基于 Ajax 技术的表单数据动态验证

Implementation of Dynamic Validity of Form Data Based on Ajax

沈泽刚　王 月　（渤海大学信息科学与工程学院 辽宁锦州 121000）

摘要：Ajax 技术是一种实现 Web 应用开发中交互性设计的方法，它的应用非常广泛。本文讨论了使用 Ajax 技术实现表单数据动态验证的实现方法。

关键词：Ajax　动态页面　表单验证　Web 开发

1　引言

在 Web 应用开发中，表单数据的验证是几乎每个应用都要涉及的功能。有的表单数据验证是采用客户端脚本实现的，如 JavaScript。有的需要服务器端技术，并且经常需要将用户填写的表单数据存储到数据库表中。使用传统的方法，对表单数据的验证只有在表单被提交后才能得到验证。这种方法的缺点是一次传输的数据量较大，并需要等待用户输入全部数据才能验证。

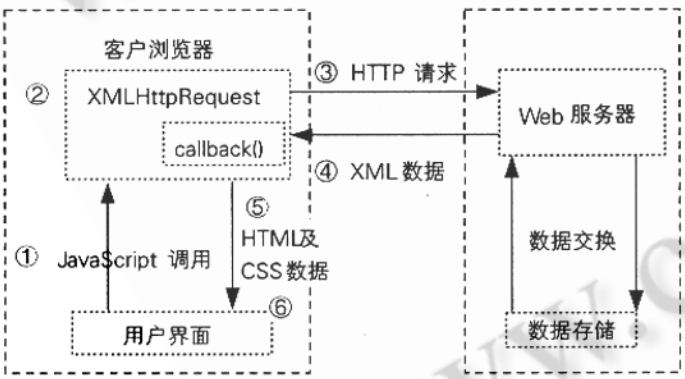

图 1　Ajax 的工作原理

使用 Ajax 技术实现表单数据验证就可以克服上述缺点，实现表单数据的动态验证。Ajax 结合了 JavaScript、DOM（Document Object Model）、CSS 以及 XMLHttpRequest 技术，主要实现在客户和服务器之间的异步动态交互。使用 Ajax 技术，通过服务器的业务逻辑可以实现在数据添加到表单域中时就可以得到动态验证。因此，如果表单中的数据是有效的，就不必将整个表单传送到服务器验证。

本文主要讨论如何使用 Ajax 技术实现表单数据的动态验证。

2　Ajax 主要技术概述

Ajax 技术主要包括下面的内容：

① 级联样式单（CSS），一种定义页面表示风格的标记语言。

② JavaScript，一种脚本语言。对 Ajax 来说关键的元素是 XMLHttpRequest，它用来在客户与服务器之间交换数据。

③ 文档对象模型（DOM），它以树型结构提供了 Web 页面的逻辑视图。

④ XML，从 Web 服务器向客户发送数据的一种格式。但也可以使用其他格式，如 HTML、JSON 或文本。

与其他 Web 应用一样，基于 Ajax 的 Web 应用也使用 HTML 或 XHTML 标记语言展示页面，或者使用如 JSP 技术从服务器端生成 Web 页面。此外，在处理 Ajax 应用中，服务器端应用系统起到了关键作用。像具有数据验证、用户标识管理以及持久性功能的 Java EE 等服务器应用系统就非常适合使用 Ajax 技术. 图 1 说明了 Ajax 的工作原理。

Ajax 实现异步通信的过程如下：

① 通过 JavaScript 技术，用户在客户端产生事件。

② JavaScript 函数在客户端创建和配置一个 XMLHttpRequest 对象，并指定 JavaScript 回调函数。

③ XMLHttpRequest 对象向 Web 服务器发送一个异步的 HTTP 请求。

④ Web 服务器处理请求并返回包含结果的 XML

文档。

⑤ XMLHttpRequest 对象调用 callback( ) 函数对响应进行处理。

⑥ 客户使用新的数据更新 HTML DOM 页面表示元素。

3　表单数据动态验证的实现

在本文创建的 Ajax 应用中，通过 HTML 表单验证数据输入。填加到 HTML 表单中的数据将异步发送到服务器。在服务器端通过 HTTP Servlet 处理表单的输入并将响应作为 XML DOM 对象返回。

3.1　发送请求

XMLHttpRequest 对象根据表单的需要验证的 uno 域进行初始化。当表单上发生 onkeyup 事件时将调用 JavaScript 函数 validateUno( )，下面是表单代码：

```
<form name = " validationForm"
  action = " validateForm"  method = " post" >
<table >
 <tr > <td > 用户号：</td >
   <td > <input type = " text"  size = " 20"  id = "
uno"  name = " uno"
       autocomplete = " off"  onkeyup = " validate-
Uno( )" > </td >
   <td > <div id = " validationMessage" > </div
> </td >
 </tr >
 ....
</table > </form >
```

在 JavaScript 的 validateUno( ) 函数中创建一个新的 XMLHttpRequest 对象：

```
var xmlHttpRequest;
function createXMLHttpRequest( ) {
  if (window. XMLHttpRequest) {
    xmlHttpRequest =
      new XMLHttpRequest( );
  }
  else if (window. ActiveXObject) {
    xmlHttpRequest =
      new ActiveXObject ( " Microsoft. XMLHT-
TP" );
  }
}
```

接下来需要打开到服务器的连接请求。在服务器端使用一个名为 FormValidationServlet 的 Servlet 处理该请求，它被映射到 validateForm。另外还要传递一个请求参数 uno，参数值通过 JavaScript 函数 encodeURI-Component(string) 进行编码。请求的代码如下：

```
var uno = document. getElementById( " uno" );
xmlHttpRequest. open( " GET" ,
  " validateForm? uno = " +
  encodeURIComponent( uno. value) , true);
```

XMLHttpRequest 对象的 onreadystatechange 属性存储了回调函数的指针，当它的内部状态发生变化时，将调用这个回调函数。XMLHttpRequest 对象的 send( ) 方法把请求发送到指定的目标资源。send( ) 方法接受一个参数，通常是一个串或 DOM 对象。这个参数作为请求体的一部分发送到目标 URL。如果为 send( ) 方法指定了参数，要保证 open( ) 方法中指定的方法为 POST。若没有数据作为请求体发送，其参数为 null。

```
xmlHttpRequest. onreadystatechange =
      processRequest;
xmlHttpRequest. send( null);
```

上述代码中将回调函数指定为 processRequest，当 XMLHttpRequest 对象的 readyState 属性发生改变时将调用该函数，该函数的定义如下：

```
function processRequest( ) {
 if( xmlHttpRequest. readyState = =4) {
   if( xmlHttpRequest. status = =200) {
     processResponse( );
   }
 }}
```

3.2　在服务器端处理请求

XMLHttpRequest 对象通过指定的 URL 发送到服务器端。由于 validateForm 被映射到 FormValidationServlet，请求方法使用的是 GET，将调用 Servlet 的 doGet( ) 方法，在 doGet( ) 方法中检索 uno 参数值，代码如下：

```
String uno = request. getParameter( " uno" );
```

为了从数据库中获得数据，使用下面代码创建一个 JDBC 连接：

```
OracleDataSource ds =
```

```
        new OracleDataSource();
    ds.setURL(jdbcUrl);
    conn =
        ds.getConnection(userid, password);
```

创建一个 Statement 对象，使用 SQL 查询根据输入表单中的 uno 值从数据库中检索数据，通过执行 Statement 对象的 executeQuery() 方法获得 ResultSet 对象，代码如下：

```
    Statement stmt = conn.createStatement();
    String query = "SELECT * FROM User
    WHERE uno = " + "'" +uno + "'";
    ResultSet rs = stmt.executeQuery(query);
```

从 Servlet 返回的响应是一个 XML DOM 对象，该对象的构建是根据 uno 域的值确定的。如果 ResultSet 对象为空，表明 uno 域的值没有在 User 表中定义，uno 域的值是有效的。如果 ResultSet 对象中有数据，表明 uno 域的值已在中定义，uno 域值是无效的。在 XML DOM 对象中使用 <valid> </valid> 元素指定 uno 域的值的有效性：

```
    if (rs.next()) {
      out.println("<user>" +
      "<valid>false</valid>" + "<uname>"
+
      rs.getString(2) + "</uname>" + "</user
>");
    } else {
      out.println("<valid>true</valid>");
    }
```

如果 uno 域的值在数据库中没有定义，将使用 JDBC 数据库连接，利用 SQL 的 INSERT 语句向数据库表中插入一用户信息。

3.3 处理响应

根据 XMLHttpRequest 对象的 readyState 属性和 status 属性值来处理请求。readyState 属性值为 4 表示请求完成，status 属性值为 200 对应于 HTTP 的"OK"状态，表示请求成功。在上述状态下将调用 processResponse() 函数，在该函数中通过下列语句获得 responseXML 属性值：

```
var xmlMessage
    =xmlHttpRequest.responseXML;
```

responseXML 对象是一个 XML DOM 对象，使用该对象的 getElementsByTagName(String) 方法可以获得 <valid/> 元素的值，代码如下：

```
var valid =
  xmlMessage.getElementsByTagName(
    "valid")[0].firstChild.nodeValue;
if(valid == "true"){
  var validationMessage =
    document.getElementById(
      "validationMessage");
  validationMessage.innerHTML = "用户号合
法";
  document.getElementById("submitForm").dis-
abled = false;
}
if(valid == "false"){
  var validationMessage = document.getElement-
ById("validationMessage");
  validationMessage.innerHTML = "用户号非法";
  document.getElementById("submitForm").dis-
abled = true;
}
```

如果 <valid/> 元素的值为 true，将 validationMessage 的值设置为"用户号合法"并使提交按钮可用，否则设置为"用户号非法"并使提交按钮不可用。

4 小结

Ajax 技术通过 XMLHttpRequest 对象实现浏览器与服务器的异步通信，在 Web 应用中最大的优点是无需完全刷新页面，与服务器无缝通信并更新部分页面内容，这可以大大改善用户的体验。

参考文献

1 [美]Ryan Asleson, Nathaniel T. Schutta 著，Ajax 基础教程[M]，北京：人民邮电出版社，2006.3.

2 张桂元 等，征服 Ajax－Web 2.0 快速入门与项目实践(Java)[M]，北京：人民邮电出版社，2006.6.

3 Deepak Vohra. Validating Forms with Ajax [OL]. http://www.oracle.com/technology/pub/articles/vohra－ajax.html. 2006.5.