AJAX Clients and the SAP NetWeaver Platform

Platform Ecosystem from SAP
<table>
<thead>
<tr>
<th>Learning Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing AJAX</td>
</tr>
<tr>
<td>Developers’ Considerations</td>
</tr>
<tr>
<td>AJAX Frameworks</td>
</tr>
<tr>
<td>Demo: AJAX Application on SAP NetWeaver</td>
</tr>
<tr>
<td>Summary</td>
</tr>
</tbody>
</table>
Learning Objectives

- Understand what AJAX is and what it can do for you
- Understand how to AJAX-enable Web applications
- Understand how to use an AJAX server-side framework with the SAP NetWeaver platform

AJAX +
AJAX refers to the **general techniques** of building interactive, desktop-like Web applications

An important part of Web 2.0

Data exchange with the server using
- JavaScript behind the scenes
- `XMLHttpRequest` object
- Hidden iFrame

Modification of the Web page’s own data structures through its DOM tree

No entire Web page **reload** each time the user requests a change
All user interactions must pass through the server **synchronously**

Web clients display only **static** (HTML) content
- The browser interacts with the server **asynchronously** – by prefetching data from the server

- Smarter interaction with the server – by **validating** the input data

- Richer user interface (UI) behavior
  - Drag and drop
  - Sliders
  - Client-side rendering
  - And so on

![Diagram of AJAX components and interaction](image-url)
What Else Do You Need to Know About AJAX?

- AJAX is a **powerful**, yet different, programming paradigm compared with the traditional Web application development model
  - A Web application is **no longer a series** of distinct Web pages
  - The **user experience** is different, because changes on the Web page are **not** as visible as a **page reload**

- Understand the power of AJAX to fully leverage its benefits

Be aware of the danger of **AJAX power abuse**
A Note on Performance

- Response time **no longer measured by user-to-server response time alone**
- **AJAX features could worsen overall performance and network traffic**
  - Overuse of asynchronous prefetching
  - Chattiness is a traffic killer
- **Browsers could end up doing more work due to additional JavaScript processing** of an AJAX request
- User abuse of an AJAXed application **could worsen overall performance** – by clicking around
A Note on Client- and Server-Side Computing

- Shifting part of the work from server to client (browser) is possible but
  - Business logic should still be implemented on the server side by business application developers
  - Data should still reside on the server side
  - AJAX is and should stay a front-end technology
  - Security issues must be seriously considered
The Development Scenarios

Building an AJAXed Web application from scratch

- Decide if JSON (JavaScript Object Notation) suits your needs

AJAXing an existing Web application

- Existing server-side coding – Java Server components
- Integration of a server-side AJAX framework
- Integration of a client-side AJAX framework
The New Development Challenges

- Most developers are familiar with server-side development and are uneasy about shifting to client-side development.

- New challenges are:
  - JavaScripting
  - DOM handling
  - Cross-browser support
  - New development tools

- Need for simple-to-use frameworks embracing AJAX capabilities but no tedious XMLHttpRequest parsing or DOM coding in JavaScript.
<table>
<thead>
<tr>
<th>Learning Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing AJAX</td>
</tr>
<tr>
<td>Developers’ Considerations</td>
</tr>
<tr>
<td><strong>AJAX Frameworks</strong></td>
</tr>
<tr>
<td>Demo: AJAX Application on SAP NetWeaver</td>
</tr>
<tr>
<td>Summary</td>
</tr>
</tbody>
</table>
Overview of AJAX Frameworks

- AJAX frameworks help programmers

- **On the client side**
  - Offer JavaScript functions to send requests to the server
  - Manage the DOM tree
  - Offer GUI widgets and libraries for easier client-side UI development

- **On the server side**
  - Process the requests in collaboration with server components
  - Often generate JavaScript code

- Existence of many commercial frameworks – server-side as well as client-side

- Frameworks and tool kits are still evolving
Client-Side AJAX Frameworks

- Rely solely upon JavaScript – no server components
- Abstract the browser differences
- Usually have their own object model
- Usually comprise a GUI widget library to help build client-side UIs
- Require different set of skills – UI development, DOM data manipulation, and XMLHttpRequest on the client side
- Reduce the amount of JavaScript but do not eliminate it

Examples
- Dojo
- Qooxdoo
- Many more . . .
Server-Side and Mixed Type Frameworks

- **Familiar server-side scripting only** – client-side AJAX behavior is generated automatically
- **Enable developers to work in their familiar IDEs** – ASP.NET
- **Use custom tag libraries**
  - AJAX on rails
  - AJAX4JSF – used by the demo
- **Mixed types** provide support on both the client and server sides
  - Google Web tool kit
  - Yahoo UI library
- **Proxy-like frameworks** enable you to invoke arbitrary logic on the server side from your JavaScript code
  - Direct Web remoting
AJAXing a Web Application

The un-AJAXed version

- Have to enter all data-entry fields
- Data input values are validated on the next page reload; users won’t know until they click on the submit button
- Cannot display list of employees in a scrollable table

The AJAXed version

- Auto-fill data entry fields by AJAXing the server
- Data input values are validated on the fly (error messages are also displayed on the fly before user submits)
- Dynamic table (scrolling data on the fly without page reload)
Employee Management Application
Behind the Scenes Stuff:
On-the-Fly Data Input Value Validation

The un-AJAXed version

```
<h:inputText
    value="#{createempbean.empid}"
    maxlength="6" size="6"
    required="true"
    requiredMessage="#{demomsgs.createemperrorwrongempid}"
    validator="#{createempbean.validatempK}">
</h:inputText>
```

Data validation does not occur until the user clicks the submit button

The AJAXed version

```
<h:inputText
    value="#{createempbean.empid}"
    maxlength="6" size="6"
    required="true"
    requiredMessage="#{demomsgs.createemperrorwrongempid}"
    validator="#{createempbean.validatempK}">
    <a4j:support event="onchange"
        reRender="errmsgdisp"
        AJAXSingle="true" />
</h:inputText>
```

Data validation occurs as soon as the user leaves the employee ID input field
More Behind the Scenes Stuff: Dynamic Table Scrolling

The un-AJAXed version

```xml
<h:dataTable first="#{emplistbean.curPos}" rows="#{emplistbean.numToDisplay}" value="#{emplistbean.all}" var="cur" ...
</h:dataTable>

One also needs to create the table navigation buttons

```xml
<h:panelGrid columns="6">
<h:column><h:outputText value="#{demomsgs.listnumtодisp}" />
</h:column>

...</h:panelGrid>
```

The AJAXed version

```xml
<d:dynGrid var="cur" value="#{emplistbean.all}" visibleLines="10">
...
</d:dynGrid>

Note that there is no need for table navigation buttons
Learning Objectives

Introducing AJAX

Developers’ Considerations

AJAX Frameworks

Demo: AJAX Application on SAP NetWeaver

Summary
Summary

- AJAX is here to stay

- Use AJAX only when it makes sense
  - Understand the power and benefits of AJAX
  - Avoid the AJAX power-abuse trap

- Understand different types of AJAX frameworks and their usage

- Learn to use a server-side framework to implement AJAX applications in the SAP NetWeaver platform

- SAP is enabling Web Dynpro to support building rich Internet applications
Questions?

Q&A