

## Homework: Tripadvisor.com & Facebook Mashup – an AJAX/JSON/Java Exercise

### 1. Objectives

- Become familiar with the AJAX, REST, JSON & XML technologies;
- Use a combination of HTML, CSS, DOM, XMLHttpRequest, XML and Java Servlets;
- Provide an interface to search and display hotel information from tripadvisor.com and post them to Facebook.

### 2. Background

#### 2.1 AJAX & JSON

AJAX (Asynchronous JavaScript + XML) incorporates several technologies:

- Standards-based presentation using XHTML and CSS;
- Dynamic display and interaction using the Document Object Model (DOM);
- Data interchange and manipulation using XML and XSLT;
- Asynchronous data retrieval using XMLHttpRequest;
- JavaScript binding everything together.

See the class slides at <http://www-scf.usc.edu/~csci571/Slides/ajax.ppt>.

JSON, short for JavaScript Object Notation, is a lightweight data interchange format. Its main application is in AJAX web application programming, where it serves as an alternative to the use of the XML format for data exchange between client and server. See the class slides at <http://www-scf.usc.edu/~csci571/Slides/JSON1.ppt>.

#### 2.2. Tripadvisor.com

Tripadvisor.com is an online service that gives you free access to find about a hotel's information and its review on the web. The Tripadvisor.com homepage is available at:

<http://www.tripadvisor.com/>

In Homework #6 a Perl (or PHP,Python) script together with your Apache server provided the “scrape” functionality. In this exercise you will re-use your scraping code to produce XML instead of HTML as you did in homework #6.

## 2.3 Facebook

**Facebook** is a global social networking website that is operated and privately owned by Facebook, Inc. Users can add friends and send them messages, and update their personal profiles to notify friends about themselves and what they are doing.

Users can additionally post news feeds to their profiles, and these feeds may include images, besides text messages.

The Facebook homepage is available at:

<http://www.facebook.com>

Facebook provides developers with an application-programming interface, called the Facebook Platform. A component of this platform is Facebook Connect, a powerful set of APIs for developers that allows them to authorize users, and send communications to Facebook, among many more things.

Below are few links for Facebook Connect:

<http://developers.facebook.com/blog/post/108/>

<http://developers.facebook.com/docs/guides/web/>

## 3. Description of the Exercise

In this exercise, you will write a web application that does the following sequence of actions:

- a) Allows a user to enter a “query” for hotel information from tripadvisor.com; the query will contain a hotel chain and city name (see Figure 1);
- b) Uses the query string to retrieve hotel information from tripadvisor.com, using the Apache/Perl CGI (or PHP, Python) from Homework #6;
- c) Displays the hotel information in the UI, in the same or similar format used in Homework #6 (see Figure 1);
- d) Allows the user to select a hotel information and Post it to Facebook by clicking on a Facebook button;
- e) Authorizes (logs in) the user to Facebook (see Figure 2);
- f) Posts a story feed of the hotel’s information to a user’s Facebook feed using the Facebook Connect API. The story to be posted has an attachment, which includes a “title” with the hotel name; a “message” with the hotel location and rating and image. The title and image will hyperlink to the reviews. (The attachment message will be “The hotel is located in XXX, YYYY and has a rating of ZZ”, as in “The hotel is located in San Francisco, California and has a rating if 5”), Users can enter the message of their choice in the message box. A snapshot is shown in Figure 3.

A snapshot of the initial user interface, together with a query and the resultant table, is shown in Figure 1.

## Hotel Search

City:




San Fransisco

Hotel chain:

Hyatt

▼

Search

Image	Name	Location	Rating out of 5	Reviews	Post to Facebook
	Hyatt Regency San Francisco	San Francisco, California	4	<a href="#">800</a>	<a href="#">facebook</a>
	Hyatt at Fisherman's Wharf	San Francisco, California	4	<a href="#">655</a>	<a href="#">facebook</a>
	Hyatt San Francisco	San Francisco, California	4	<a href="#">850</a>	<a href="#">facebook</a>

**Figure 1. User Interface**

To implement this exercise you are required to write a combination of HTML, JavaScript and Java Servlet programs. The top-level interface consists of three areas:

- A Form area including an edit box to enter the city name, a dropdown to select the hotel chain and a “Search” button;
- A dynamic area that displays a table with the hotel information corresponding to the query, additional properties of the hotel (name, location, rating, no of reviews along with the link to view them when clicked and hotel figure);
- A “Facebook” button that posts the story feed having attachment of details, corresponding to the selected hotel, to a user’s feed page.

Once city has been entered in the edit box along with the selection of the hotel chain, and the “Search” button is pressed, the form calls a JavaScript function. This function first performs basic validation of the input. If the input is empty, the request is not forwarded to the Java Servlet. Instead an alert with an error message is presented to the user, asking

to refine the search. Once the validation is successful, the JavaScript function executes an XMLHttpRequest to start an asynchronous transaction with a Java Servlet running under Tomcat, and passing the “query strings” as parameters of the transaction.

The Java Servlet in turn performs the following three steps:

First, the Java Servlet extracts the query string and then it calls the “modified” Perl CGI (or PHP, Python) program from Homework #6 to retrieve data from tripadvisor.com corresponding to the city name and hotel chain included in the query. For example, if your server was using port 8866, the following REST query with the query string having city name “San Francisco” and hotel chain “Hyatt”:

`http://cs-server.usc.edu:8866/cgi-bin/hotel.pl?city=San+Francisco&chain=Hyatt`

would return the following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<hotels total="3">

  <hotel name="Hyatt Regency San Francisco" location="San Francisco,
California" no_of_stars="4" no_of_reviews="800"
image_url="http://mediacdn.tripadvisor.com/media/ProviderThumbnails/dir
s/ce/6b/ce6b15fe895b93fa5b5502dfe653cde55.jpg"
review_url="http://www.tripadvisor.com/Hotel_Review-g60713-d81103-
Reviews-Hyatt_Regency_San_Francisco-San_Francisco_California.html
#REVIEWS" />

  <hotel name="Hyatt at Fisherman's Wharf" location="San Francisco,
California" no_of_stars="4" no_of_reviews="655"
image_url="http://mediacdn.tripadvisor.com/media/ProviderThumbnails/dir
s/a0/9a/a09a782883a390cb5ee08f6441c9b6cb5.jpg"
review_url="http://www.tripadvisor.com/Hotel_Review-g60713-d81087-
Reviews-Hyatt_at_Fisherman_s_Wharf-San_Francisco_California.html
#REVIEWS" />

  <hotel name="Hyatt San Francisco" location="San Francisco,
California" no_of_stars="4" no_of_reviews="850"
image_url="http://mediacdn.tripadvisor.com/media/ProviderThumbnails/dir
s/f4/4e/f44eb92cd4f50d967689caa0d9c2d0e97.jpg"
review_url="http://www.tripadvisor.com/Hotel_Review-g60713-d80999-
Reviews-Grand_Hyatt_San_Francisco-San_Francisco_California.html
#REVIEWS" />

</hotels>
```

Notice that in Homework #6 your CGI Perl script produced HTML. In this exercise, the output must be changed to XML.

Second, the Java Servlet extracts from this XML the **hotel information**.

Lastly, the Java Servlet produces a JSON file that is returned asynchronously to the original XMLHttpRequest.

The format of the JSON file that needs to be generated is as follows:

```
{
  "hotels":{
    "hotel":[
      {
        "name":"Hyatt Regency San Francisco",
        "location":"San Francisco, California",
        "no_of_stars":"4",
        "no_of_reviews":"800",
        "image_url":"http://mediacdn.tripadvisor.com/media/ProviderThumbnais/dirs/ce/6b/ce6b15fe895b93fa5b5502dfe653cde55.jpg",
        "review_url":" http://www.tripadvisor.com/Hotel_Review-g60713d81103-Reviews- Hyatt_Regency_San_Francisco-San_Francisco_California.html#REVIEWS "
      },
      {
        "name":"Hyatt at Fisherman's Wharf",
        "location":"San Francisco, California",
        "no_of_stars":"4",
        "no_of_reviews":"655",
        "image_url":"http://mediacdn.tripadvisor.com/media/ProviderThumbnail s/dirs/a0/9a/a09a782883a390cb5ee08f6441c9b6cb5.jpg ",
        "review_url":"http://www.tripadvisor.com/Hotel_Review-g60713-d81087-Reviews-Hyatt_at_Fisherman_s_Wharf-San_Francisco_California.html#REVIEWS "
      },
      {
        "name":"Hyatt San Francisco",
        "location":"San Francisco, California",
        "no_of_stars":"4",
        "no_of_reviews":"850",
        "image_url":"http://mediacdn.tripadvisor.com/media/ProviderThumbnail s/dirs/f4/4e/f44eb92cd4f50d967689caa0d9c2d0e97.jpg",
        "review_url":"http://www.tripadvisor.com/Hotel_Review-g60713-d80999-Reviews-Grand_Hyatt_San_Francisco-San_Francisco_California.html#REVIEWS"
      }
    ]
  }
}
```

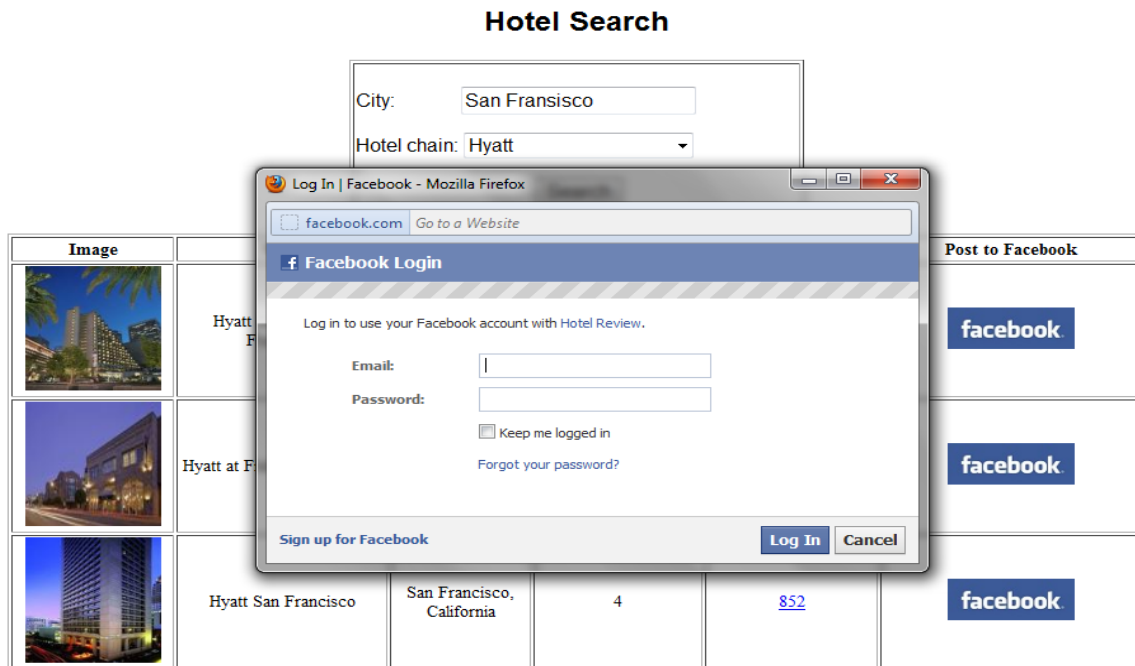
After obtaining the query results from the callback of XMLHttpRequest, the JavaScript program displays the table with the hotel information properties in the “dynamic” area of the web page. **Also note, successive queries will clear the data of the dynamic area and overwrite it with new data.**

Next, the user is allowed to click on a “Facebook” button adjacent to the hotel in the table. When the button is pressed, the web application does the following:

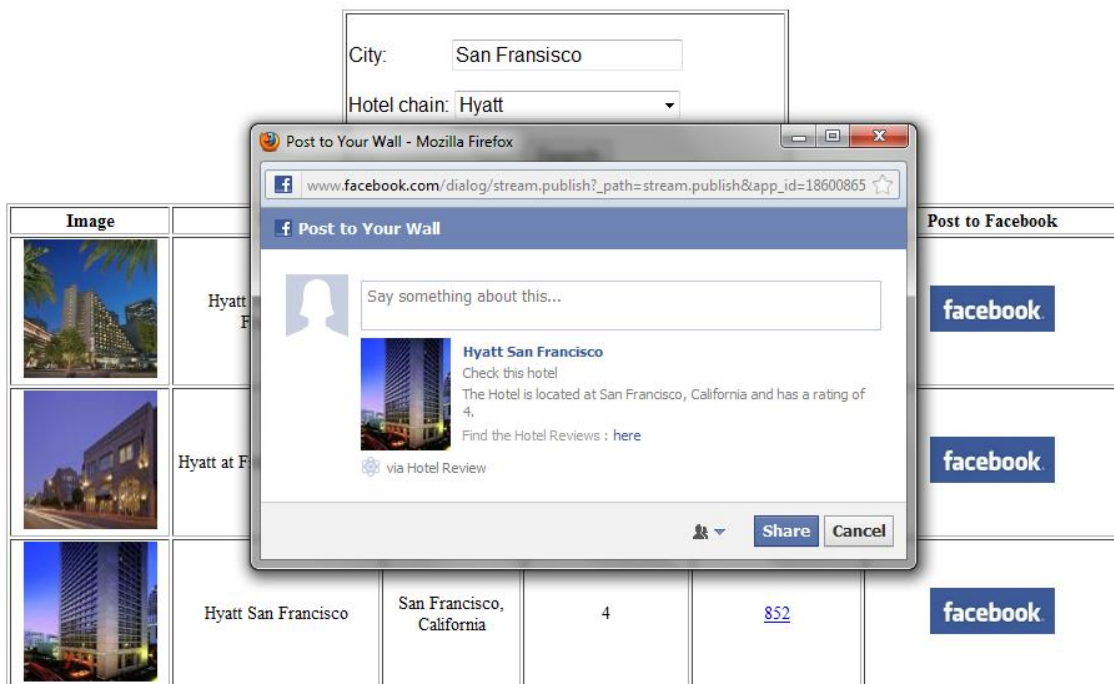
- a) Authorizes the user to Facebook (i.e. logs him/her in) using the application and user credentials if the user is not already logged in to Facebook;
- b) Posts a story feed having an attachment, which includes the hotel name, location, image, rating and link to the reviews to the user’s News Feed, as outlined in section 3-f.

The above two steps are performed using the Facebook Connect API, using the JavaScript Client Library for Facebook API, documented at:

<http://developers.facebook.com/blog/post/73/>



**Figure 2 – Facebook login**



**Figure 3 – Post to user's feed**

## 4. Implementation Hints

### 1. Step 1: Writing your JavaScript Program – set up Google Ajax transaction

The JavaScript invoked by the Search button click event should do all of the following:

- a. Assign the “callback” function;
- b. Assemble the `url` parameter of the GET as a reference to the Java Servlet to be invoked:

```
BASE_URL + "/servlet/ajax_hotel"
```

- c. Call the `XMLHttpRequest` method (see Ajax Slide 24) and create the request object.
- d. Prepare the GET `XMLHttpRequest` using the `setRequestHeader` method:

```
req.open("GET", url, true);  
req.onreadystatechange = myCallback;  
req.setRequestHeader("Connection", "Close");  
req.setRequestHeader("Method", "GET" + url + "HTTP/1.1");
```

### 2. Step 2: Writing your JavaScript Program – Execute Ajax Transaction

The JavaScript should finally invoke the `XMLHttpRequest send` method (see Ajax slide 24).

The “callback” function should check for completion of the transaction (request `readyState` equal to 4 and `status` equal to 200 (see AJAX slide 27 and JSON slide 5); use `eval()` and the `responseText` method to retrieve the resulting JSON data (see JSON slide 5), and displaying the returned hotel information properties to the “dynamic” area.

### 3. Step 3: Use the Java Servlet to respond to `XMLHttpRequest` and retrieve the nearest hotel information listings

The Java Servlet referred above as `/servlet/ajax_hotel` (see 1.c above) should be invoked using `doGet()`.

The Java Servlet should do all of the following:

- a. Initiating a connection with the Perl server side CGI, using the Apache server from Homework #6 (or PHP, Python), to retrieve the hotel listings using scraping code from Homework #6.

### 4. Step 4: Use the Java Servlet to retrieve the XML file content

You may have to use an XML parser (JAXP, for example). If you are hand coding using JAXP, the steps to retrieve the XML file content may be as follows:

Step a: Get the XML content based on the URL above in Step 3.a.

- You need to open a URL connection to get the file you want. To create a URL connection:

```
URL url = new URL(urlString);  
  
URLConnection urlConnection = url.openConnection();  
  
urlConnection.setAllowUserInteraction(false);  
  
InputStream urlStream = url.openStream();  
  
//read content
```

Step b: Parse the XML file using an XML parser

- Any XML parser can be used to parse the XML file. You can use methods like `getNodeName()` to access these elements. A good choice might be the JDOM library, which you get from:

<http://www.jdom.org/downloads/index.html>

### *5. Step 5: Use the Java Servlet to process the XML data*

As you parse the data, you will build an output string, converting the XML data into JSON format, as described in section 3.

Finally you will return the JSON as a single string to the calling JavaScript program. To easily create a JSON string, you might find useful the JSON-RPC library available at:

<http://mirrors.ibiblio.org/pub/mirrors/maven/com.metaparadigm/jars/json-rpc-1.0.jar>

The Java Servlet should handle exceptions such as `MalformedURLException` and `IOException`. The Java Servlet should also handle error responses sent from the Apache servlet and reply with an appropriate error, a JSON message to the initial JavaScript XMLHttpRequest. This way, the JavaScript callback function will be able to inform the user that an error has occurred.

### *6. Step 6: Writing your JavaScript Program – Post hotel details to Facebook*

Once the hotel properties are displayed in the dynamic area, the user should be able to click on the “Facebook” button for the corresponding hotel in the table.



## *7. Step 7: Writing your JavaScript Program – Authorize Facebook User*

Once the user clicks on the Facebook button, the program invokes the Facebook Connect API and authorizes the user. The recommended API to use is FB.Init, which are documented at:

<http://developers.facebook.com/docs/reference/javascript/FB.init/>

Also look at the code listed under “Loading” in the JavaScript SDK page at:

<https://developers.facebook.com/docs/reference/javascript/>

## *8. Step 8: Writing your JavaScript Program – Post Hotel information to Facebook News Feed*

There are several methods to post a message to the user’s feed page (the “wall”).

One such method uses the `Fb.ui()` API with the “stream.publish” method, documented at:

<http://developers.facebook.com/docs/reference/javascript/FB.ui/>

The “stream.publish” method is documented at:

<http://developers.facebook.com/docs/reference/rest/stream.publish/>

The “attachment” object of the stream is documented at:

<http://developers.facebook.com/docs/guides/attachments/>

See the “image” media type documentation.

Once the user is authorized, and an appropriate session token has been obtained, the text of the selected hotel is posted to the user’s news feed page (the “wall”). A subsequent posting will not require the user to log in again.

Additional information that may be useful to you is available in the Facebook for Websites web page at:

<https://developers.facebook.com/docs/guides/web/>

## **5. Prerequisites**

This homework requires the use of the following components:

1. A servlet-based web server, Tomcat 4.1.27. Instructions on how to load Tomcat 4.1.27 can be found here: <http://www->

- [scf.usc.edu/~csci571/2006Spring/tomcatinstall.html](http://scf.usc.edu/~csci571/2006Spring/tomcatinstall.html). A tar version of Tomcat 4.1.27 can be found here: <http://www-scf.usc.edu/~csci571/download/jakarta-tomcat-4.1.27.tar>.
- The Java Servlet library, which has functionality similar to Perl's LWP library, to perform HTTP transactions using methods such as `doGet()` or `doPost()` from Java.
  - A Java XML parser library. You may use the JDOM 1.0, an object model that uses XML parsers to build documents, available in the Download section of the class website. Additional information on JDOM is available at <http://www.jdom.org/>. You may also use JAXP, the Java API for XML Parsing, version 1.1, included in the Java JDK 1.4 (import `javax.xml.parsers.*`) and documented at: <http://java.sun.com/xml/jaxp/dist/1.1/docs/api/>. A good tutorial on JAXP is available at <http://www-106.ibm.com/developerworks/xml/library/x-jaxp1.html>.
  - You need to create a Facebook Platform application:  
To do that you will need to add the Facebook Developer application: go to <http://www.facebook.com/developers/>, and click **Create New App**. Once you've added the Facebook Developer application to your account, you can begin creating your application for Facebook. You should be getting an **API Key** and **Application Secret** (see Figure 4 below) that you will have to use with the JavaScript Client Library `FB.init` API.

The screenshot displays the Facebook App 'Hotel Review' settings page. At the top, there are buttons for 'Edit App' and '+ Create New App'. The main content is organized into several sections:

- Settings:** Includes an 'Edit Settings' link. Below it, a 'Summary' section lists:
  - App ID/API Key: [Redacted]
  - App Secret: [Redacted]
  - Site URL: <http://cs-server.usc.edu:37281/>
  - Deauthorize URL: <http://cs-server.usc.edu:37281/>
  - Contact Email: lakshmanisha@gmail.com
  - Support Email: lakshmanisha@gmail.com
- Open Graph:** Includes an 'Edit Open Graph' link. A message states: 'You have not added any actions, objects, or profile units. Get started using the Open Graph.'
- Roles:** Includes an 'Edit Roles' link. It shows a list of roles with one admin role assigned to a user profile picture.
- Insights:** Includes a 'See All' link. It displays two columns of data:
  - Users:**
    - 1 Daily New Users
    - 0 Daily Active Users
  - Sharing:**
    - 2 Daily Content Shared
    - 0 Feedback per Share

At the bottom of the page, there are links for 'About', 'Platform Policies', and 'Privacy Policy'.

**Figure 4 – Sample Facebook App Page**

## 6. Deployment Structure

To write your own Java Servlets program using Tomcat 4.1.27, you need to:

1. Successfully install Tomcat 4.1.27 on your machine.
2. Go to `$CATALINA_HOME/webapps/examples` directory.
3. Place the HTML, CSS and JavaScript (.js) files in the Tomcat servlets subdirectory.
4. Place your Java Servlets file (.java) in the `/WEB-INF/classes` folder. So the path of your Servlets file is [http://server\\_name:port/examples/servlet/your\\_servlet\\_name](http://server_name:port/examples/servlet/your_servlet_name)
5. Add appropriate sections to the `WEB-INF/web.xml` file, as in:

```
<servlet>
  <servlet-name>ajax_hotel</servlet-name>
  <display-name>AJAX Hotel</display-name>
  <servlet-class>AJAXHotel</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ajax_hotel</servlet-name>
  <url-pattern>/servlet/ajax_hotel</url-pattern>
</servlet-mapping>
```

6. To avoid `UTFDataFormatException` during file IO operation, you have to use JDK 1.3 or later for Tomcat. In the `.cshrc` file under your home directory, add the entries:

```
setenv JAVA_HOME /usr/j2se

setenv PATH /usr/j2se/bin:${PATH}
```

7. Before you issue a request to your Java Servlet file, you need to compile it. You might need a Java Servlet class to compile your code, so open the `.cshrc` file, and add the full path to the Tomcat file that implements the Servlet class APIs located in `../jakarta-tomcat-4.1.27/common/lib/servlet.jar` to your `CLASSPATH` variable, and use the variable with the `-classpath` switch of the `javac` compiler.
8. Then run `source .cshrc` and you are ready to compile your Java files.

## 7. Material You Need to Submit

On your course homework page, your link for this homework should go to a page that includes your JavaScript/HTML program (a page similar to the one depicted in the picture in section 3). You should submit all source code files including HTML (.html), Cascading Style Sheets (.CSS), JavaScript (.js), Java Servlets (.java) and a README file electronically to the csci571 account so that it can be graded and compared to all other students' code via the MOSS code comparison tool.

## **8. Partial Credit**

If you complete Step 1 through 6 as listed in Section 4, you will obtain 5 points. If you also complete Steps 7 and 8 (authorizing a user to Facebook and posting the story to the user's Facebook news feed), you will obtain the full 10 points for the exercise.