Carnegie
Mellon
University

95-733 Internet of Things
Server-side Java and AJAX
Week Two

Week Two

- Objectives:
- Understand AJAX and server side Java
- Introduce websockets with a demonstration

A Complete Example From IBM

Demonstrations:

See article by Philip McCarthy on Ajax at:
http://www.ibm.com/developerworks/library/j-ajax1/

Eclipse workspace mm6/95-733
Visit http://localhost:8080/AjaxProject/index.jsp

Netbeans workspace mm6/95-733/AjaxUsingNetbeans
ACoolAjaxShoppingCart and
ACoolAjaxShoppingCartImproved

Main Features of Index.jsp (1)

This page needs access to these imported Java packages.

```
<%@ page import="java.util.*" %>
<%@ page import="developerworks.ajax.store.*"%>
```

The store package contains
Item.java, Catalog.java and Cart.java

Main Features of Index.jsp (2)

The header holds pointers to script code to be included in the page and then executed by the browser.

```
<script type="text/javascript"
        language="javascript" src="ajax1.js">
</script>
<script type="text/javascript"
        language="javascript" src="cart.js">
</script>
```

This is not constrained by the same origin policy! These scripts could be fetched from any URL.

Main Features of Index.jsp (3)

The table body tag encloses table rows that are generated on the server. This is Java (part of JSP) and is not JavaScript.

```
<tbody>

 <%
   for (Iterator<Item> I =
        new Catalog().getAllItems().iterator();
        I.hasNext() ; ) {
                Item item = I.next();
  %>
```

Main Features of Index.jsp (4)

The addToCart call is a call on JavaScript.

```
  <!-- For each Item do -->
  <tr>
     <td><%= item.getName() %></td>
     <td><%= item.getDescription() %></td>
     <td><%= item.getFormattedPrice() %></td>
     <td>
      <button onclick="addToCart('<%= item.getCode() %>')">
                       Add to Cart
      </button>
     </td>
  </tr>
    <% } %>
</tbody>
```

Main Features of Index.jsp (5)

The <ul> and <span> tags each have a unique identifier. These identifier will be used by the event handler.

```
<div style=”…”>
   <h2>Cart Contents</h2>
   <ul id="contents"></ul>
   Total cost:
   <span id="total">$0.00</span>
</div>
```

Main Features of Item.java

```java
package developerworks.ajax.store;

import java.math.BigDecimal;

public class Item {
  private String code;
  private String name;
  private String description;
  private int price;

  public Item(String code,String name,String description,int price) {
    this.code=code;
    this.name=name;
    this.description=description;
    this.price=price;
  }

  public String getCode() {
    return code;
  }
```

Getter and setters on code, name,description and price.

Main Features of Catalog.java(1)

```
package developerworks.ajax.store;
import java.util.*;
public class Catalog {

  private static Map<String,Item> items;

  static {
   items = new HashMap<String,Item>();
   items.put("hat001",new Item("hat001",
        "Hat","Stylish bowler hat (SALE!)",
         1999));
   :
}
```

Main Features of Catalog.java(2)

```
  public Collection<Item> getAllItems() {
    return items.values();
   }

  public boolean containsItem(String itemCode)
  {
    return items.containsKey(itemCode);
   }

  public Item getItem(String itemCode) {
    return items.get(itemCode);
  }
}
```

Main Features of Cart.java(1)

```
package developerworks.ajax.store;

public class Cart {

   private HashMap<Item,Integer> contents;
   public Cart() {
    contents = new HashMap<Item,Integer>();
  }
```

Main Features of Cart.java(2)

```
public void addItem(String itemCode) {
    Catalog catalog = new Catalog();
    if (catalog.containsItem(itemCode)) {
      Item item = catalog.getItem(itemCode);
      int newQuantity = 1;
      if (contents.containsKey(item)) {
         Integer currentQuantity = contents.get(item);
         newQuantity += currentQuantity.intValue();
      }
      contents.put(item, new Integer(newQuantity));
    }
 }
```

Main Features of Cart.java(3)

```
public String toXml() {

 StringBuffer xml = new StringBuffer();
  xml.append("<?xml version=\"1.0\"?>\n");
  xml.append("<cart generated=\""+
       System.currentTimeMillis()+
        "\" total=\""+getCartTotal()+"\">\n");

  for (Iterator<Item> I = contents.keySet().iterator() ; I.hasNext() ; ) {
    Item item = I.next();
    int itemQuantity = contents.get(item).intValue();

    xml.append("<item code=\""+item.getCode()+"\">\n");
    xml.append("<name>");
     :
    xml.append("</cart>\n");
  return xml.toString();
}
```

Main Features of CartServlet.java(1)

```
private Cart getCartFromSession(HttpServletRequest  req)
 {
  HttpSession session = req.getSession(true);
  Cart cart = (Cart)session.getAttribute("cart");
  if (cart == null) {
       cart = new Cart();
       session.setAttribute("cart", cart);
   }
   return cart;
}
```

Main Features of CartServlet.java(2)

```
public void doPost(HttpServletRequest req,
                   HttpServletResponse res)
                   throws java.io.IOException {


    Cart cart = getCartFromSession(req);

    String action = req.getParameter("action");
    String item = req.getParameter("item");
```

Main Features of CartServlet.java(3)

```
 if ((action != null)&&(item != null)) {

    if ("add".equals(action)) {
      cart.addItem(item);

    } else if ()…..
  }

  String cartXml = cart.toXml();
  res.setContentType("text/xml");
  res.getWriter().write(cartXml);
}
```