**Course: Web Application Development**
**Lab Instructor: Dr. Nguyen Van Sinh**
**Email: nvsinh@hcmiu.edu.vn**

# Lab 09 - AJAX

**Content:**

- Introduction
- Practice
- Exercise

**Duration**: 3 hours

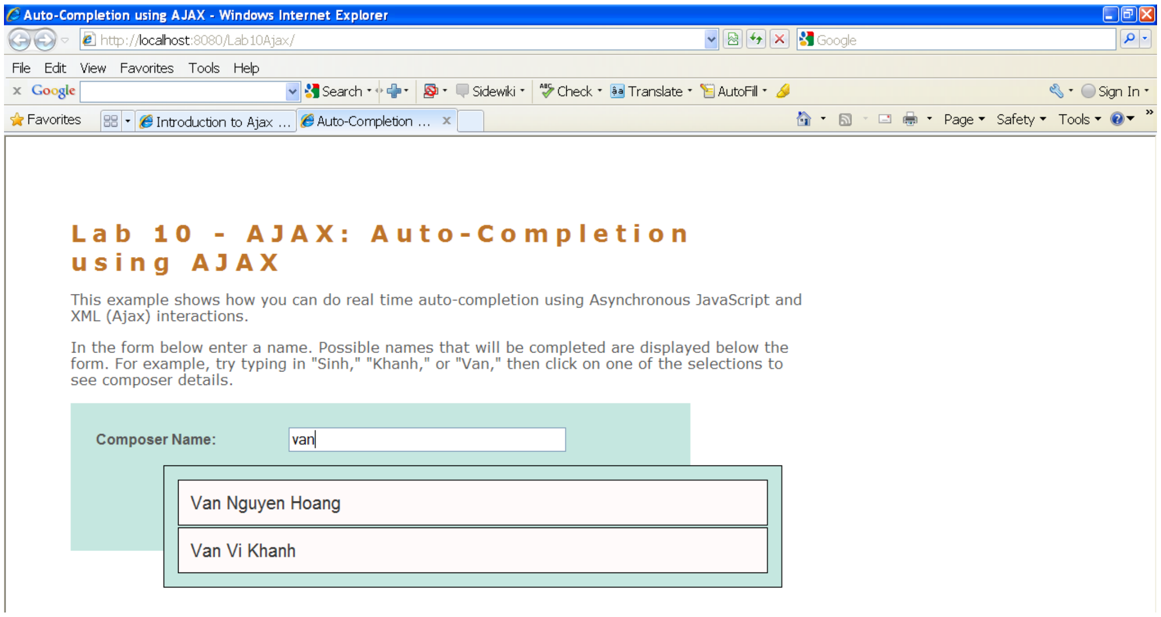
## Part 1: Introduction

Ajax stands for Asynchronous JavaScript and XML. In essence, Ajax is an efficient way for a web application to handle user interactions with a web page -- a way that reduces the need to do a page refresh or full page reload for every user interaction. This enables rich behavior (similar to that of a desktop application or plugin-based web application) using a browser. Ajax interactions are handled asynchronously in the background. As this happens, a user can continue working with the page. Ajax interactions are initiated by JavaScript code. When the Ajax interaction is complete, JavaScript updates the HTML source of the page. The changes are made immediately without requiring a page refresh. Ajax interactions can be used to do things such as validate form entries (while the user is entering them) using server-side logic, retrieve detailed data from the server, dynamically update data on a page, and submit partial forms from the page.

The NetBeans IDE provides strong support for Ajax development, including JavaScript and CSS editors, an all new PHP editor, a JavaScript debugger which is based on the Firebug plugin for Firefox, a client-side monitor to examine HTTP messages, an Ajax-ready environment for your choice of server-side scripting language (e.g., JSP, PHP, Ruby, Groovy), out-of-the-box support for MySQL and Java DB, and extensive integration support for web frameworks and third-party JavaScript toolkits.

## Part 2: Simple Example

In this part, you will implement an application which is combined with HTML, CSS, JavaScript, Java and JSP.


Auto-Completion using AJAX - Windows Internet Explorer
http://localhost:8080/Lab10Ajax/
File Edit View Favorites Tools Help
Favorites
Introduction to Ajax ...
Auto-Completion ...

Lab 10 - AJAX: Auto-Completion using AJAX

This example shows how you can do real time auto-completion using Asynchronous JavaScript and XML (Ajax) interactions.

In the form below enter a name. Possible names that will be completed are displayed below the form. For example, try typing in "Sinh," "Khanh," or "Van," then click on one of the selections to see composer details.

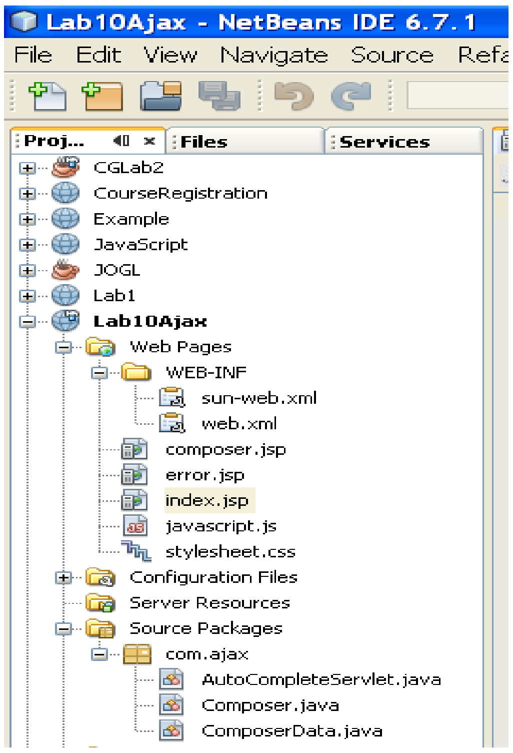
Composer Name: van

Van Nguyen Hoang

Van Vi Khanh

This is the main interface

**Step1:** Choose File > New Project. Under Categories, select Web. Under Projects, select Web Application and click Next.
In the Name and Location panel, enter Lab10Ajax for Project Name. The Project Location field enables you to specify the location of the project on your computer. Leave other options at their defaults and click Next.
You also create many pages as the diagram below:


Lab10Ajax - NetBeans IDE 6.7.1
File Edit View Navigate Source Refa
Proj... Files Services

```
CGLab2
CourseRegistration
Example
JavaScript
JOGL
Lab1
Lab10Ajax
  Web Pages
    WEB-INF
      sun-web.xml
      web.xml
    composer.jsp
    error.jsp
    index.jsp
    javascript.js
    stylesheet.css
  Configuration Files
  Server Resources
  Source Packages
    com.ajax
      AutoCompleteServlet.java
      Composer.java
      ComposerData.java
```

**Step2:**
**- Copy source code below and past to inside the body tag of file Index.jsp:**

```
 <form name="autofillform" action="autocomplete">
           <table border="0" cellpadding="5">
               <tbody>
                   <tr>
                       <td><strong>Composer Name:</strong></td>
                       <td>
                           <input type="text"
                                  size="40"
                                  id="complete-field"
                                  onkeyup="doCompletion()">
                       </td>
                   </tr>
                   <tr>
                       <td id="auto-row" colspan="2">
                                             <table  id="complete-table"
class="popupBox" />
                       </td>
                   </tr>
               </tbody>
           </table>
        </form>
```

**- Copy source code below and past to inside the body tag of file error.jsp:**

```
      <html>
          <head>
                              <link   rel="stylesheet"   type="text/css"
      href="stylesheet.css">
                    <meta  http-equiv="Content-Type"  content="text/html;
      charset=UTF-8">
              <title>Seach Error</title>
          </head>
          <body>
              <h2>Seach Error</h2>

                <p>An error occured while performing the search. Please
      try again.</p>

              <p>Go back to <a href="index.jsp" class="link">application
      home</a>.</p>
          </body>
      </html>
```

**- Copy source code below and past to inside the body tag of file composer.jsp:**

```
      <html>
        <head>
          <title>Composer Information</title>

          <link rel="stylesheet" type="text/css" href="stylesheet.css">
        </head>
```

```
  <body>

    <table>
      <tr>
        <th colspan="2">Composer Information</th>
      </tr>
      <tr>
        <td>First Name: </td>
        <td>${requestScope.composer.firstName}</td>
      </tr>
      <tr>
        <td>Last Name: </td>
        <td>${requestScope.composer.lastName}</td>
      </tr>
      <tr>
        <td>ID: </td>
        <td>${requestScope.composer.id}</td>
      </tr>
      <tr>
        <td>Category: </td>
        <td>${requestScope.composer.category}</td>
      </tr>
    </table>

      <p>Go back to <a href="index.jsp" class="link">application
home</a>.</p>
  </body>
</html>
```

**- Copy source code below and past to inside the body tag of file javascript.js:**

```
var req;
var isIE;
var completeField;
var completeTable;
var autoRow;

function init() {
    completeField = document.getElementById("complete-field");
    completeTable = document.getElementById("complete-table");
    autoRow = document.getElementById("auto-row");
    completeTable.style.top = getElementY(autoRow) + "px";
}

function doCompletion() {

            var url = "autocomplete?action=complete&id=" +
escape(completeField.value);
    req = initRequest();
    req.open("GET", url, true);
    req.onreadystatechange = callback;
    req.send(null);
}

function initRequest() {
    if (window.XMLHttpRequest) {
        if (navigator.userAgent.indexOf('MSIE') != -1) {
            isIE = true;
        }
        return new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        isIE = true;
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
}

function callback() {
    clearTable();

    if (req.readyState == 4) {
        if (req.status == 200) {
            parseMessages(req.responseXML);
        }
    }
}

function appendComposer(firstName,lastName,composerId) {

    var row;
    var cell;
    var linkElement;

    if (isIE) {
        completeTable.style.display = 'block';
        row = completeTable.insertRow(completeTable.rows.length);
        cell = row.insertCell(0);
    } else {
        completeTable.style.display = 'table';
        row = document.createElement("tr");
        cell = document.createElement("td");
        row.appendChild(cell);
        completeTable.appendChild(row);
    }

    cell.className = "popupCell";

    linkElement = document.createElement("a");
    linkElement.className = "popupItem";
                linkElement.setAttribute("href",    "autocomplete?
action=lookup&id=" + composerId);
     linkElement.appendChild(document.createTextNode(firstName + "
" + lastName));
    cell.appendChild(linkElement);
}

function clearTable() {
    if (completeTable.getElementsByTagName("tr").length > 0) {
        completeTable.style.display = 'none';
          for (loop = completeTable.childNodes.length -1; loop >=
0 ; loop--) {
            completeTable.removeChild(completeTable.childNodes[loo
p]);
        }
    }
}

function getElementY(element){

    var targetTop = 0;

    if (element.offsetParent) {
        while (element.offsetParent) {
            targetTop += element.offsetTop;
            element = element.offsetParent;
        }
    } else if (element.y) {
        targetTop += element.y;
    }
    return targetTop;
}

function parseMessages(responseXML) {

    // no matches returned
    if (responseXML == null) {
        return false;
    } else {

                                      var     composers     =
responseXML.getElementsByTagName("composers")[0];

        if (composers.childNodes.length > 0) {
            completeTable.setAttribute("bordercolor", "black");
            completeTable.setAttribute("border", "1");

              for (loop = 0; loop < composers.childNodes.length;
loop++) {
                var composer = composers.childNodes[loop];
                                               var   firstName   =
composer.getElementsByTagName("firstName")[0];
                                                var   lastName   =
composer.getElementsByTagName("lastName")[0];
                                              var   composerId   =
composer.getElementsByTagName("id")[0];
                appendComposer(firstName.childNodes[0].nodeValue,
                    lastName.childNodes[0].nodeValue,
                    composerId.childNodes[0].nodeValue);
            }
        }
    }
}
```

**- Copy source code below and past to inside the body tag of file stylesheets.css:**

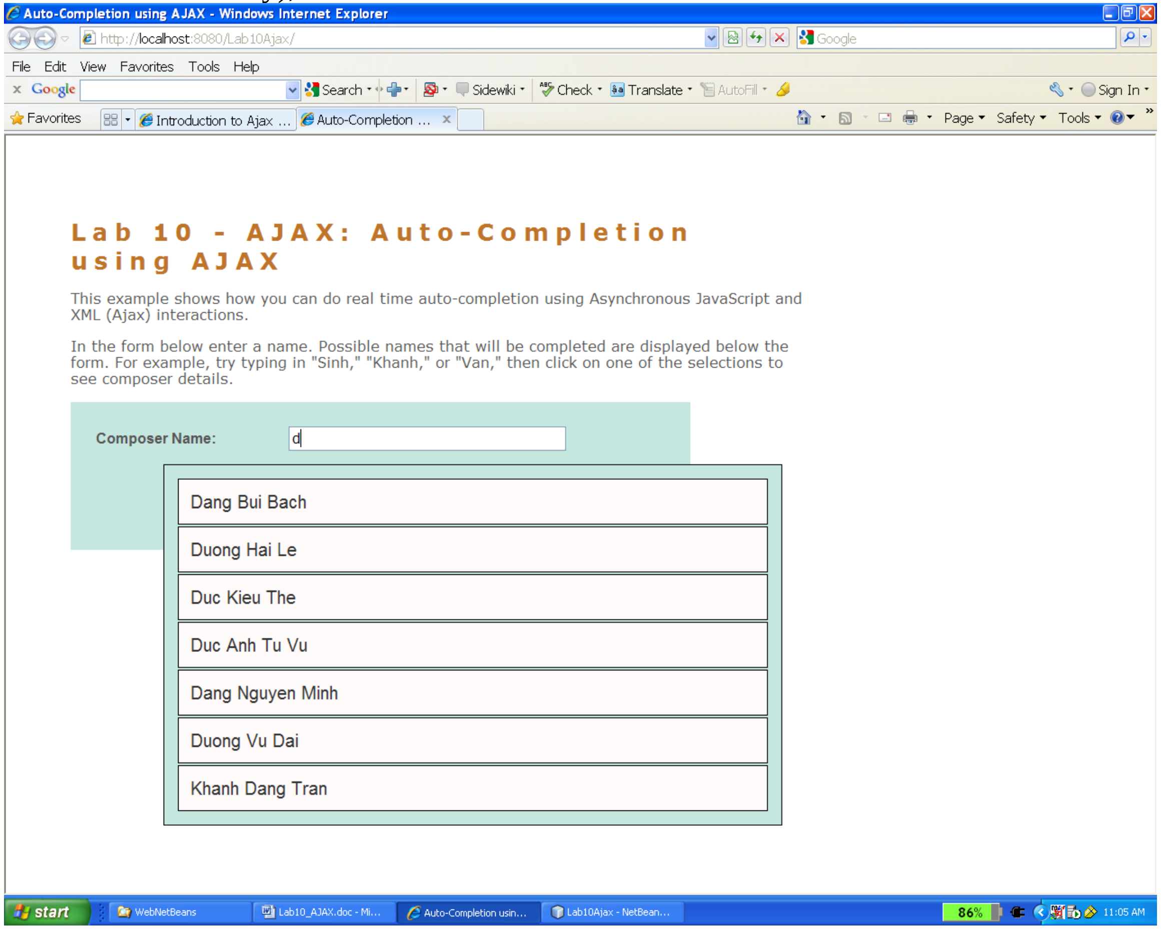
```
    body {
       font-family: Verdana, Arial, sans-serif;
       font-size: smaller;
       padding: 50px;
       color: #555;
```

```
   width: 650px;
}

h1 {
   letter-spacing: 6px;
   font-size: 1.6em;
   color: #be7429;
   font-weight: bold;
}

h2 {
   text-align: left;
   letter-spacing: 6px;
   font-size: 1.4em;
   color: #be7429;
   font-weight: normal;
   width: 450px;
}

table {
   width: 550px;
   padding: 10px;
   background-color: #c5e7e0;
   font-family: sans-serif;
}

td {
   padding: 10px;
}

a {
  color: #be7429;
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

.popupBox {
  position: absolute;
  left: 140px;
}

.popupCell {
   background-color: #fffafa;
}

.popupCell:hover {
  background-color: #f5ebe9;
}

.popupItem {
  color: #333;
  text-decoration: none;
  font-size: 1.2em;
}
```

**- Copy source code of three java files form teacher's computer and paste to AutoCompleteServlet.java; Composer; ComposerData:**

- ComposerData.java that retains composer data using a [HashMap](HashMap). A HashMap allows you to store pairs of linked items in key-value pairs. You also create a Composer class that enables the servlet to retrieve data from entries in the HashMap.

When finish, right click at the project's name and build (remember: you must check source code carefully), the result should as below:



Lab 10 - AJAX: Auto-Completion using AJAX

This example shows how you can do real time auto-completion using Asynchronous JavaScript and XML (Ajax) interactions.

In the form below enter a name. Possible names that will be completed are displayed below the form. For example, try typing in "Sinh," "Khanh," or "Van," then click on one of the selections to see composer details.

Composer Name: d

Dang Bui Bach
Duong Hai Le
Duc Kieu The
Duc Anh Tu Vu
Dang Nguyen Minh
Duong Vu Dai
Khanh Dang Tran

## Part 3: Assignment 4:

Apply this application into your project!