逻辑处理后返回客户端时对结果进行渲染，渲染的主要作用是将请求处理的数据结果转化为数据流传回客户端。

(4) 客户端接收服务器返回的请求：服务器端应用程序处理完请求后经过渲染将数据以字符串的形式返回客户端。主要应用 XMLHttpRequest 对象的 responseText 属性，其作用是将响应信息作为字符串返回。在本文中使用 JSON 算法（JSON，JavaScript Object Notation 是一种轻量级的数据交换格式）将返回的字符串反序列化为 JSON 响应序列，客户端通过 JavaScript 引擎中的 eval() 函数将数据恢复成 Java 对象，它可以迅速地更新表单数据，让人感觉应用程序是立即完成的，表单没有提交或刷新而用户得到了新数据。

3 结束语

应用软件的开发者除了考虑系统的可交互性外还要考虑用户使用方便快捷。Ajax 技术为 B/S 应用程序提供了一种实现良好可交互性的选择，采用它实现的客户端，大部分数据交互在页面之间完成，减少页面组成而且每次提交请求不需要刷新整个页面。本文通过一个具体项目实践说明了应该如何采用 Ajax 来设计用户界面与服务器交互程序的实现方法。但是，Ajax 只是一种技术，没有固定的模式，其具体实现需要根据具体应用进行设计，以便向用户提供更方便、更快捷、更及时的服务。

参考文献：

- [1] Jesse James Garrett. Ajax: A New Approach to Web Applications. Adaptive Path[M]. 2005.
- [2] 李明伍. Ajax 及其在数字图书馆中的应用[J]. 现代情报, 2006 (5): 5.
- [3] Ajax 实战(Ajax in action 中文版)[EB/OL]. <http://book.csdn.net/bookfiles/13/10013151.shtml>, 2006-07-08.
- [4] 如何使用 Ajax 开发 Web 应用程序 [EB/OL]. <http://www.telerss.com/blog/p=35>, 2006-06-07
- [5] 庄幸隆. Java 程序设计—EJB, XML 与数据库[M]. 北京: 清华大学出版社, 2003, 8.