

Aplicații Web cu JSF și AJAX

Aplicațiile Web au nevoie de multe ori de actualizarea (periodică a) unor informații afișate în paginile dinamice folosite. Actualizarea poate fi lăsată în sarcina utilizatorului, care ar trebui să efectueze anumite acțiuni (apăsarea unor butoane), lucru care nu rezolvă întotdeauna problema (utilizatorul poate fi ocupat sau poate nu are o pregătire tehnică sau ...). Actualizarea poate fi făcută automat, de către aplicație, care periodic reîmprospătează conținutul paginilor. Nici această modalitate nu poate fi acceptată, utilizatorul putând fi nemulțumit de faptul că aplicația "decide" singură să schimbe conținutul afișat.

Indiferent de care din modalitățile de mai sus discutăm, există o deficiență majoră a lor: se împrospătează **întreaga pagină**, ceea ce impactează negativ în condițiile în care pagina este voluminoasă.

La modul general, pentru a actualiza informațiile dintr-o pagină sau dintr-o zonă a unei pagini se face o cerere HTTP către server, care răspunde cu o întreagă pagină nouă (eventual aceeași, actualizată).

O soluție elegantă a actualizării parțiale a unei pagini este **AJAX**, Asynchronous JavaScript and XML, o combinație de elemente și acțiuni de pe client și server. Astfel, anumite elemente HTML pregătesc terenul pentru acțiunile ce vor fi executate de secvențe JavaScript speciale. Rolul JavaScript este de a iniția o cerere HTTP (GET sau POST) și de a face modificările care se impun pe baza răspunsului primit de la componenta dedicată de pe server (pagină JSP, servlet).

Operațiile necesare pentru folosirea AJAX pot fi realizate manual sau apelând la una dintre bibliotecile specializate existente (Prototype, Script.aculo.us, jQuery). Detalii despre astfel de biblioteci pot fi găsite la http://en.wikipedia.org/wiki/Comparison_of_JavaScript_frameworks sau la http://ajaxpatterns.org/Java_Ajax_Frameworks sau prin căutare google.

1. AJAX clasic

Folosirea AJAX pur presupune următoarele etape:

- JavaScript, definirea unui obiect HTTP request.

```
1 // Get the browser-specific request object, either for
2 // Firefox, Safari, Opera, Mozilla, Netscape, IE 8, or IE 7 (top entry);
3 // or for Internet Explorer 5 and 6 (bottom entry).
4
5 function getRequestObject() {
6     if (window.XMLHttpRequest) {
7         return(new XMLHttpRequest());
8     } else if (window.ActiveXObject) {
9         return(new ActiveXObject("Microsoft.XMLHTTP"));
10    } else {
11        // Don't throw Error: this part is for very old browsers,
12        // and Error was implemented starting in JavaScript 1.5.
13        return(null);
14    }
15 }
```

- JavaScript, inițierea cererii HTTP:
 - Creare obiect **local** request.
 - Stabilire handler anonim pentru preluarea răspunsului (atributul *onreadystatechange* al cererii), printr-un mecanism similar cu pointerii la funcții din limbajul C.
 - Inițiere propriu-zisă cerere.
 - Trimitere (asincronă) date.

```

17 // Make an HTTP request to the given address.
18 // Display result in an alert box.
19
20 function ajaxAlert(address) {
21     var request = getRequestObject();
22     request.onreadystatechange =
23     function() { showResponseAlert(request); };
24     request.open("GET", address, true);
25     request.send(null);
26 }

```

- JavaScript, tratarea răspunsului primit.

```

28 // Put response text in alert box.
29
30 function showResponseAlert(request) {
31     if ((request.readyState == 4) && // raspuns complet
32         (request.status == 200)) { // fara erori HTTP
33         alert(request.responseText);
34     }
35 }

```

- HTML, folosire XHTML în loc de HTML 4.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">

```

- HTML, încărcare JavaScript.

```

5 <head><title>Ajax Basics: Part I</title>
6 <link rel="stylesheet"
7       href="./css/styles.css"
8       type="text/css"/>
9 <script src="./scripts/ajax-utils.js"
10        type="text/javascript"></script>
11 </head>

```

- HTML, cedare control pentru inițiere cerere HTTP.

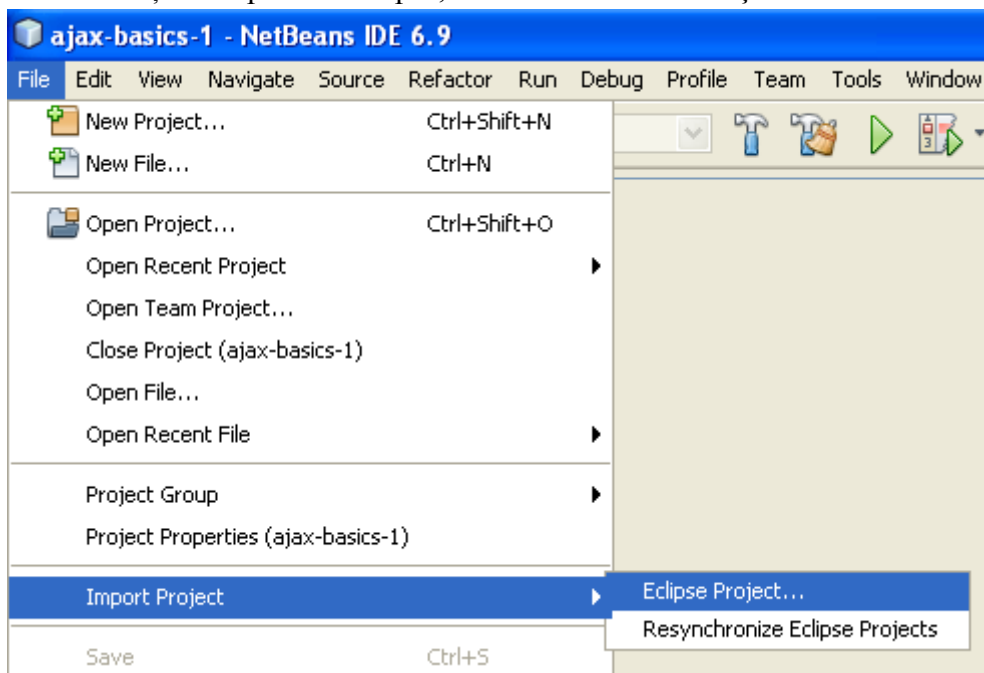
```

20 <input type="button" value="Show Server Time"
21        onclick='ajaxAlert("show-time.jsp")' />

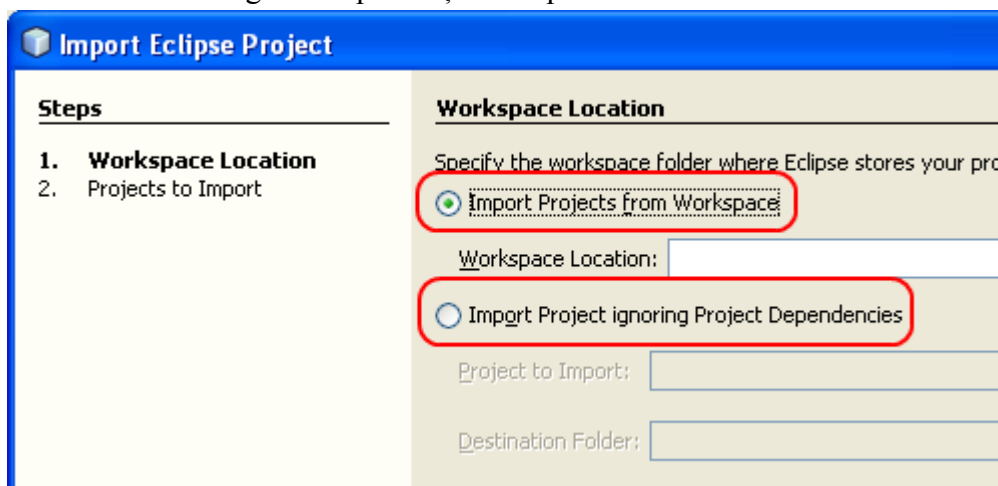
```

1.1. Exemplu AJAX clasic

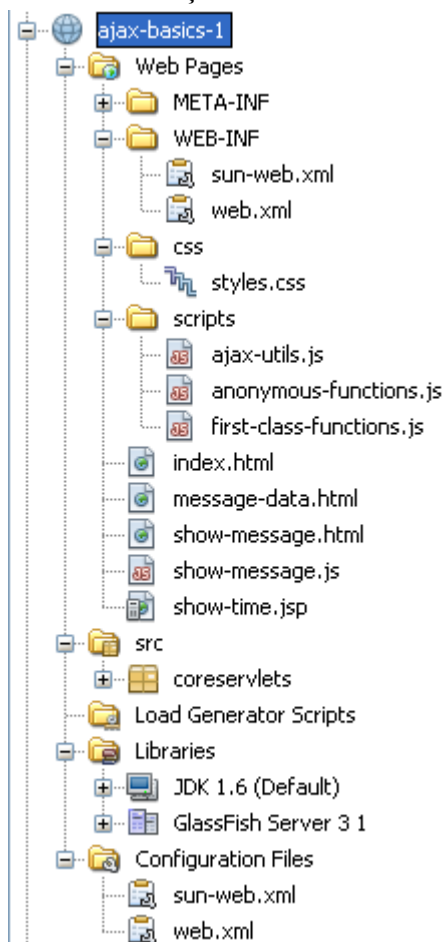
Preluăți arhiva cu surse de la adresa <http://courses.coreservlets.com/Course-Materials/ajax-code/ajax-basics-1.zip>. Arhiva conține un proiect Eclipse, care va trebui deschis și modificat folosind NetBeans.



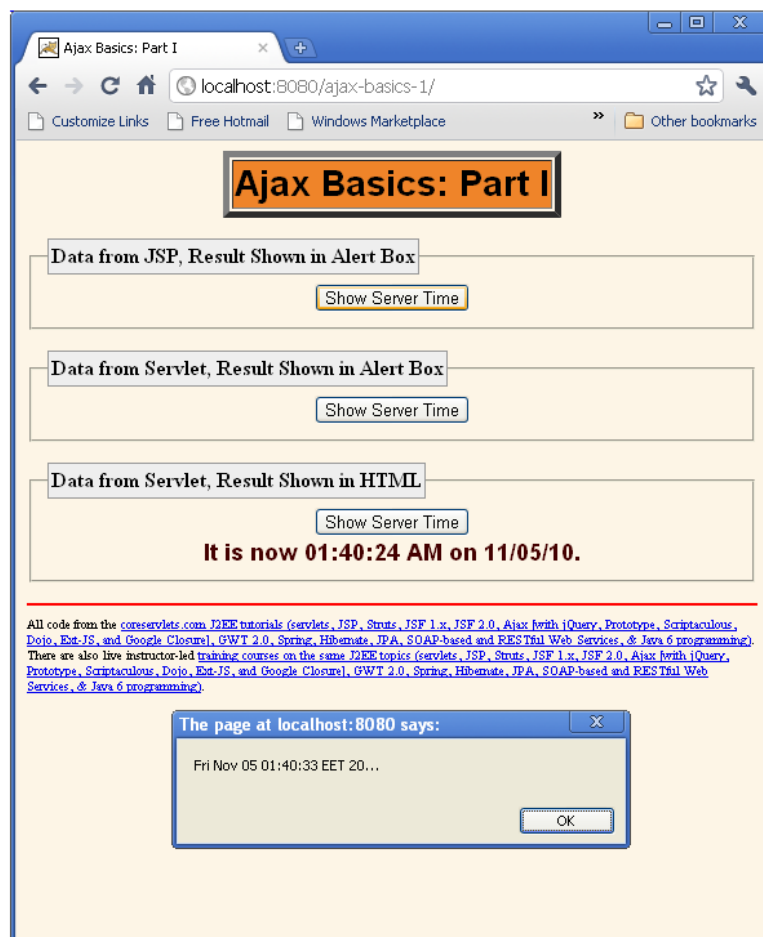
Importul unui proiect Eclipse în NetBeans este descris la adresa <http://netbeans.org/kb/docs/java/import-eclipse.html>. Se poate alege fie varianta cu preluare workspace Eclipse, fie varianta în care se ignoră dependențele din proiect.



Analizați organizarea proiectului, identificând fișierele în care intervine AJAX.



Rulați proiectul, observând funcționarea celor 3 zone care apar:

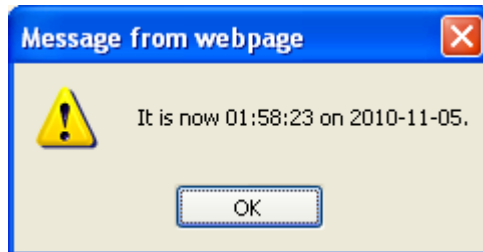


Modificați aplicația astfel încât fereastra Alert să apară la trecerea mouse-ului peste zona următoare:

Data from JSP, Result Shown in Alert Box

Verificați dacă este sau nu necesar rebuild pentru executarea variantei modificate.

Modificați aplicația astfel încât servletul să formateze răspunsul astfel:



Modificați aplicația astfel încât să se afișeze un mesaj ("Hello world!") în zona 3 la trecerea mouse-ului peste zona următoare:

Data from JSP, Result Shown in Alert Box

2. JSF și AJAX

JSF 2.0 integrează AJAX prin folosirea tag-ului **f:ajax**, cu următoarele atribute:

- **render**: elementele client care trebuie modificate;
- **execute**: elementele trimise spre server;
- **event**: evenimentele utilizator la care se răspunde;
- **onevent**: funcție JavaScript cu efecte laterale.

Avantajele folosirii AJAX integrat în JSF 2.0:

- Sunt permise actualizări ale elementelor JSF.
- Nu mai e necesară scrierea de cod JavaScript.
- Apelurile AJAX folosesc bean-uri.
- Nu mai e necesară scrierea de servleți.

Atributul *render* marchează elementul care va fi actualizat. În exemplul de mai jos, la apăsarea butonului se rulează acțiunea precizată pe server, se calculează noua valoare a elementului JSF precizat prin ID, se trimite răspunsul înapoi pe client, unde se înlocuiește valoarea curentă cu noua valoare.

```

22 <h:commandButton value="Show Number"
23     action="#{numberGenerator.randomize}">
24     <f:ajax render="numField1"/>
25 </h:commandButton><br/>
26 <h2><h:outputText value="#{numberGenerator.number}"
27     id="numField1"/></h2>

```

De obicei elementul care se reafixează este **h:outputText**. Dacă se reafixează mai multe elemente, ID-urile acestora vor fi separate prin spațiu atunci când se completează valoarea atributului *render*.

Există 4 valori speciale ce pot fi folosite aici:

- **@this**: folosit implicit, elementul care include **f:ajax**.
- **@form**: elementul **h:form** care include **f:ajax**, se folosește pentru a transfera mai multe câmpuri (elementele din formular pot să nu aibă un ID explicit).
- **@none**: nu trimite nimic, e util atunci când elementul implicat se modifică la fiecare reevaluare a sa.
- **@all**: pentru toate elementele JSF din pagina.

Atributul *execute* enumeră elementele de intrare care vor fi transmise către server pentru prelucrarea uzuală (validator, setter). Dacă sunt mai multe, se va folosi tot separatorul spațiu. De obicei se folosesc elementele **h:inputText** sau **h:selectOneMenu**.

Cele 4 valori speciale se folosesc de cele mai multe ori aici.

```

35 <h:inputText value="#{ numberGenerator.range}"
36         id="rangeField" ><br/>
37 <h:commandButton value="Show Number"
38         action="#{ numberGenerator.randomize}" >
39 <f:ajax execute="rangeField"
40         render="numField3" />
41 </h:commandButton><br/>
42 <h2><h:outputText value="#{ numberGenerator.number}"
43         id="numField3" /></h2>

```

Atributul *event* reține evenimentul la care se reacționează.

```

92 <h:inputText value="#{ temperatureConverter.fTemp}" >
93 <f:ajax event="keyup"
94         render="cField kField" />
95 </h:inputText><br/>

```

JSF 2.0 adaugă 2 evenimente noi, **action** (implicit pentru **h:commandButton** și **h:commandLink**) și **valueChange** (implicit pentru **h:inputText**, **h:inputSecret**, **h:inputTextarea**, butoane radio, checkbox și elemente de meniu de tip **h:selectOneMenu**).

Atributul *onevent* reține numele funcției ce se va rula înainte de eveniment, în timpul desfășurării acestuia sau după producerea lui.

```

114 <h:commandButton value="Show Current Balance"
115         action="#{ bankingBeanSlow.showBalance}" >
116 <f:ajax execute="@form"
117         render="ajaxMessage2"
118         onevent="showWorkingIndicator" />
119 </h:commandButton>

```

Funcția primește un singur parametru, cu o serie de proprietăți:

- status, poate fi "begin", "complete" sau "success".
- source, evenimentul declanșator.
- responseCode, responseText, responseXML.

Folosirea **f:ajax** impune folosirea `<h:head>` în loc de `<head>`, astfel încât structura unei pagini va fi următoarea:

```

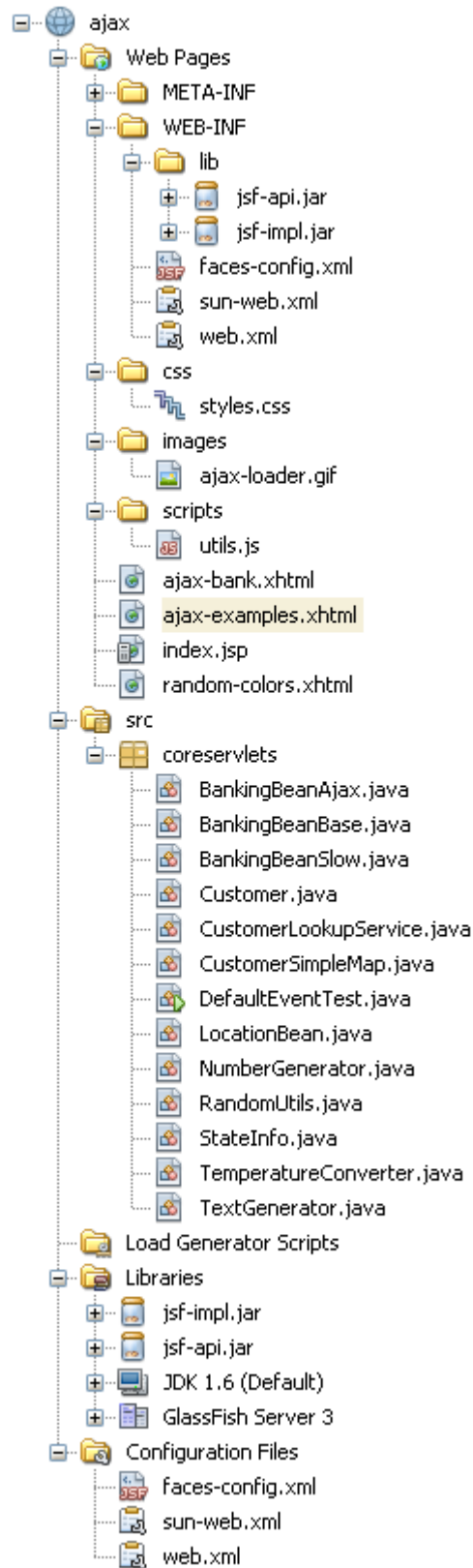
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:h="http://java.sun.com/jsf/html"
5       xmlns:f="http://java.sun.com/jsf/core">
6 <h:head><title>JSF 2.0: Ajax Support</title>
7 <.../>
9 <.../>
11 </h:head>
12 <h:body>
13 <.../>
145 </h:body>
146 </html>

```

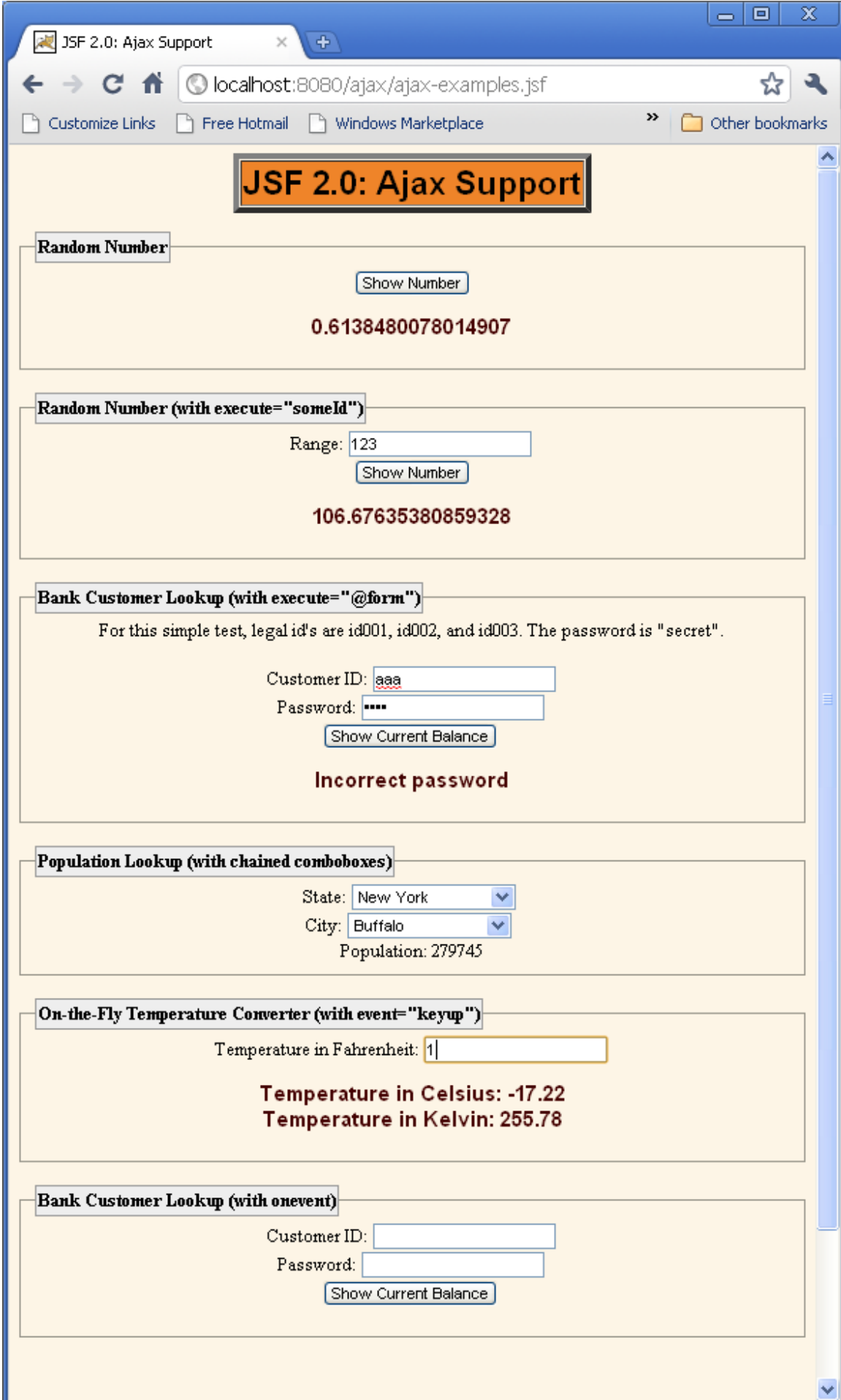
2.1. Exemplu AJAX și JSF

Preluăți arhiva cu surse de la adresa <http://www.coreservlets.com/JSF-Tutorial/jsf2/code/ajax.zip>. Arhiva conține un proiect Eclipse, care va trebui deschis și modificat folosind NetBeans.

Analizați organizarea proiectului, identificând fișierele în care intervine AJAX.



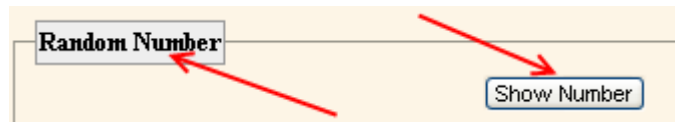
Rulați proiectul, observând funcționarea celor 6 zone care apar:



The screenshot shows a web browser window with the title "JSF 2.0: Ajax Support" and the URL "localhost:8080/ajax/ajax-examples.jsf". The page contains six distinct Ajax examples, each in a separate box:

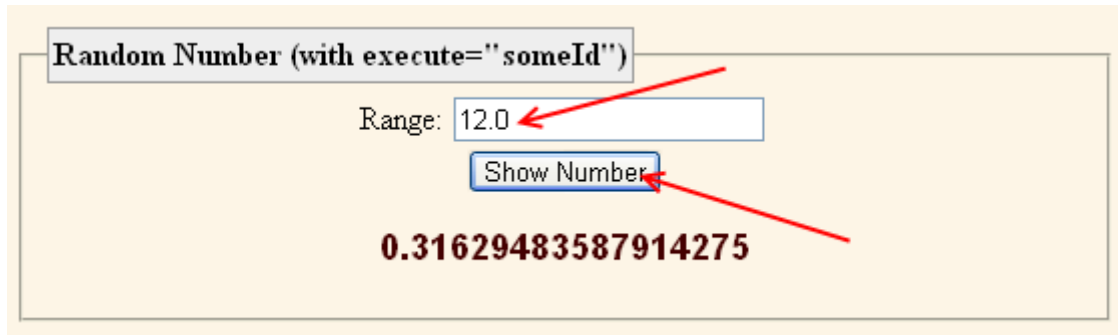
- Random Number:** A "Show Number" button is clicked, resulting in the display of the number "0.6138480078014907".
- Random Number (with execute="someId"):** A "Range" input field is set to "123", and the "Show Number" button is clicked, resulting in the display of the number "106.67635380859328".
- Bank Customer Lookup (with execute="@form"):** The "Customer ID" field contains "aaa" and the "Password" field contains "****". The "Show Current Balance" button is clicked, resulting in the message "Incorrect password".
- Population Lookup (with chained comboboxes):** The "State" dropdown is set to "New York" and the "City" dropdown is set to "Buffalo". The "Population" field displays "279745".
- On-the-Fly Temperature Converter (with event="keyup"):** The "Temperature in Fahrenheit" input field contains "1". The "Temperature in Celsius" field displays "-17.22" and the "Temperature in Kelvin" field displays "255.78".
- Bank Customer Lookup (with onevent):** The "Customer ID" and "Password" fields are empty. The "Show Current Balance" button is visible.

Modificați aplicația astfel încât numărul aleator să se actualizeze la trecerea mouse-ului peste elementele indicate din zona următoare:



Verificați dacă este sau nu necesar rebuild pentru executarea variantei modificate.

Modificați aplicația astfel încât numărul aleator să se actualizeze pe măsură ce se completează valoarea câmpului range sau când butonul este selectat (primește focus).



Modificați aplicația astfel încât verificarea câmpurilor *customerID* și *password* să se facă pe măsură ce se introduc date în aceste câmpuri.

3. Resurse

Resurse AJAX utile:

- <http://courses.coreservlets.com/Course-Materials/ajax-basics.html>
- <http://courses.coreservlets.com/Course-Materials/pdf/ajax/Ajax-Basics-1.pdf>
- <http://courses.coreservlets.com/Course-Materials/pdf/ajax/Ajax-Basics-2.pdf>
- <http://www.coreservlets.com/JSF-Tutorial/jsf2/#Ajax>
- <http://courses.coreservlets.com/Course-Materials/pdf/jsf/jsf2/JSF2-Ajax.pdf>
- <http://courses.coreservlets.com/Course-Materials/ajax.html>

Resurse JSF utile:

- <http://download.oracle.com/javaee/6/javaxserverfaces/2.0/docs/pdlldocs/facelets/h/tld-summary.html>
- <http://www.coreservlets.com/JSF-Tutorial/jsf1/>
- <http://www.coreservlets.com/JSF-Tutorial/jsf2/>
- <https://javaeetutorial.dev.java.net/servlets/ProjectDocumentList>
- <http://download.oracle.com/javaee/6/javaxserverfaces/2.0/docs/pdlldocs/facelets/>
- <http://netbeans.org/kb/docs/web/jsf20-intro.html>