JSP Unified Expression Language

By:

Prof. Tushar Gohil

Asst. Prof.

IT Deptt.

SCET, Surat.

Mike

• Hard code developer
• Handles all business logic and backend matters
• Expert in Java, Database, XML etc.

Ernie

• Jack of all trades
• Not an expert in anything, but will eventually get the job done….

Philippe

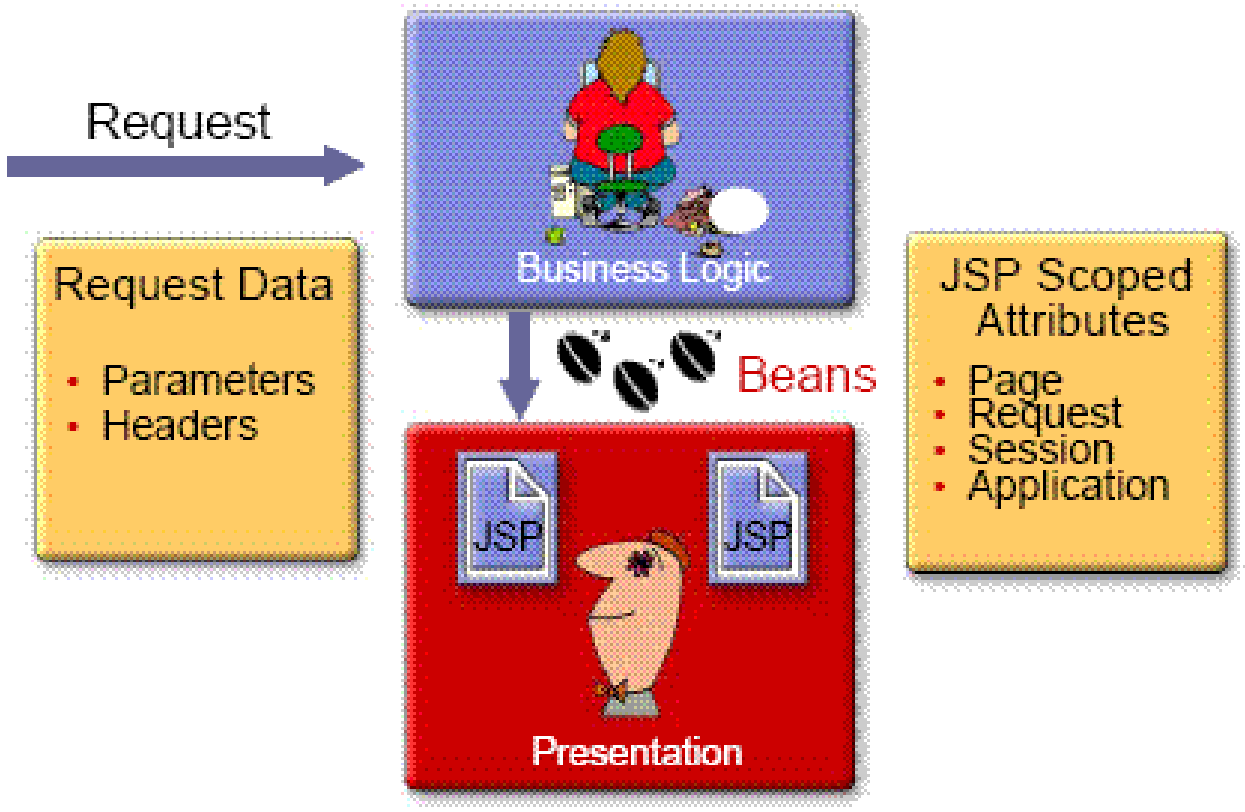
• Web site designer
• Knows how to make a Website look really cool !
• HTML / JavaScript expert

EL Purpose

- Java as the scripting language in JSP scares many people (e.g. Philippe)

- Can we simplify ?
  - Expression Language (EL)
    - A language adapted for the Web Developer

EL Benefits


Request
Request Data
• Parameters
• Headers
Business Logic
Beans
Presentation
JSP
JSP
JSP Scoped Attributes
• Page
• Request
• Session
• Application

JSP Before EL …

1. Must Declare

2. Must Know Type

```
<%
      Person p = (Person) request.getAttribute(“person”)
%>
……….
Person Name:  <%= p.getName() %>
………
<%  if (p.getAddress( ).equals(“defence”) ) {   %>
   …….
<% } %>
```

3. Awkward Syntax

4. Knowledge of Scripting Language required even for simple manipulations

JSP After EL …

1. Direct access

2. Easier syntax

```
Person Name:  $ { p.name }
…
<c:if  test = “$ {p.address == param.add }” >
   ${ p.name }
</c:if>
```

3. All app data easily accessible

4. Better adapted expression language

Unified EL overview

- The unified expression language allows page authors to use simple expressions to perform the following tasks:
  - Dynamically read application data stored in JavaBeans components, various data structures, and implicit objects
  - Dynamically write data, such as user input into forms, to JavaBeans components
  - Invoke arbitrary static and public methods
  - Dynamically perform arithmetic operations

EL Syntax

- Format

  ${ validExpression }

- Valid Expressions
  - Literals
  - Operators
  - Variables (object references)
  - Implicit call to function using property name

Classifying EL Expressions

- EL Expressions can be categorized into the following types:
  - **Immediate and Deferred expressions**
  - **Value expressions**
  - **Method expressions**

Immediate Expressions

- **Immediate evaluation** means that the JSP engine evaluates the expression and returns the result immediately when the page is first rendered.
- Those expressions that are evaluated immediately use the ${   } syntax
- These expressions can only be used within template text or as the value of a JSP tag attribute that can accept runtime value.

- The following code shows example immediate evaluation expression

```
<fmt:formatNumber value="${sessionScope.cart.total}"/>
```

Immediate evaluation expressions are always read-only value expressions. The expression shown above can only get the total price from the cart bean; it cannot set the total price.

Deferred Expressions

- **Deferred evaluation** means that the technology using the expression language can employ its own machinery to evaluate the expression sometime later during the page’s life cycle, whenever it is appropriate to do so.
- Those expressions that are evaluated deferred use the #{ } syntax
- The following example shows a JavaServer Faces inputText tag, which represents a text field component into which a user enters a value.

  <h:inputText id="name" value="#{customer.name}" />
- For an initial request of the page containing this tag, the JavaServer Faces implementation evaluates the #{customer.name} expression during the render response phase of the life cycle. During this phase, the expression merely accesses the value of name from the customer bean, as is done in immediate evaluation.
- For a postback, the JavaServer Faces implementation evaluates the expression at different phases of the life cycle, during which the value is retrieved from the request, validated, and propagated to the customer bean.

Value Expressions

- Value expressions can either yield a value or set a value.
- They are categorized into rvalue and lvalue expressions.
- **Rvalue expressions** are those that can read data, but cannot write it.
- **Lvalue expressions** can both read and write data.

- All expressions that are evaluated immediately use the ${} delimiters and are always rvalue expressions.
- Expressions whose evaluation can be deferred use the #{} delimiters and can act as both rvalue and lvalue expressions.

Value Expressions

- Consider these two value expressions:

  ${customer.name}

  #{customer.name}
- The former uses immediate evaluation syntax, whereas the latter uses deferred evaluation syntax. The first expression accesses the name property, gets its value, and the value is added to the response and rendered on the page. The same thing can happen with the second expression. However, the tag handler can defer the evaluation of this expression to a later time in the page life cycle, if the technology using this tag allows it.
- In the case of JavaServer Faces technology, the latter tag's expression is evaluated immediately during an initial request for the page. In this case, this expression acts as an rvalue expression. During a postback, this expression can be used to set the value of the name property with user input. In this situation, the expression acts as an lvalue expression.

Value Expressions

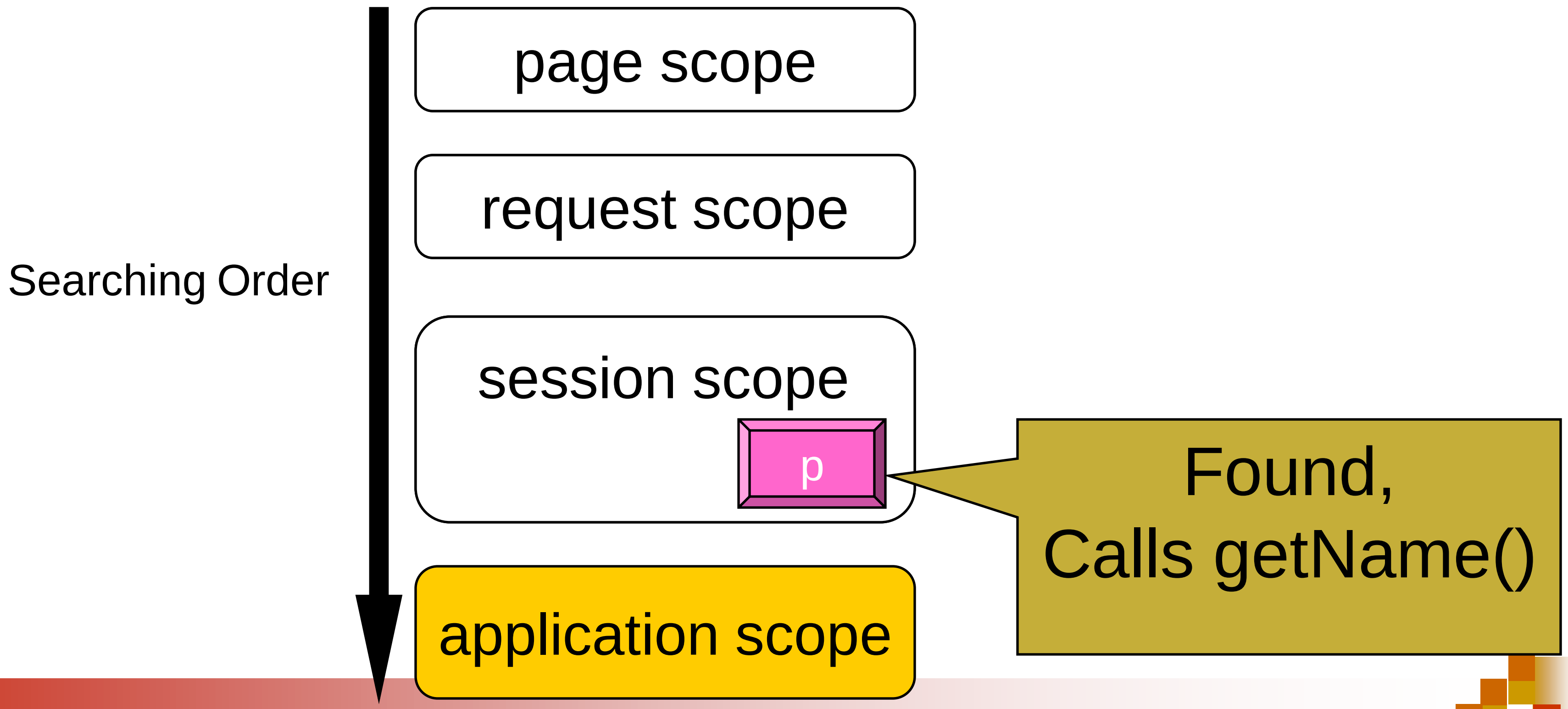
- Both rvalue and lvalue expressions can refer to the following objects and their properties or attributes:
  - JavaBeans components
  - Collections
  - Java SE enumerated types
  - Implicit objects

To refer to these objects, you write an expression using a variable that is the name of the object. The following expression references a managed bean called customer:

${customer}

Here while evaluating the expression ${customer}, the container will
look for customer in the page, request, session, and application scopes and will return its value. If customer is not found, a null value is returned.

- Search for a scoped variable, E.g. **${** p.name **}**



Value Expressions

- To reference an enum constant with an expression, use a String literal.

  For example, consider this Enum class:

  public enum Suit {hearts, spades, diamonds, clubs}

  ${Suit.hearts}

- To refer to properties of a bean , items of a collection, or attributes of an implicit object, you use the . or [] notation.

  To reference the name property of the customer bean,

  use either the expression ${customer.name}

  or

  the expression ${customer["name"]}.

  You can also combine the [] and . notations, as shown here:

  ${customer.address["street"]}

Method Expressions

- They are used to call public methods.
- Such expressions are usually deferred.
- In JSF, a component tag represents a component on a page. The component tag uses method expressions to invoke methods that perform some processing for the component. These methods are necessary for handling events that the components generate and for validating component data, For example

```
<h:form>
 <h:inputText
        id="name"
        value="#{customer.name}"
        validator="#{customer.validateName}"/>
<h:commandButton
        id="submit" action="#{customer.submit}" />
 </h:form>
```

The inputText tag displays as a text field. The validator attribute of this inputText tag references a method, called validateName, in the bean, called customer.

EL literals

Literals	Literal Values
Boolean	true or false
Integer	Similar to Java e.g. 243, -9642
Floating Point	Similar to Java e.g. 54.67, 1.83
String	Any string delimited by single or double quote e.g. “hello” , ‘hello’
Null	null

EL literals (cont.)

- Examples
  - ${ false } <%-- evaluate to false --%>
  - ${ 8*3 } <%-- evaluate to 24 --%>

EL Operators

Type	Operator
Arithmetic	+ - * / (div) % (mod)
Grouping	( )
Logical	&& (and) || (or) ! (not)
Relational	== (eq) != (ne) < (lt) > (gt) <= (le) >= (ge)
Empty	prefix operation to determine value is null or empty, returns boolean value
Conditional	? :

EL Operators (cont.)

- Examples
  - ${ (6*5) + 5 } <%-- evaluate to 35 --%>
  - ${ (x >= min) && (x <= max) }
  - ${ empty name }
    - Returns true if name is
      - Empty string (“”),
      - Null etc.

EL Identifiers

- **Identifiers**
  - Represents the name of the object
  - Objects stored in JSP scopes (page, request, session, application) referred as scoped variables
- EL has 11 reserved identifiers, corresponding to 11 implicit objects
- All other identifiers assumed to refer to scoped variables

EL Identifiers
Implicit Objects [1]

Category	Implicit Object	Description
JSP	pageContext	used to access JSP implicit objects
Scopes	pageScope	A *Map* associating names & values of page scoped attributes
	requestScope	A *Map* associating names & values of request scoped attributes
	sessionScope	A *Map* associating names & values of session scoped attributes
	applicationScope	A *Map* associating names & values of page scoped attributes

EL Identifiers (cont.)

Implicit Objects [2]

Category	Operator	Description
Request Parameters	Param	Maps a request parameter name to a single String parameter value
	paramValues	Maps a request parameter name to an array of values
Request Headers	header	Maps a request header name to a single header value
	headerValues	Maps a request header name to an array of values

EL Identifiers (cont.)

Implicit Objects [3]

Category	Operator	Description
Cookies	cookie	A *Map* storing the cookies accompanying the request by name
Initialization parameters	initParam	A *Map* storing the context initialization parameters of the web application by name

EL Identifiers (cont.)

Implicit Objects

- Examples
  - ${ pageContext.response }
    - evaluates to response implicit object of JSP
  - ${ param.name }
    - Equivalent to **request.getParameter("name")**
  - ${ cookie.name.value }
    - Returns the value of the first cookie with the given name
    - Equivalent to

```
if (cookie.getName().equals("name") {
    String val = cookie.getValue();
}
```

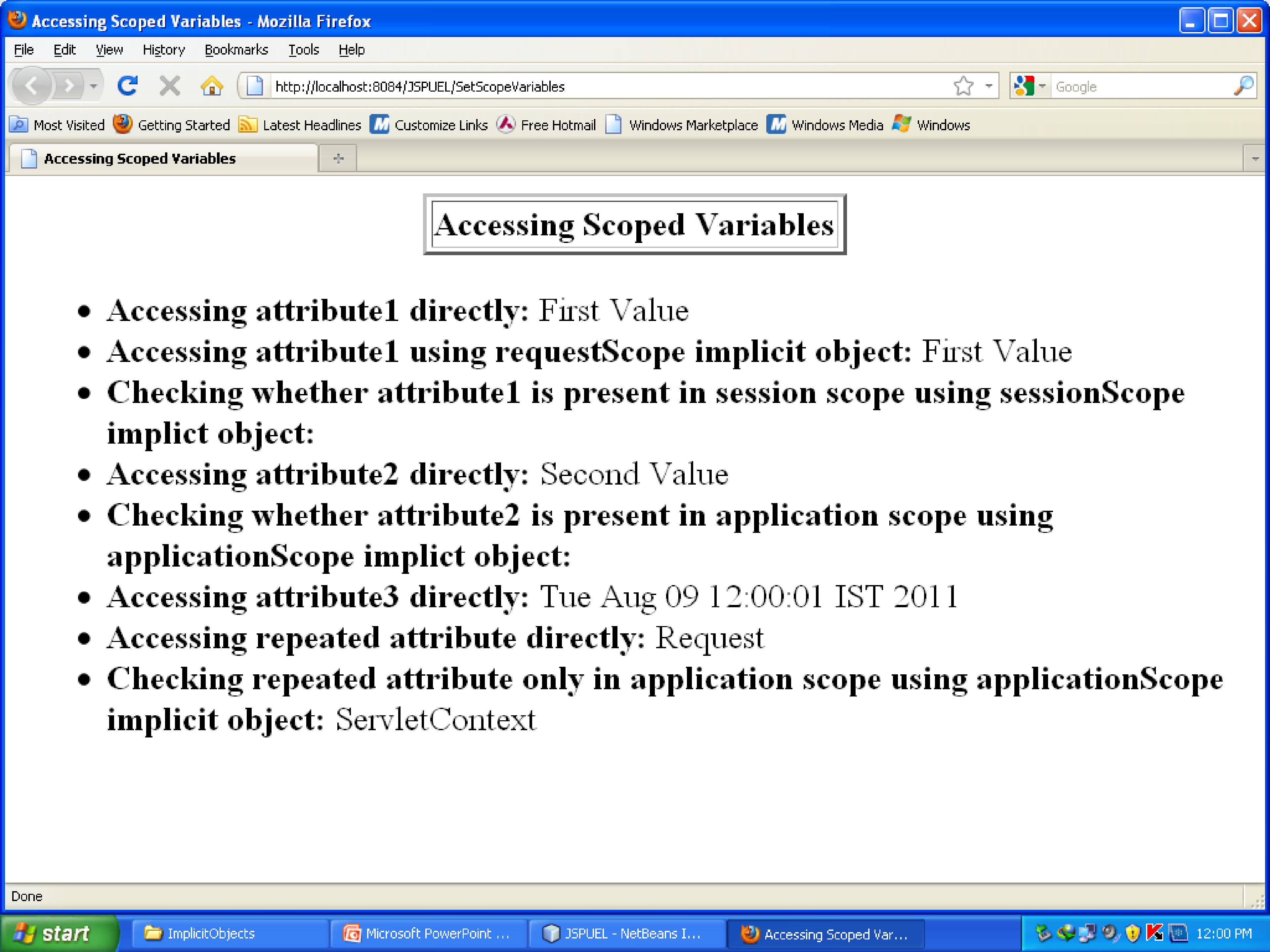
Using EL Scope Objects – SetScopeVariables.java

```
import java.io.*; import javax.servlet.*; import javax.servlet.http.*;
public class SetScopeVariables extends HttpServlet
{
public void doGet(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException {
req.setAttribute("attribute1", "First Value");
HttpSession session = req.getSession();
session.setAttribute("attribute2", "Second Value");
ServletContext application = getServletContext();
application.setAttribute("attribute3",new java.util.Date());
req.setAttribute("repeated", "Request");
session.setAttribute("repeated", "Session");
application.setAttribute("repeated", "ServletContext");
RequestDispatcher dispatcher =
req.getRequestDispatcher("GetScopeVariables.jsp");
dispatcher.forward(req, res);
}
}
```

Using EL Scope Objects – GetScopeVariables.jsp

```
<HTML> <HEAD> <TITLE>Accessing Scoped Variables</TITLE> </HEAD>
<BODY>
<TABLE BORDER=2 ALIGN="CENTER“> <TR><TH> Accessing Scoped Variables
</TABLE>
<UL>
<LI><B>Accessing attribute1 directly:</B> ${attribute1}
<LI><B>Accessing attribute1 using requestScope implicit object:</B>
${requestScope.attribute1}
<LI><B>Checking whether attribute1 is present in session scope using
sessionScope implict object:</B>${sessionScope.attribute1}
<LI><B>Accessing attribute2 directly:</B> ${attribute2}
<LI><B>Checking whether attribute2 is present in application scope using
applicationScope implict object:</B> ${applicationScope.attribute2}
<LI><B>Accessing attribute3 directly:</B> ${attribute3}
<LI><B>Accessing repeated attribute directly:</B> ${repeated}
<LI><B>Checking repeated attribute only in application scope using
applicationScope implicit object:</B> ${applicationScope.repeated}
</UL>
</BODY>
</HTML>
```

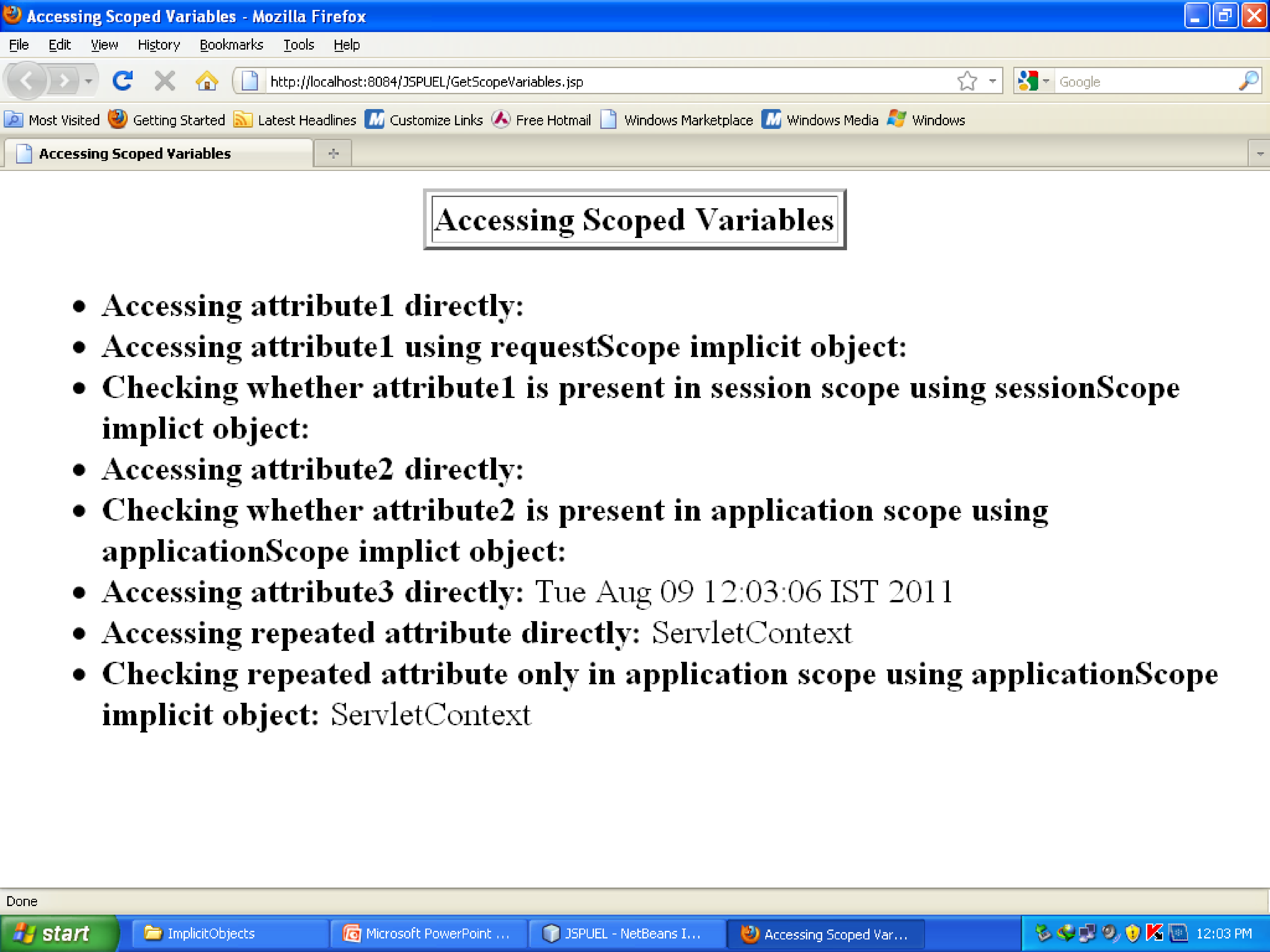
Now Run the SetScopeVariables Servlet


Accessing Scoped Variables - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8084/JSPUEL/SetScopeVariables
Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Accessing Scoped Variables
Done
start ImplicitObjects Microsoft PowerPoint ... JSPUEL - NetBeans I... Accessing Scoped Var... 12:00 PM

Accessing Scoped Variables

- **Accessing attribute1 directly:** First Value
- **Accessing attribute1 using requestScope implicit object:** First Value
- **Checking whether attribute1 is present in session scope using sessionScope implict object:**
- **Accessing attribute2 directly:** Second Value
- **Checking whether attribute2 is present in application scope using applicationScope implict object:**
- **Accessing attribute3 directly:** Tue Aug 09 12:00:01 IST 2011
- **Accessing repeated attribute directly:** Request
- **Checking repeated attribute only in application scope using applicationScope implicit object:** ServletContext

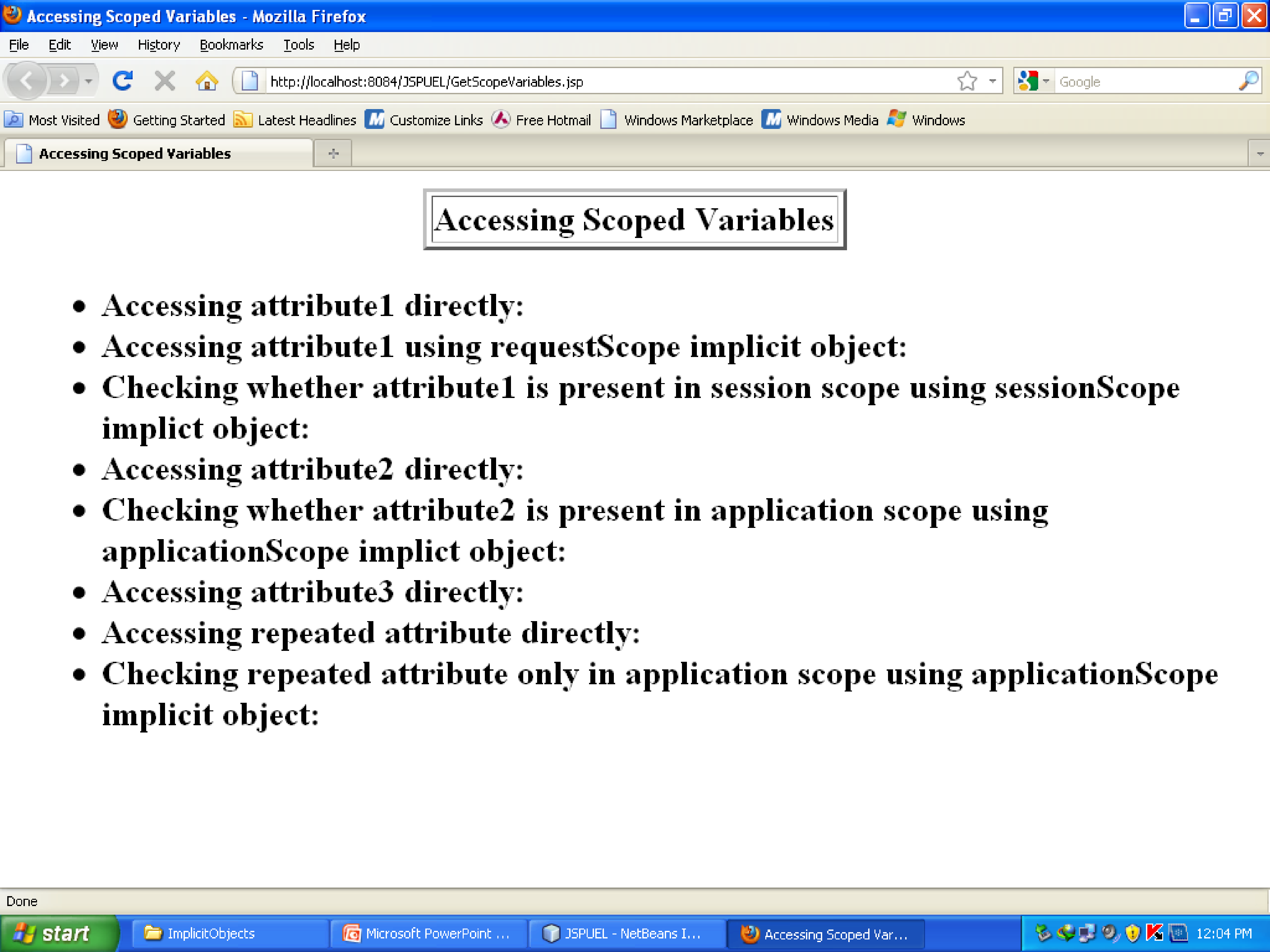
Now Run the GetScopeVariable.jsp Directly



Accessing Scoped Variables

- **Accessing attribute1 directly:**
- **Accessing attribute1 using requestScope implicit object:**
- **Checking whether attribute1 is present in session scope using sessionScope implict object:**
- **Accessing attribute2 directly:**
- **Checking whether attribute2 is present in application scope using applicationScope implict object:**
- **Accessing attribute3 directly:** Tue Aug 09 12:03:06 IST 2011
- **Accessing repeated attribute directly:** ServletContext
- **Checking repeated attribute only in application scope using applicationScope implicit object:** ServletContext

Now Restart the Tomcat and then Run the GetScopeVariable.jsp Directly

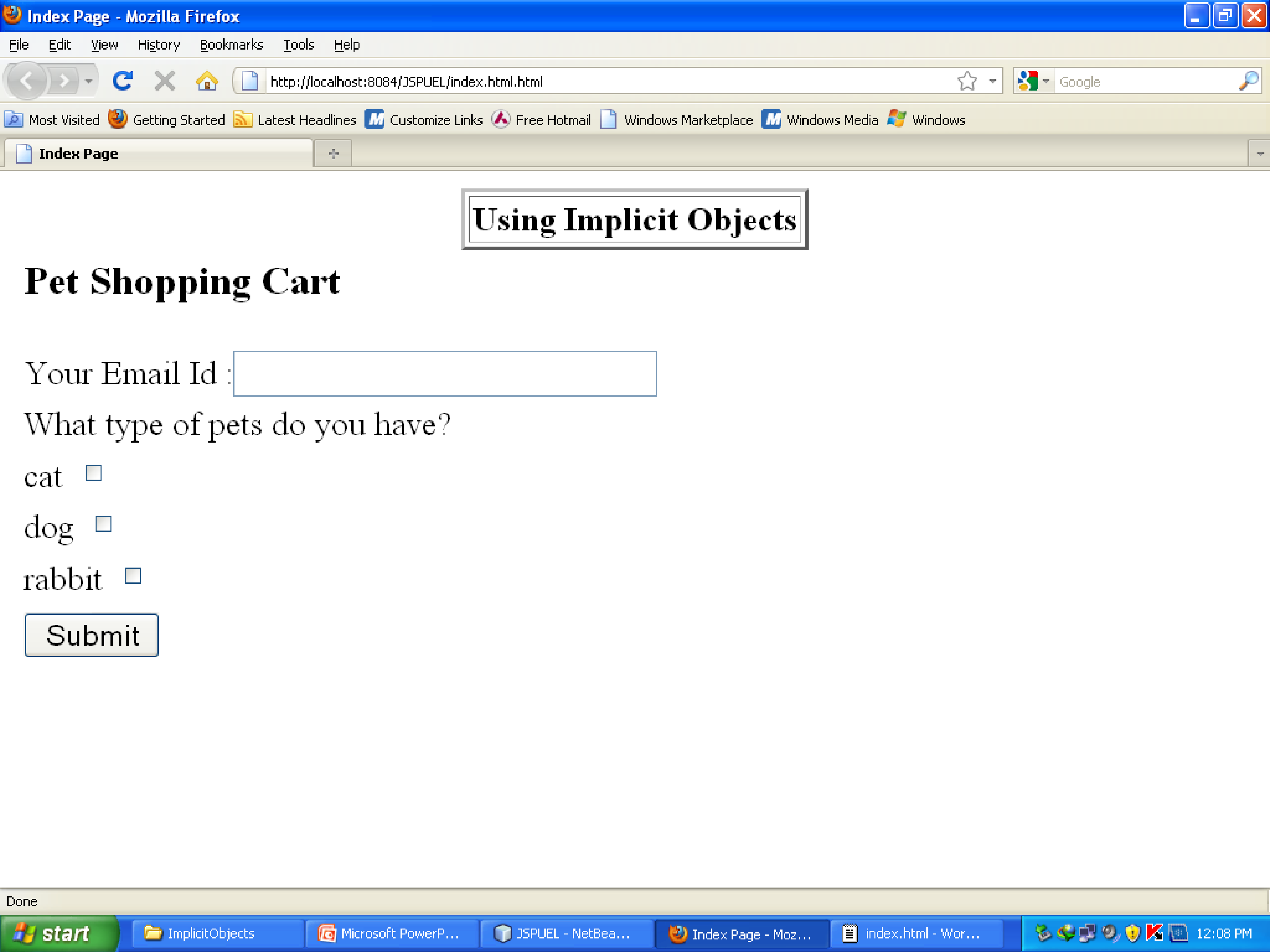

Accessing Scoped Variables - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8084/JSPUEL/GetScopeVariables.jsp
Google
Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Accessing Scoped Variables
Done
start ImplicitObjects Microsoft PowerPoint ... JSPUEL - NetBeans I... Accessing Scoped Var...
12:04 PM

Accessing Scoped Variables

- Accessing attribute1 directly:
- Accessing attribute1 using requestScope implicit object:
- Checking whether attribute1 is present in session scope using sessionScope implict object:
- Accessing attribute2 directly:
- Checking whether attribute2 is present in application scope using applicationScope implict object:
- Accessing attribute3 directly:
- Accessing repeated attribute directly:
- Checking repeated attribute only in application scope using applicationScope implicit object:

Using EL implicit Objects – index.html

```
<html> <head> <title>Index Page</title> </head> <body>
<form action="impobject2.jsp" method="get">
<table BORDER=2 ALIGN="CENTER"> <TR><Th> Using Implicit Objects </th></TR>  </table>
<table>  <tr><td colspan=2><h3>Pet Shopping Cart</h3></td></tr> <tr><td>
Your Email Id :<input type="text”name=email size="30"/> </td></tr>
<tr><td> What type of pets do you have? </td></tr>
<tr><td> cat <input type="checkbox" name="pettype“ value="cat" /> </td></tr>
<tr><td> dog <input type="checkbox" name="pettype“ value="dog" /> </td></tr>
<tr><td> rabbit <input type="checkbox“ name="pettype" value="rabbit"/> </td></tr>
</table>
<table>
<tr><td>
<input type=submit value="Submit">
</td></tr>
</table>
</form>
</body>
</html>
```


Index Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8084/JSPUEL/index.html.html
Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Index Page
Done
start ImplicitObjects Microsoft PowerP... JSPUEL - NetBea... Index Page - Moz... index.html - Wor... 12:08 PM

Using Implicit Objects

Pet Shopping Cart

Your Email Id :

What type of pets do you have?

cat ☐

dog ☐

rabbit ☐

Submit

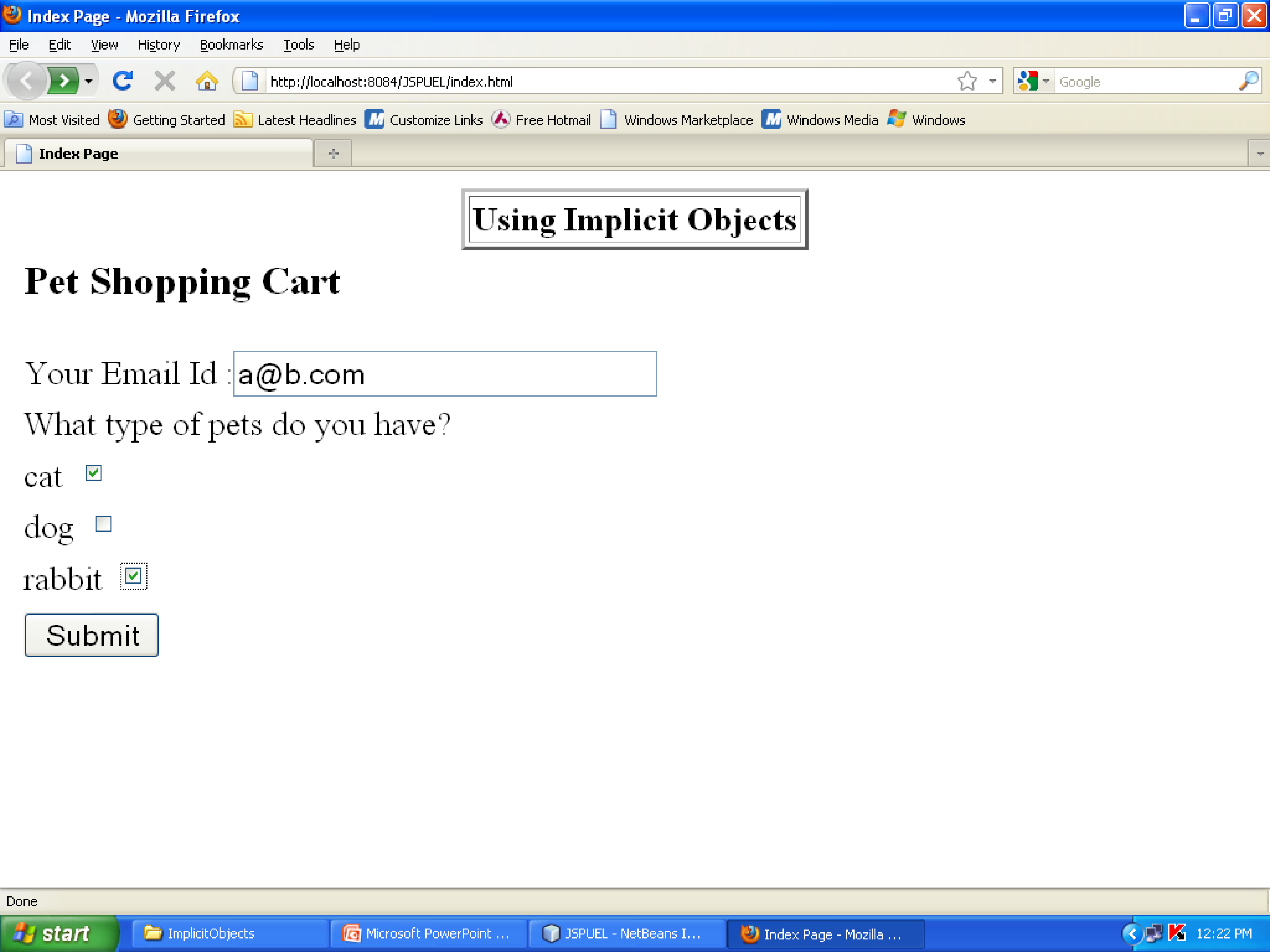
Using EL implicit Objects – impobject2.jsp

```
<HTML> <HEAD> <TITLE>Using Implicit Objects</TITLE> </HEAD>
<BODY> <TABLE BORDER=2 ALIGN="CENTER"> <TR><TH>
Using Implicit Objects </TH></TR>
</TABLE> <P>
<UL>
<LI><B>Your Email ID is :</B> ${param.email}

<LI><B>The First Pet You have Checked is :</B> ${paramValues.pettype['0']}
<LI><B>The Second Pet You have Checked is : :</B> ${paramValues.pettype['1']}
<LI><B>The Third Pet You have Checked is : :</B> ${paramValues.pettype['2']}

<LI><B>User-Agent Header:</B> ${header["User-Agent"]}
<LI><B>JSESSIONID Cookie Value:</B>
${cookie.JSESSIONID.value}
<LI><B>Server:</B> ${pageContext.servletContext.serverInfo}
</UL>
</BODY> </HTML>
```

Now Run the index.html First and Give the input and Click on Submit


Index Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8084/JSPUEL/index.html
Google
Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Index Page
Done
start ImplicitObjects Microsoft PowerPoint ... JSPUEL - NetBeans I... Index Page - Mozilla ... 12:22 PM

Using Implicit Objects

Pet Shopping Cart

Your Email Id : a@b.com

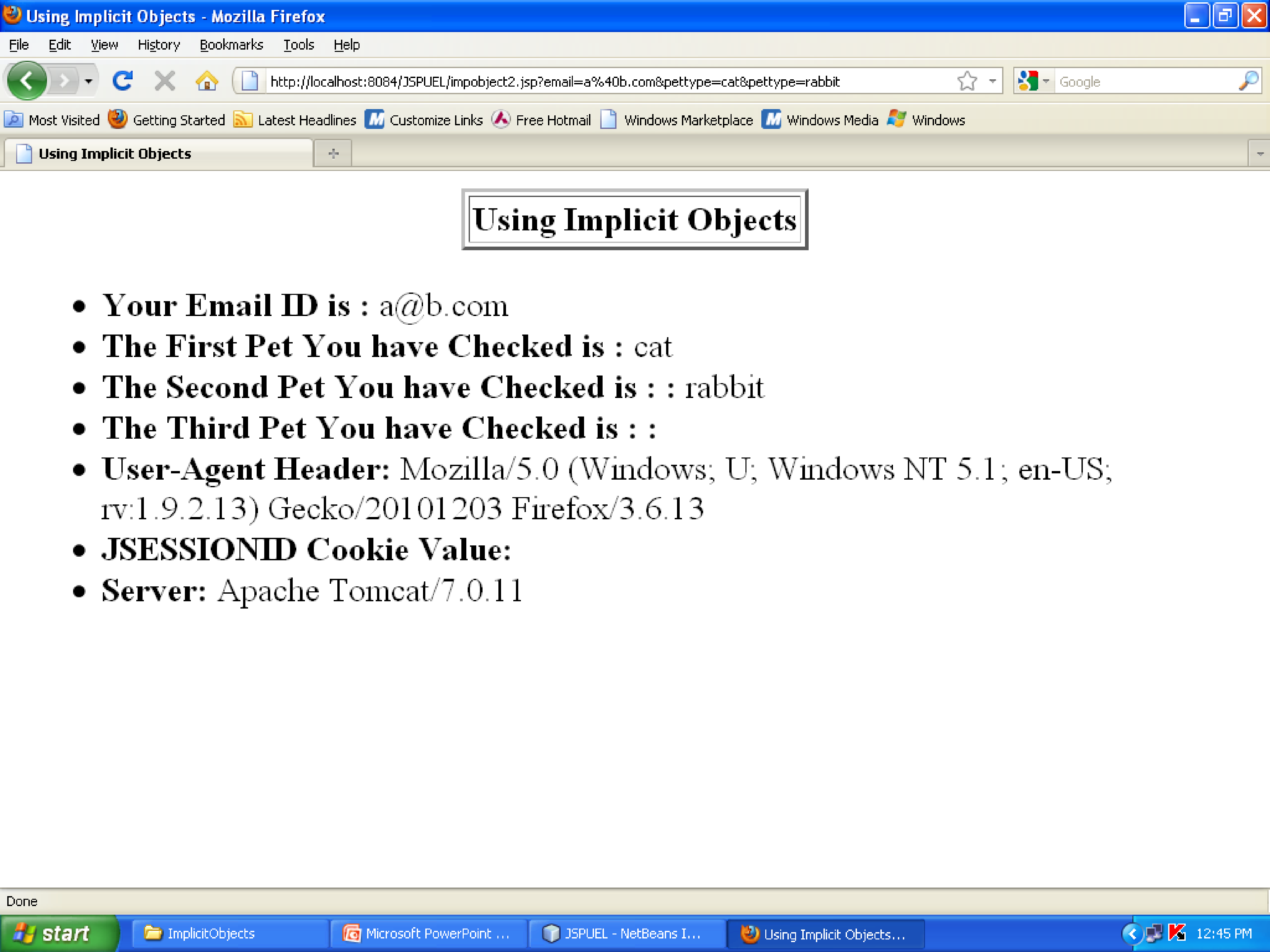
What type of pets do you have?

cat [✓]

dog [ ]

rabbit [✓]

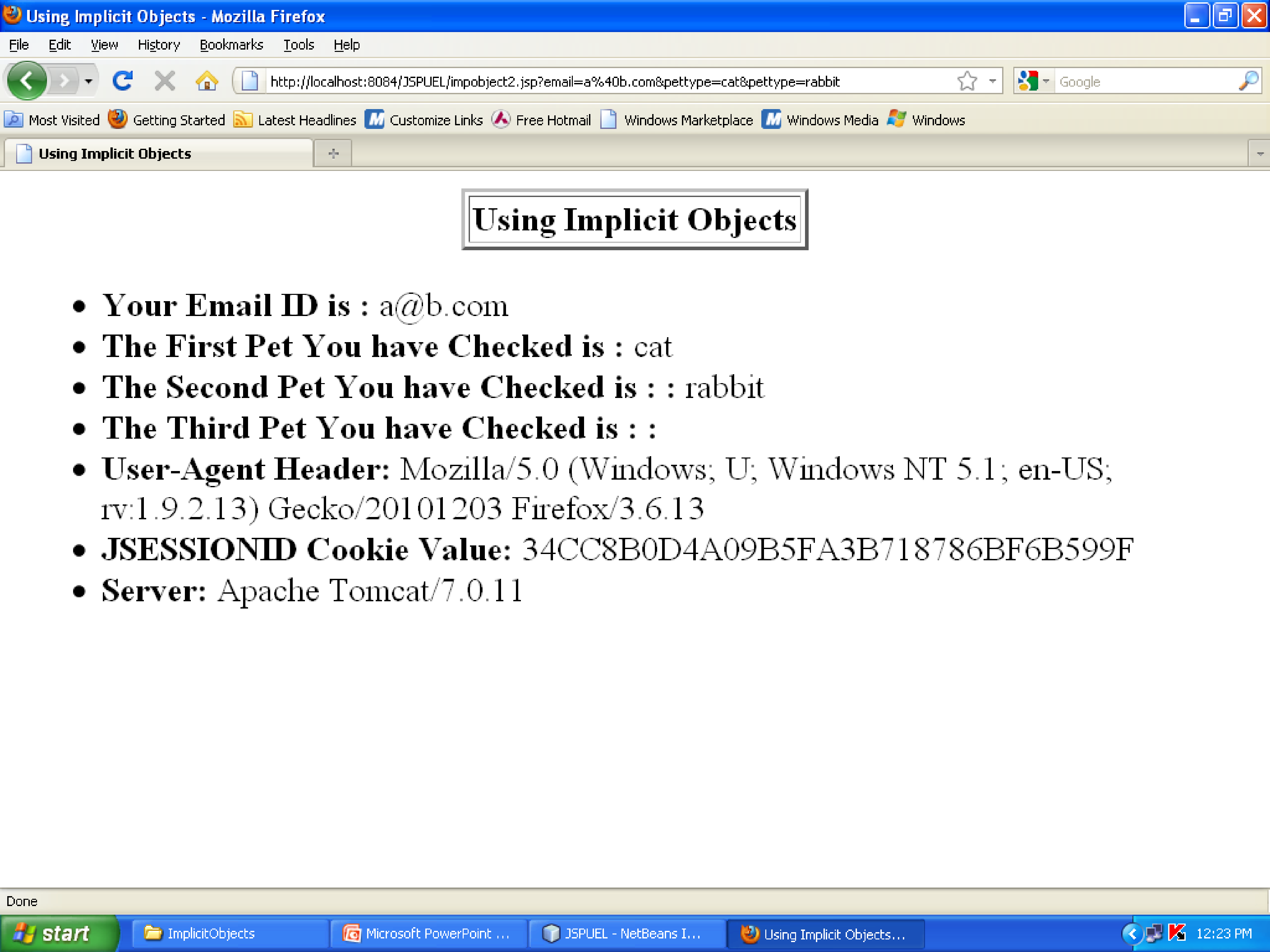
Submit


Using Implicit Objects - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8084/JSPUEL/impobject2.jsp?email=a%40b.com&pettype=cat&pettype=rabbit
Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Using Implicit Objects
Done
start ImplicitObjects Microsoft PowerPoint ... JSPUEL - NetBeans I... Using Implicit Objects...
12:45 PM

Using Implicit Objects

- **Your Email ID is :** a@b.com
- **The First Pet You have Checked is :** cat
- **The Second Pet You have Checked is : :** rabbit
- **The Third Pet You have Checked is : :**
- **User-Agent Header:** Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13
- **JSESSIONID Cookie Value:**
- **Server:** Apache Tomcat/7.0.11

Refresh


Using Implicit Objects - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8084/JSPUEL/impobject2.jsp?email=a%40b.com&pettype=cat&pettype=rabbit
Google
Most Visited Getting Started Latest Headlines Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Using Implicit Objects
Done
start
ImplicitObjects
Microsoft PowerPoint ...
JSPUEL - NetBeans I...
Using Implicit Objects...
12:23 PM

Using Implicit Objects

- **Your Email ID is :** a@b.com
- **The First Pet You have Checked is :** cat
- **The Second Pet You have Checked is : :** rabbit
- **The Third Pet You have Checked is : :**
- **User-Agent Header:** Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13
- **JSESSIONID Cookie Value:** 34CC8B0D4A09B5FA3B718786BF6B599F
- **Server:** Apache Tomcat/7.0.11