**Tutorial: Create Autocomplete feature with Java – JSP / jQuery**

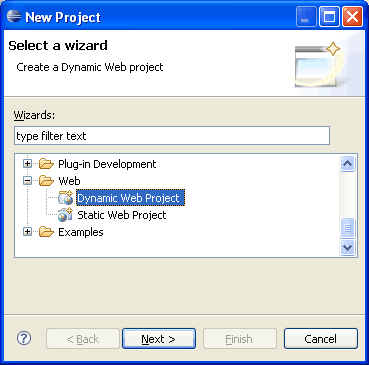
* By [Viral Patel](http://viralpatel.net/blogs/author/viral/) on June 8, 2009

Autocomplete is a common feature available in lot of web tools and services. You will find lots of implementation of autocomplete features. Let us see how can we implement a simple Autocomplete feature for Country names in Java-JSP and jQuery.

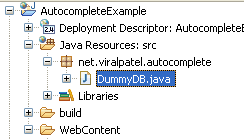
**Related:** [Spring MVC – AutoComplete tutorial with JSON & JQuery](http://viralpatel.net/blogs/spring-3-mvc-autocomplete-json-tutorial/)

We will use [jQuery’s autocomplete plugin](http://jqueryui.com/demos/autocomplete/) for our example. Just go to JQuery Autocomplete UI page and download JQuery UI JS and CSS files. I have used Eclipse for the development of this demo.

Create a dymanic web project by clicking File -> New -> Project -> Dynamic Web Project and name it**AutocompleteExample**.



Once the project is created, create a package **net.viralpatel.autocomplete** and a Java class file**DummyDB.java**. DummyDB.java is the class that will simulate the database connection and it will provide the data for our example.



Following is the content of DummyDB.java.

package net.viralpatel.autocomplete;

import java.util.ArrayList;

import java.util.List;

import java.util.StringTokenizer;

public class DummyDB {

private int totalCountries;

private String data = "Afghanistan, Albania, Zimbabwe";

private List<String> countries;

public DummyDB() {

countries = new ArrayList<String>();

StringTokenizer st = new StringTokenizer(data, ",");

while(st.hasMoreTokens()) {

countries.add(st.nextToken().trim());

}

totalCountries = countries.size();

}

public List<String> getData(String query) {

String country = null;

query = query.toLowerCase();

List<String> matched = new ArrayList<String>();

for(int i=0; i<totalCountries; i++) {

country = countries.get(i).toLowerCase();

if(country.startsWith(query)) {

matched.add(countries.get(i));

}

}

return matched;

}

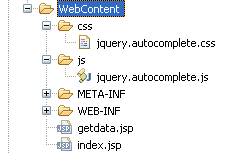
}

The DummyDB.java contains the list of all the countries in a comma separated string value and a methodgetData () that will return the list of countries starting with the string query passed as argument to that method. Thus if we pass “IN” to this method, it will return as all the countries starting with IN.

You may want to change this code and add the database implementation here. Just a simple “SELECT \* FROM <table> WHERE country LIKE ” query will serve the purpose.

Once this is done, we will create the JSP files that will be called by the autocomplete plugin to get the data. Download [jquery.autocomplete.css](http://www.pengoworks.com/workshop/jquery/lib/jquery.autocomplete.css) and [jquery.autocomplete.js](http://www.pengoworks.com/workshop/jquery/lib/jquery.autocomplete.js) as you will need it in our demo.

Copy jquery.autocomplete.css into **css** folder in WebContent and jquery.autocomplete.js into **js** folder.



Create **index.jsp** and paste following code into it.

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<link rel="stylesheet" type="text/css" href="css/jquery.autocomplete.css" />

<script type="text/javascript"

src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js"></script>

<script src="js/jquery.autocomplete.js"></script>

<style>

input {

font-size: 120%;

}

</style>

</head>

<body>

<h3>Country</h3>

<input type="text" id="country" name="country"/>

<script>

$("#country").autocomplete("getdata.jsp");

</script>

</body>

</html>

**Update:** In above code I have removed the old way of including jQuery library from Google AJAX API and changed it to simple <script> tag.  
index.jsp will have a textbox “country” and it calls the server code for doing autocompletion by calling following code:

$("#country").autocomplete("getdata.jsp");

Note that the #country is the id of the textbox where we want to add autocomplete feature and the “getdata.jsp” is the server script that gets called.

Next, Create getdata.jsp and copy following code into it.

<%@page import="java.util.Iterator"%>

<%@page import="java.util.List"%>

<%@page import="net.viralpatel.autocomplete.DummyDB"%>

<%

DummyDB db = new DummyDB();

String query = request.getParameter("q");

List<String> countries = db.getData(query);

Iterator<String> iterator = countries.iterator();

while(iterator.hasNext()) {

String country = (String)iterator.next();

out.println(country);

}

%>

getdata.jsp will just call DummyDB class’s getdata method to fetch list of countries and print it in the response. One thing we need to note here is that the Autocomplete plugin will take input separated by line feed, that is the list of countries should come in separate line. Following can be the desired output:

Denmark

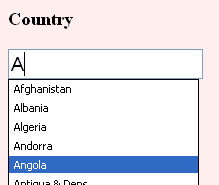
Djibouti

Dominica

Dominican Republic

East Timor

Ecuador

Execute the code in eclipse and you will get the desired output:  


来源： <<http://viralpatel.net/blogs/tutorial-create-autocomplete-feature-with-java-jsp-jquery/>>