DNA of Ajax Applications

Designing and Building Ajax Applications with Patterns

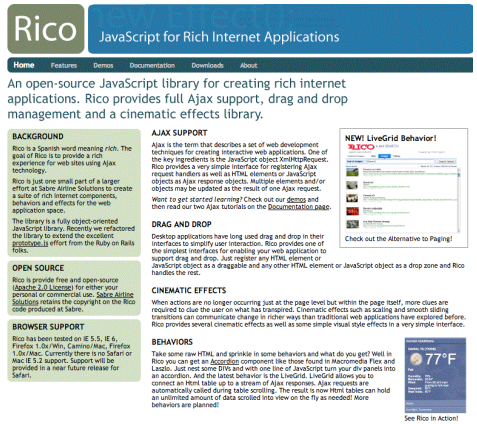
adaptive path

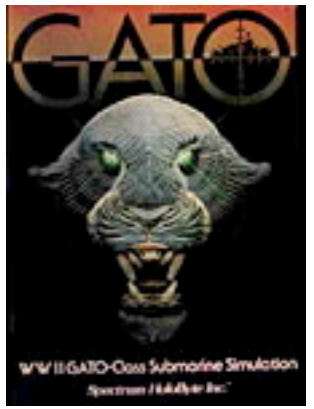
Bill W. Scott, Y! Ajax Evangelist

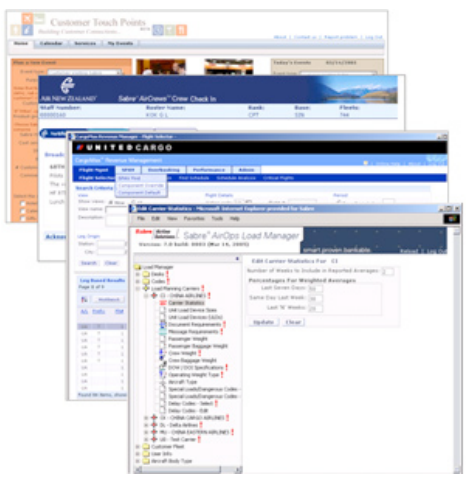
b.scott@yahoo.com

Background

- Games, research, science, military, biz
- Apps, frameworks, widget sets
- Developer, designer
- Object-oriented background
- Software architect, user experience manager
- Yahoo! Ajax Evangelist, Yahoo! Patterns Curator
- http://looksgoodworkswell.com





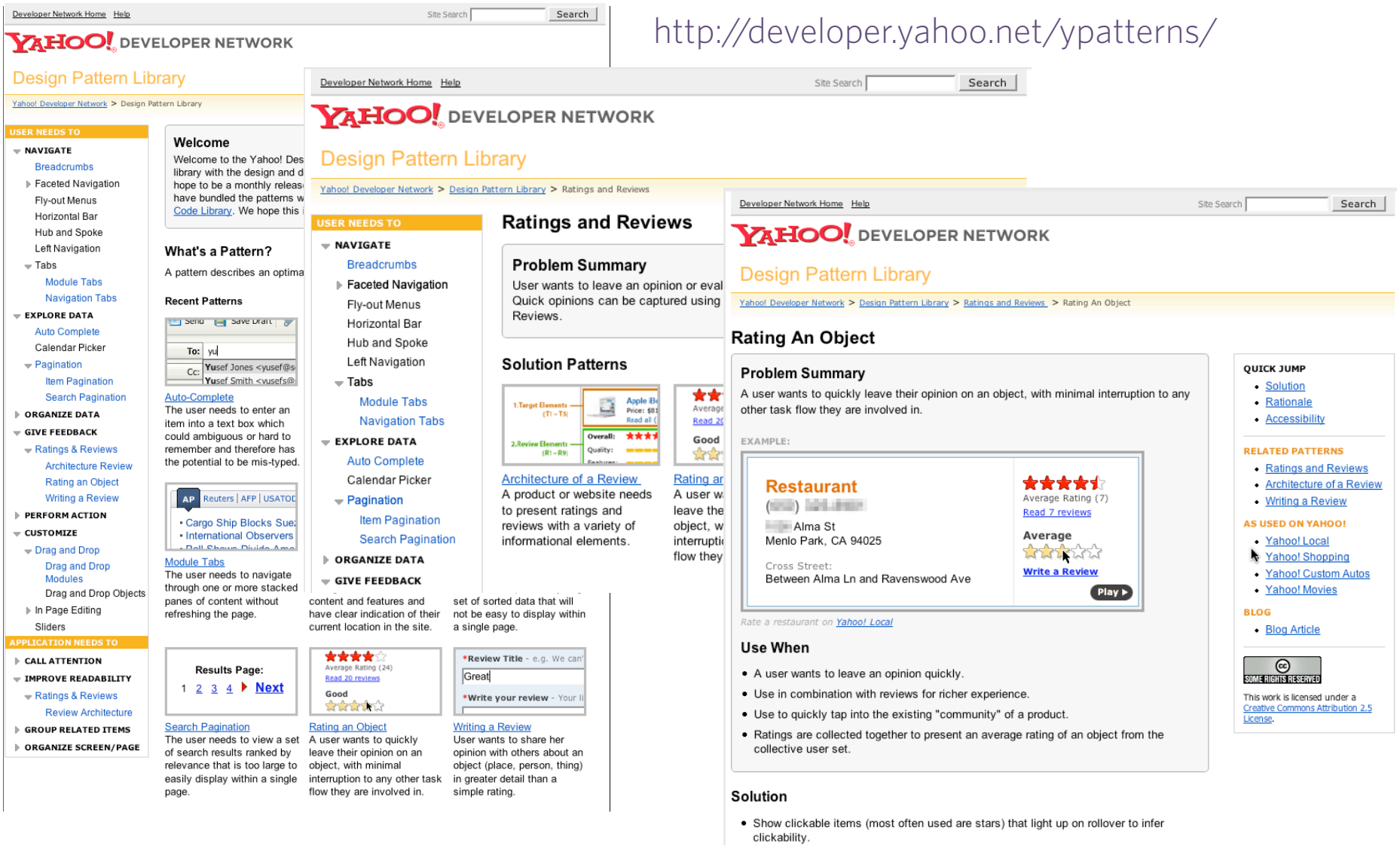


Ajax at Yahoo!

- Acquisition of Oddpost led to mission to evangelize *Ajax goodness throughout Yahoo*
- Summer 2005 - Ajax Evangelism Team Founded
  - Doug Crockford - JSON, Y! DHMTL Architect
  - Iain Lamb - Founder of Oddpost/Y!Mail Beta
  - Me

Yahoo! Pattern Library Release

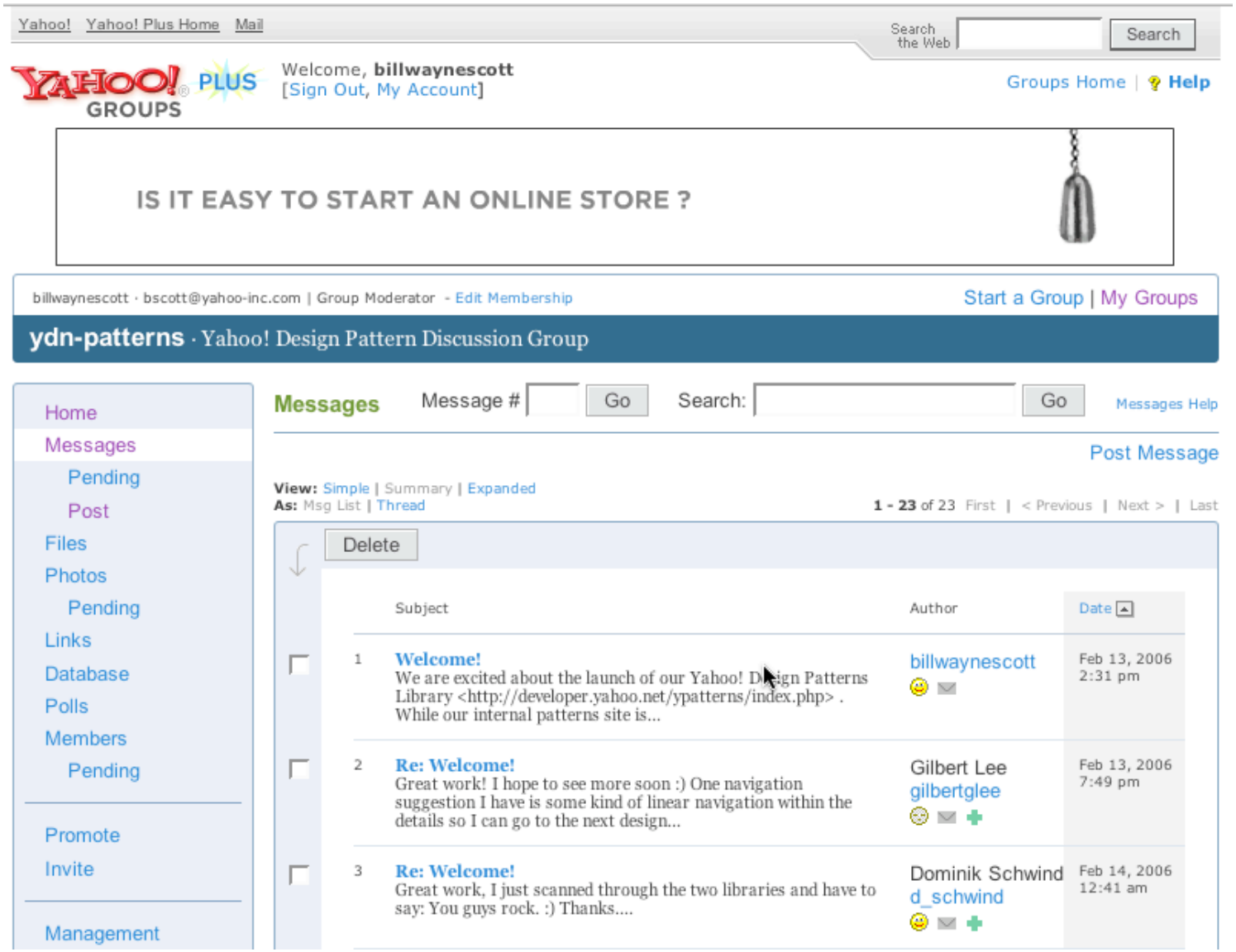
http://developer.yahoo.net/ypatterns/



http://yuiblog.com

Yahoo! Patterns Discussion Group

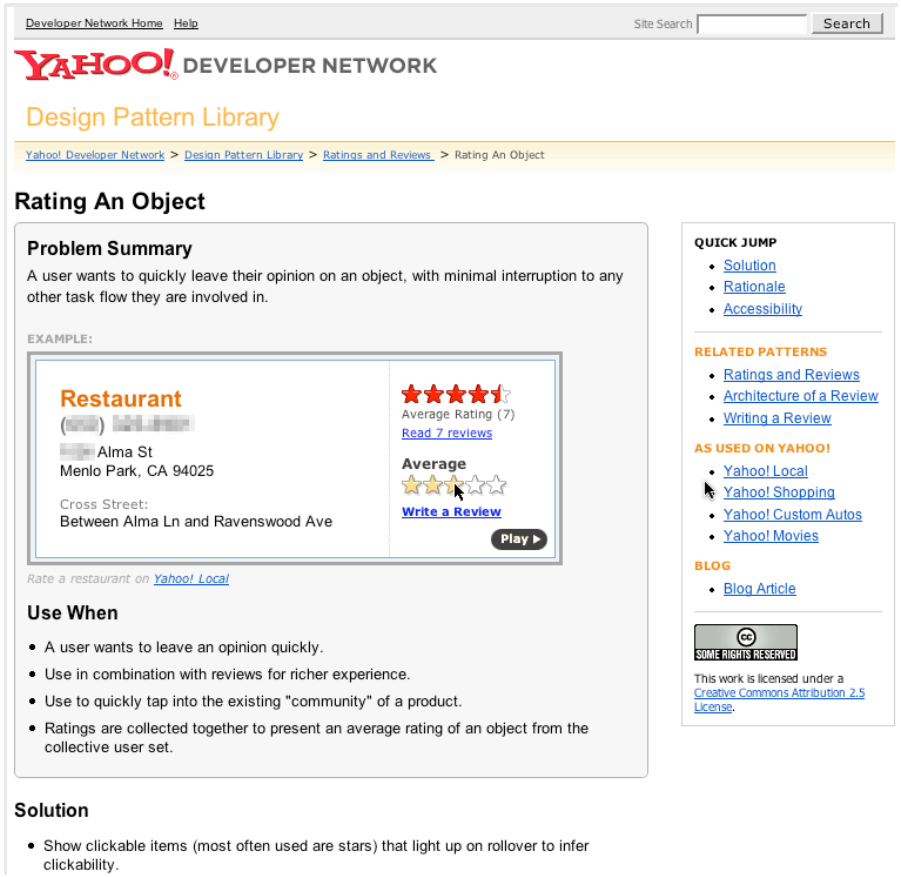
http://groups.yahoo.com/group/ydn-patterns/



Breadcrumbs. Module Tabs. Navigation Tabs. Auto Complete. Pagination. Item Pagination. Search Pagination. Ratings and Reviews. Architecture of a Review. Rating an Object. Writing a Review. Drag and Drop. Drag and Drop Modules. Transition. Dim. Brighten. Cross Fade. Contract. Expand. Fade In. Fade Out. Move. Self-Healing. Slide. Highlight. Invitation. Cursor Invitation. Drop Invitation. Tool Tip Invitation. Hover Invitation. Links. Alphanumeric Links. Text Links.

Drag and Drop. Drag and Drop Modules. In Page Editing. In Page Custom Editing. Direct State Editing. Grid Cell Editing. Inline Custom Editing. Inline Tag Editing. Popup Custom Editing. Slide-out Custom Editing. Inline Text Editing. Persistent Portals. Inline Reordering. Indication. Busy Indication. Cursor Busy. In Context Busy. In Context Progress. Inline Status. Character Counter. Auto Complete. Balloon Error Tip. Deferred Content Loading. Dynamic Goal. Narrowing Choices. Refining Search. Live Search. Dynamic Filter. Invitation. Cursor Invitation. Drop Invitation. Tool Tip Invitation. Hover Invitation. Detail Zoom. Opacity Focus. Configurable Module - Faceplate. Configurable Module - Flip It. Configurable Module - Inline Configure. Configurable Module - Slide Out Drawer. Slide Out. Flip. Opacity Fade. Endless Scrolling. Expandable Paging Boundary. Fresh Content. Hover Detail. In Place Drill Down. Inline Assistant. Inline Validation. Validate Then Suggest. On Demand Refresh. Periodic Refresh. Resizable Modules. Scrolling Modules. Auto Save. In Context Tools. Remembered Collection. Remembered Preferences. Auto Form Fill. Rating an Object. Transition. Brighten Transition. Cross Fade Transition. Dim Transition. Expand Transition. Fade In Transition. Fade Out Transition. Flip Transition. Move Transition. Self-Healing Transition. Collapse Transition. Slide Transition. Rich Internet Object. Available. Selected. Identifiable Object.

Y! Patterns as a Platform (Web 2.0)

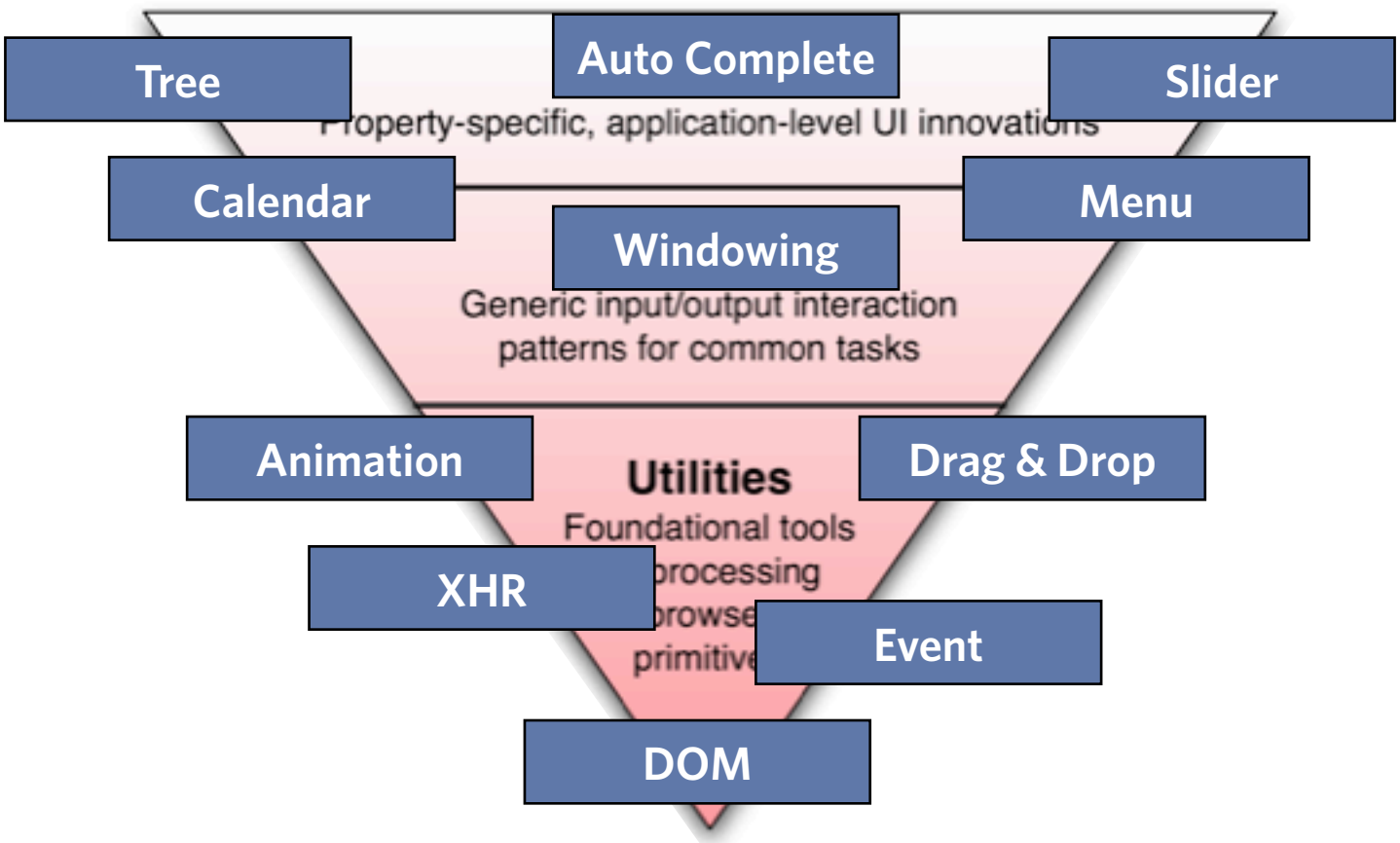


Yahoo! service responds with a JSON representation of a pattern. Allows anyone to mashup the patterns into a corporate pattern library

Pattern Service

Corporate Branded Pattern Site (mashup)

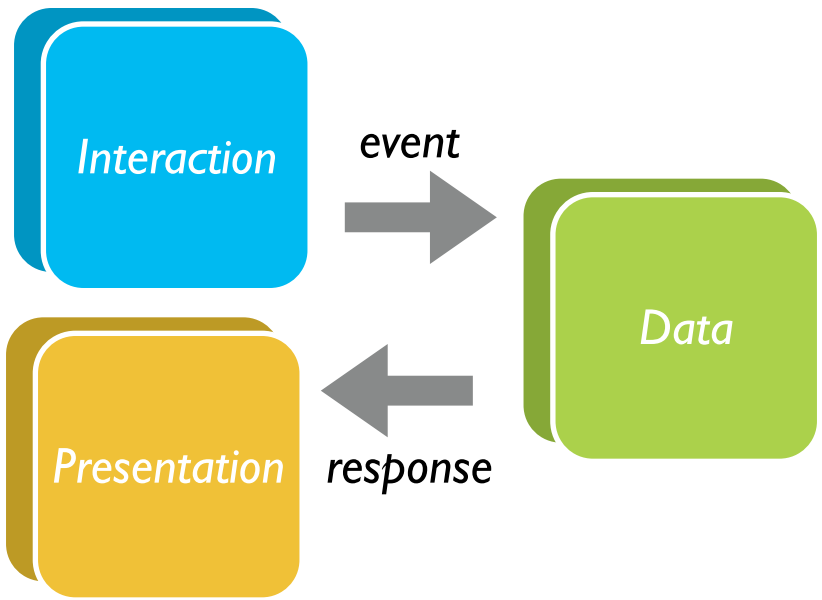
Yahoo! UI Library



Richness

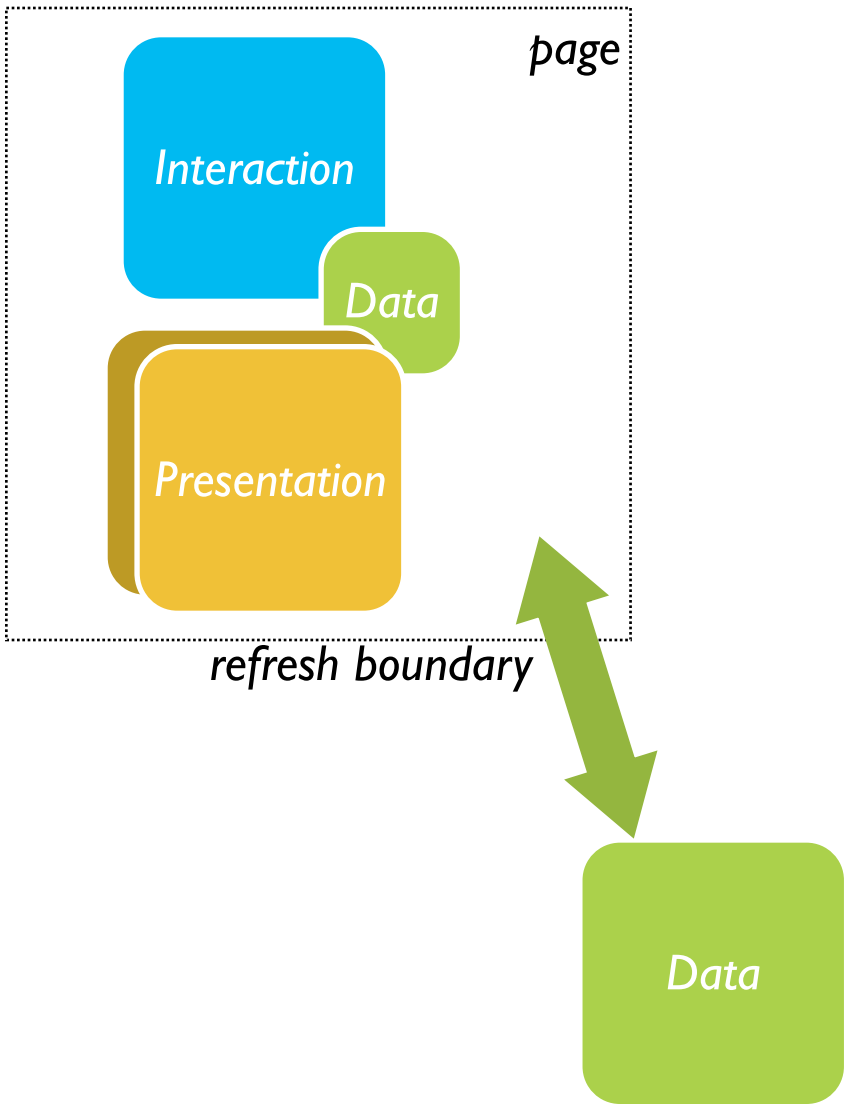
Defining Richness: Web in 3D

- Interaction Dimension
- Data Dimension
- Presentation Dimension

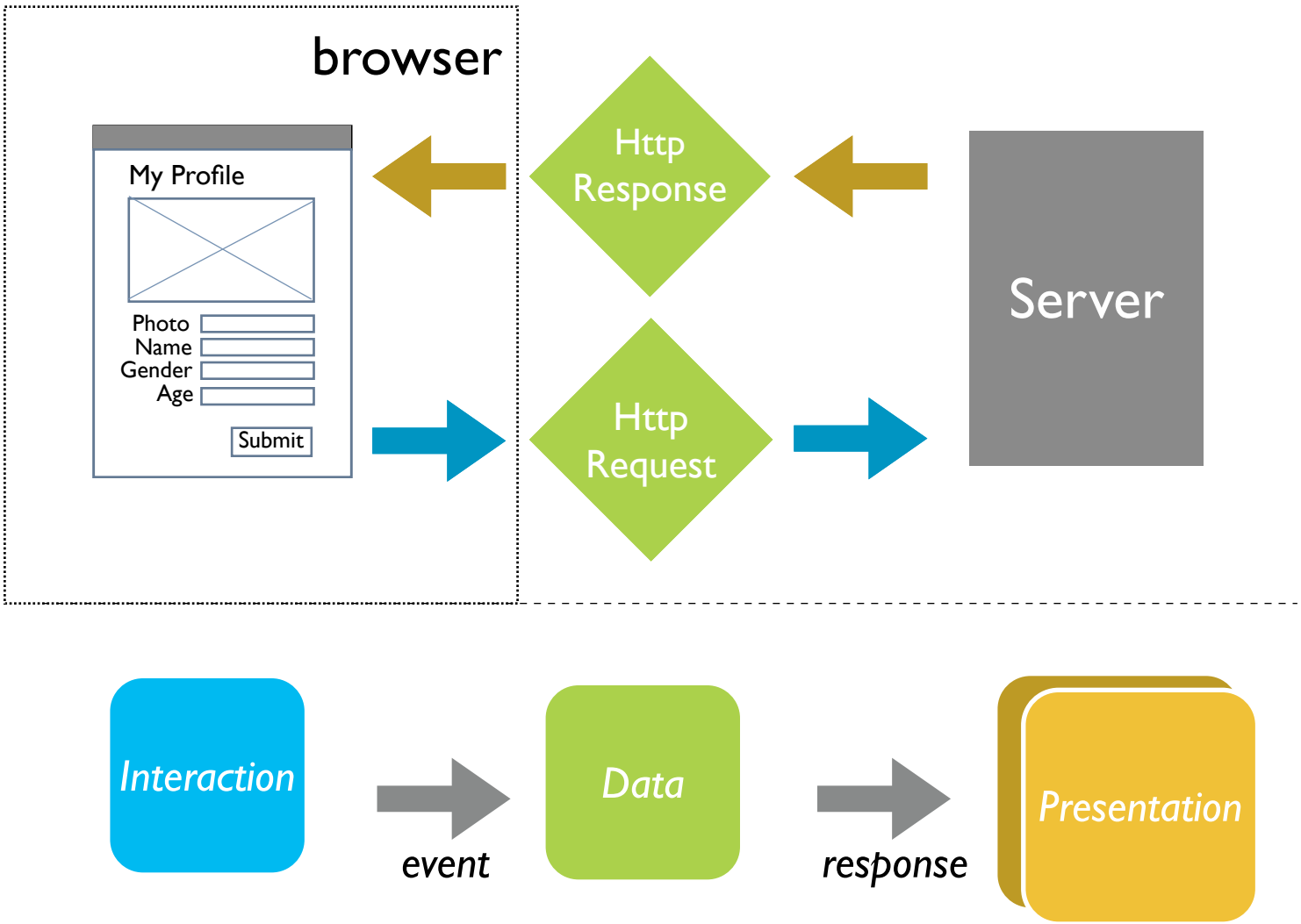


Classic Web

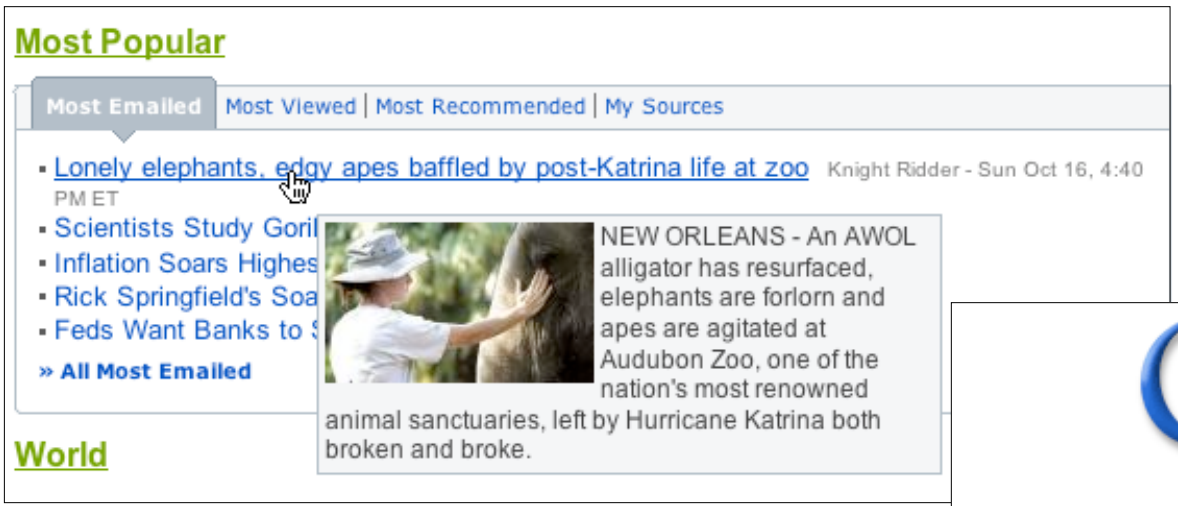
- Interaction
  course-grained
- Data
  loaded with the page or
  obtained via refresh
- Presentation:
  page-based updates



Classic Web Illustrated

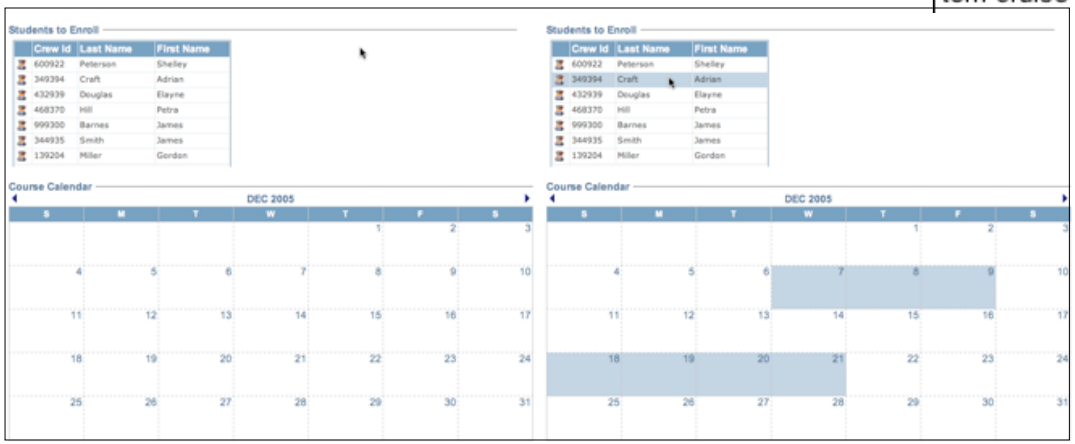


Ajax = Just-in-Time. Immediacy.



Just-in-time data

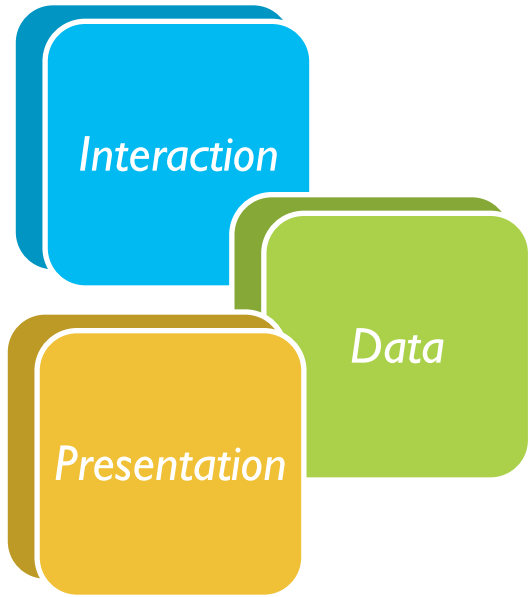
Just-in-time presentation



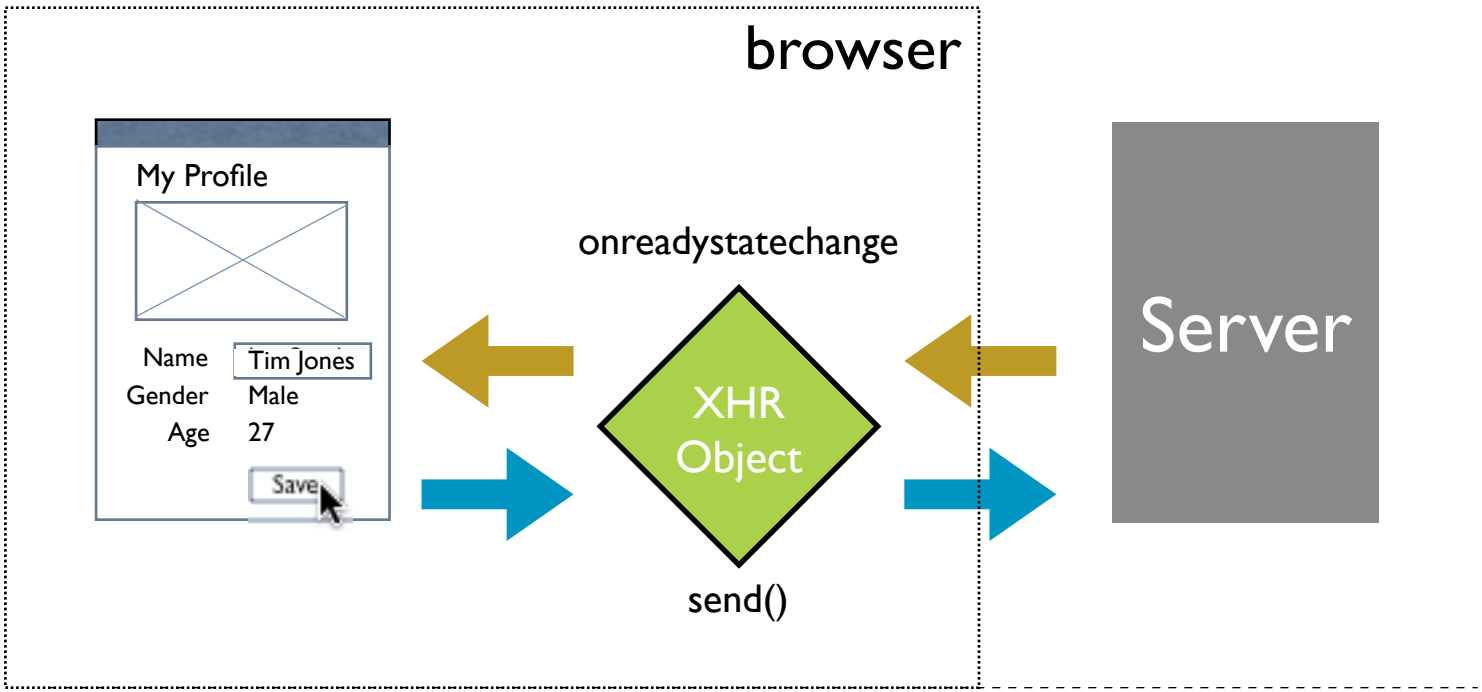
Just-in-time logic

Ajax Model - New & Improved!

- New! Deeper Interaction
- Now with Just-in-Time Data & Just-in-Time Logic!
  - ★ XMLHttpRequest (XHR) is the secret sauce!
- Now with Richer Interface!
- All dimensions are closer



Ajax Illustrated



AJAX vs Ajax

- AJAX = Asynchronous JavaScript and XML
  - Strict definition is using XMLHttpRequest (XHR) to retrieve XML within a web page
- Ajax = The set of technologies that allow web applications to provide rich interaction, just-in-time information and dynamic interfaces without required page refresh
- The Secret Sauce

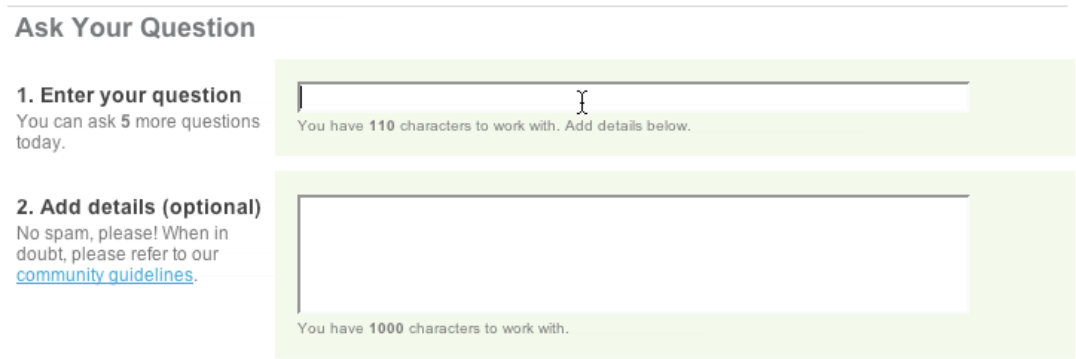
  - Ajax = XHR + DHTML (HTML, CSS, JavaScript) + Rich design

Other Remote Scripting Techniques

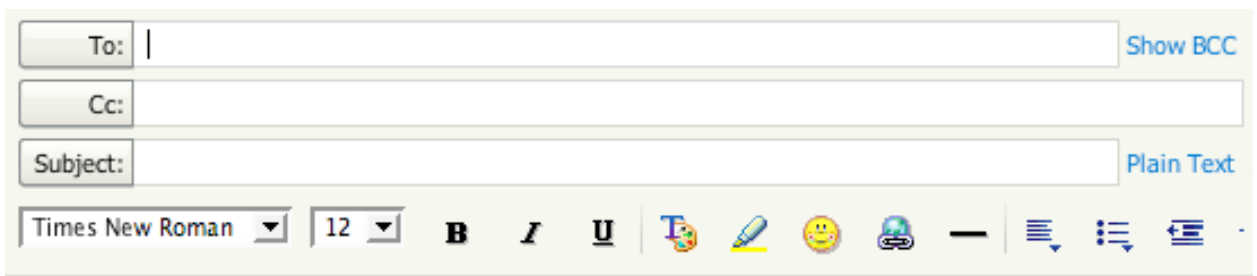
- Hidden IFrame
- <img> src
- <script> src hack
- CSS href hack
- JS to faceless Java applets
- JS to faceless Flash
- NO CONTENT Response
- Cookies

Patterns

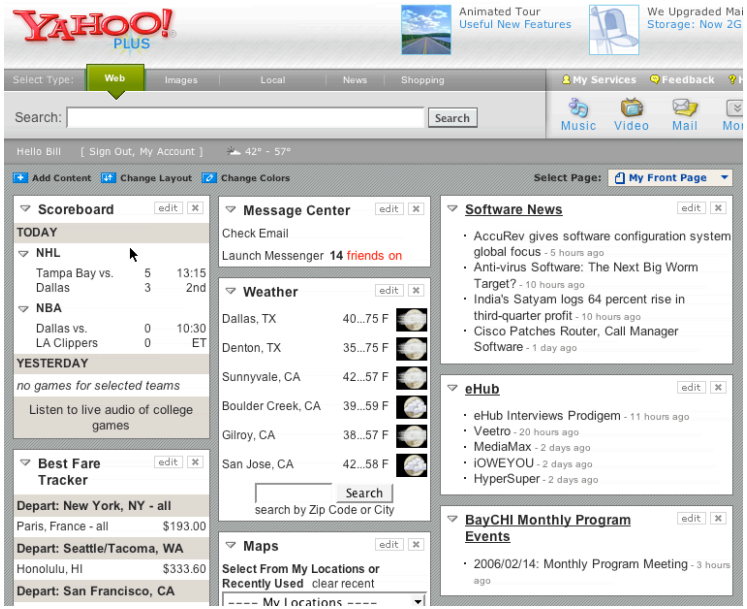
Drag and Drop. Drag and Drop Modules. In Page Editing. In Page Custom Editing. Direct State Editing. Grid Cell Editing. Inline Custom Editing. Inline Tag Editing. Popup Custom Editing. Slide-out Custom Editing. Inline Text Editing. Persistent Portals. Inline Reordering. Indication. Busy Indication. Cursor Busy. In Context Busy. In Context Progress. Inline Status. Character Counter. Auto Complete. Balloon Error Tip. Deferred Content Loading. Dynamic Goal. Narrowing Choices. Refining Search. Live Search. Dynamic Filter. Invitation. Cursor Invitation. Drop Invitation. Tool Tip Invitation. Hover Invitation. Detail Zoom. Opacity Focus. Configurable Module - Faceplate. Configurable Module - Flip It. Configurable Module - Inline Configure. Configurable Module - Slide Out Drawer. Slide Out. Flip. Opacity Fade. Endless Scrolling. Expandable Paging Boundary. Fresh Content. Hover Detail. In Place Drill Down. Inline Assistant. Inline Validation. Validate Then Suggest. On Demand Refresh. Periodic Refresh. Resizable Modules. Scrolling Modules. Auto Save. In Context Tools. Remembered Collection. Remembered Preferences. Auto Form Fill. Rating an Object. Transition. Brighten Transition. Cross Fade Transition. Dim Transition. Expand Transition. Fade In Transition. Fade Out Transition. Flip Transition. Move Transition. Self-Healing Transition. Collapse Transition. Slide Transition. Rich Internet Object. Available. Selected. Identifiable Object.



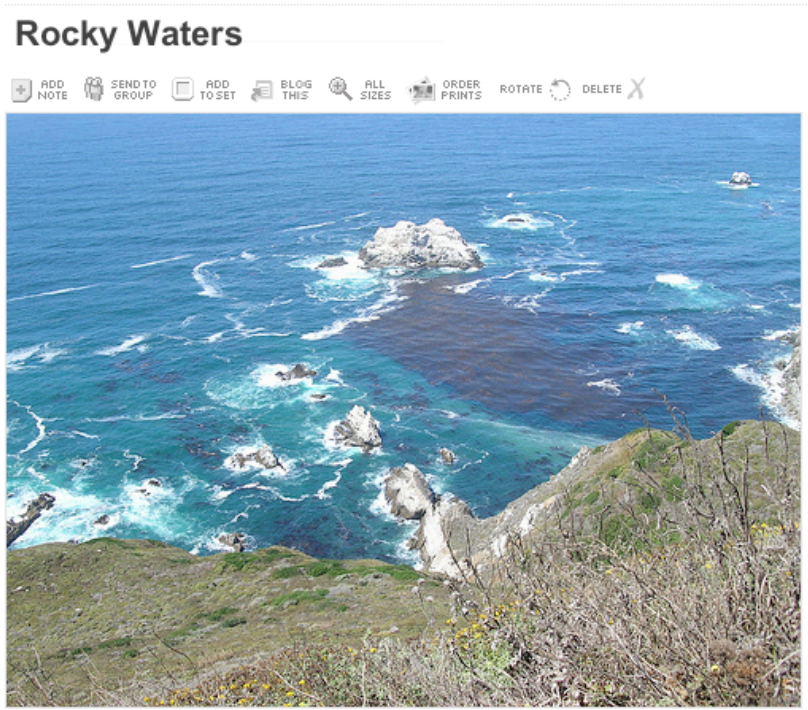
pattern. Live Suggest.



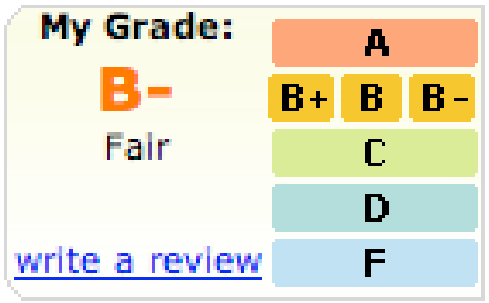
pattern. Auto Complete.



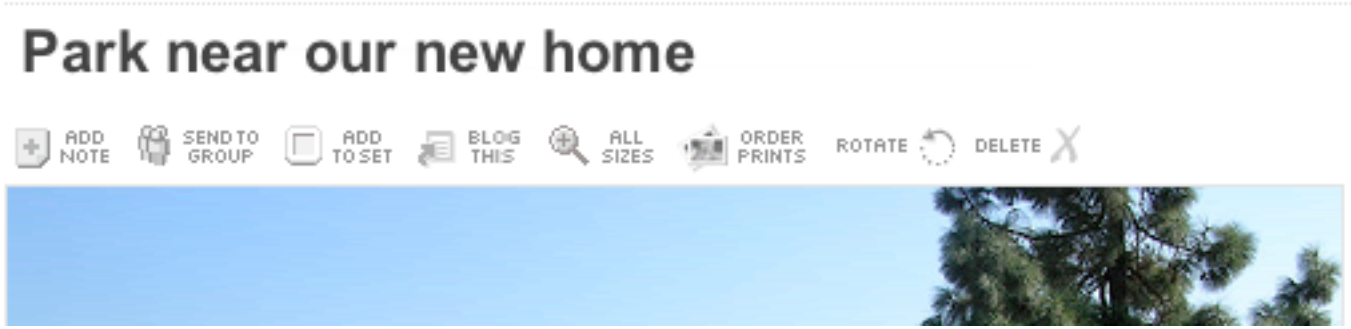
pattern. Drag and Drop.



pattern. Inline Editing.

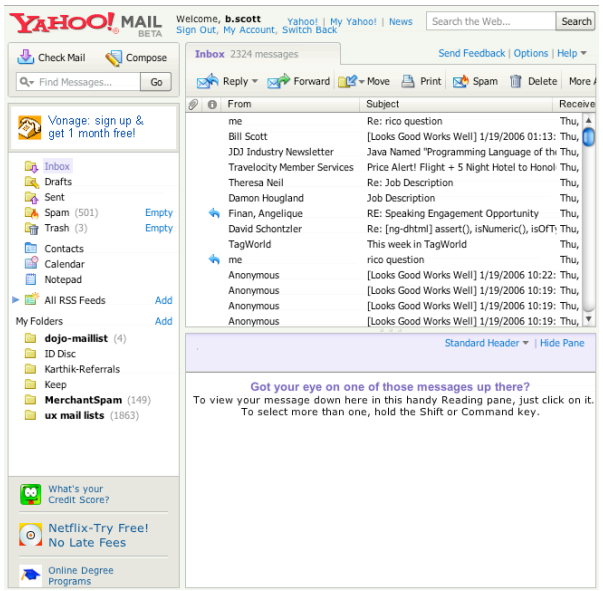


pattern. Hover Invitation.

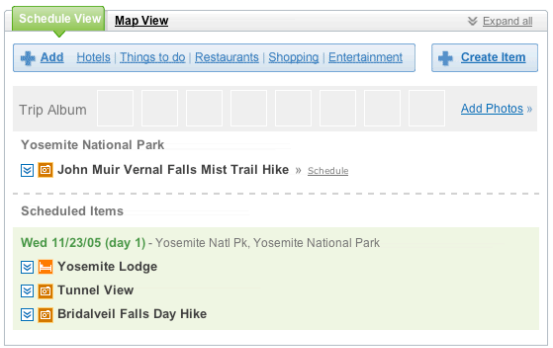


pattern. Tooltip Invitation + Hover Invitation.

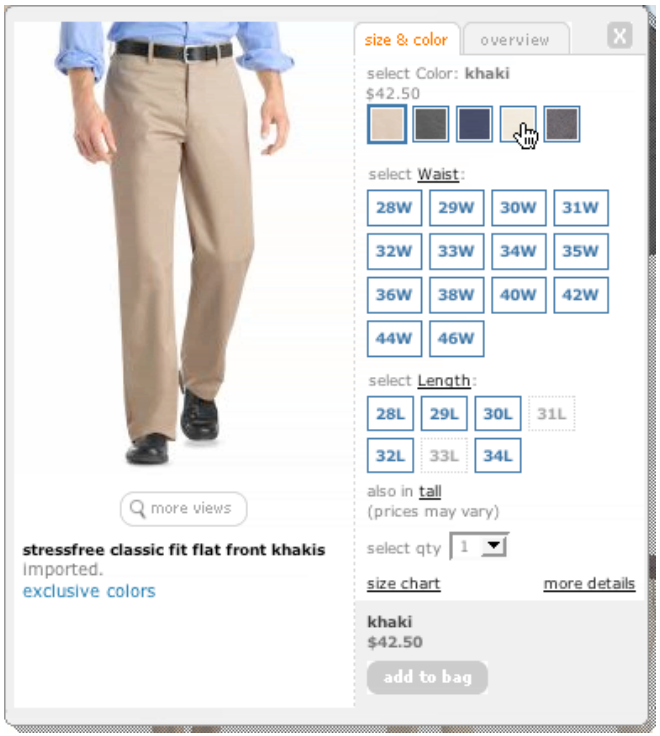
Patterns. In Page Interaction.



pattern. Endless Scrolling.

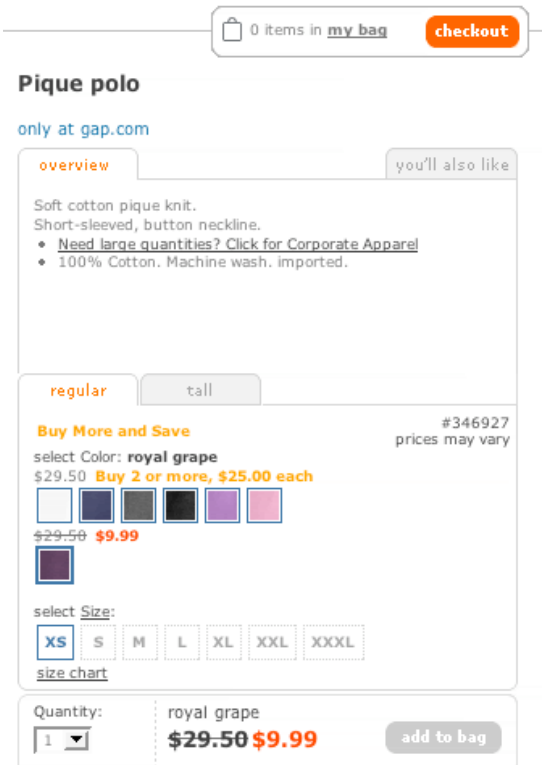


pattern. In Context Expand.



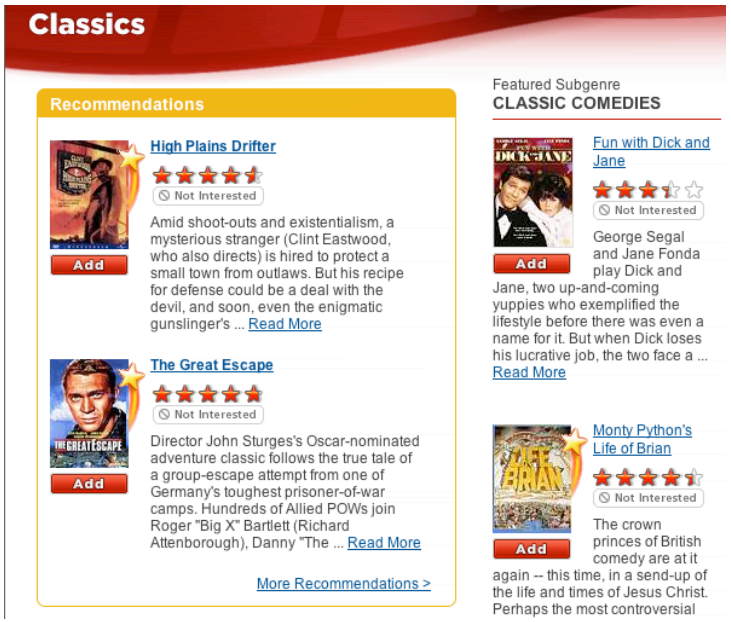
pattern. Inline Assistant.

pattern. Hover Details.

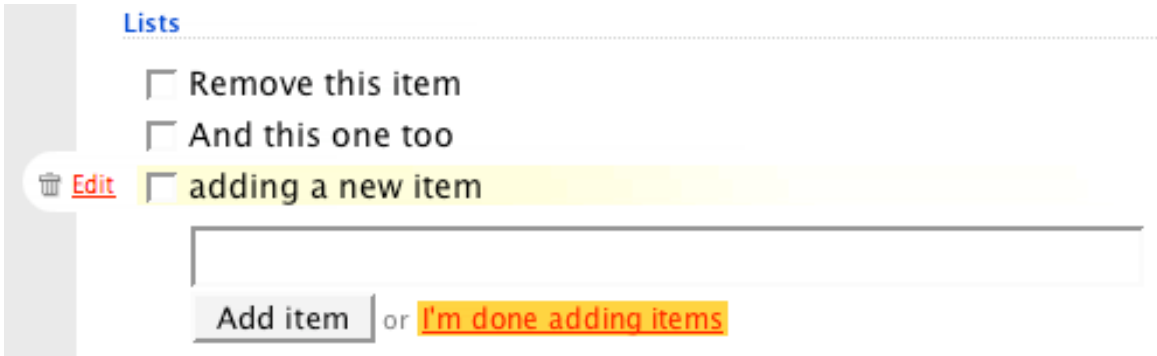


pattern. Remembered Collection.

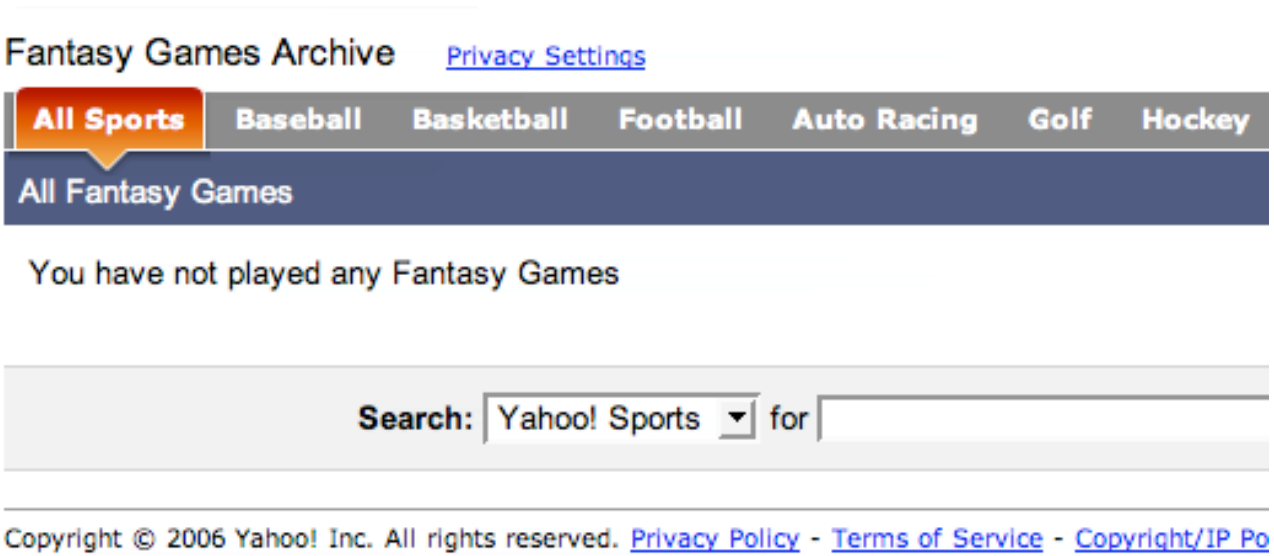
pattern. Rating an Object.



pattern. In Page Action.

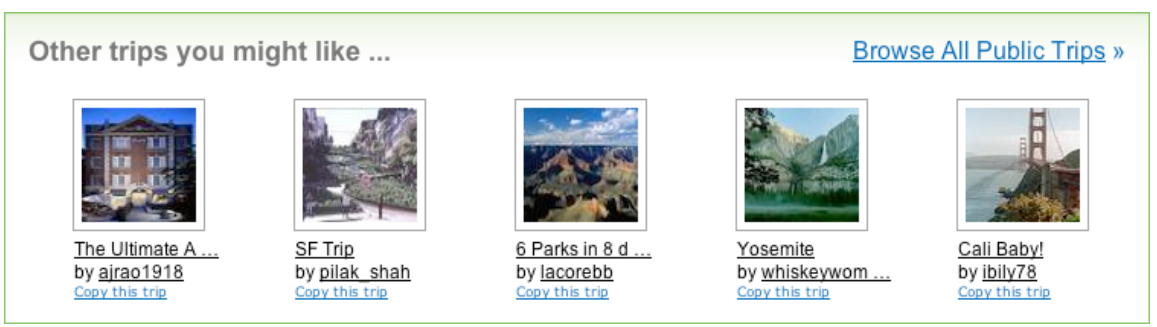


pattern. Fade Transition + Self-Healing Transition.



pattern. Slide + Fade In + Fade Out.

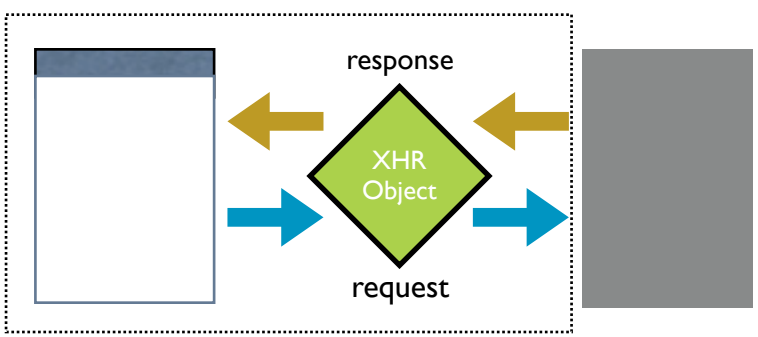
pattern. Shareable Object.



So What Can XHR Do?

- Make a request
- Return a response

- And do it asynchronously



Ajax Pattern DNA

- Every Ajax design pattern consists of a
  - **Trigger** (event or timer)
  - **Operation** (ajax)
  - **Update** (dom)

Trigger

- Every pattern of interaction starts with
  - a user event
  - a timer event

mouseout hover keypress keydown mousedown drop filter choices mouseup drag click mousedown select focus blur resize move timeout

Operation

Five basic operations define what you can do

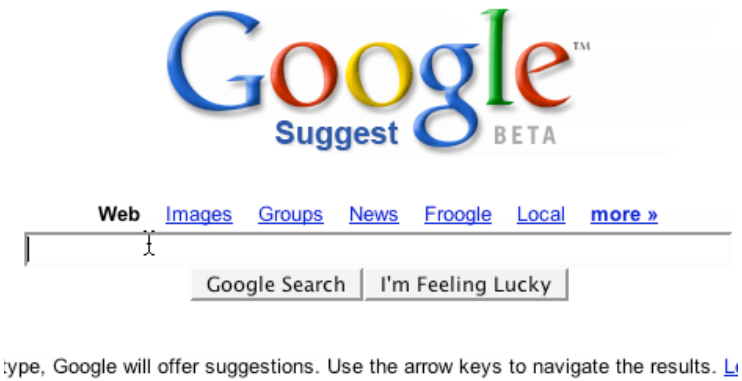
- **Lookup** I can get information when I need it
- **Persist** I can save in real-time
- **Validate** I can prevent errors early
- **Invoke** I can make things happen now
- **Message** I can communicate instantly

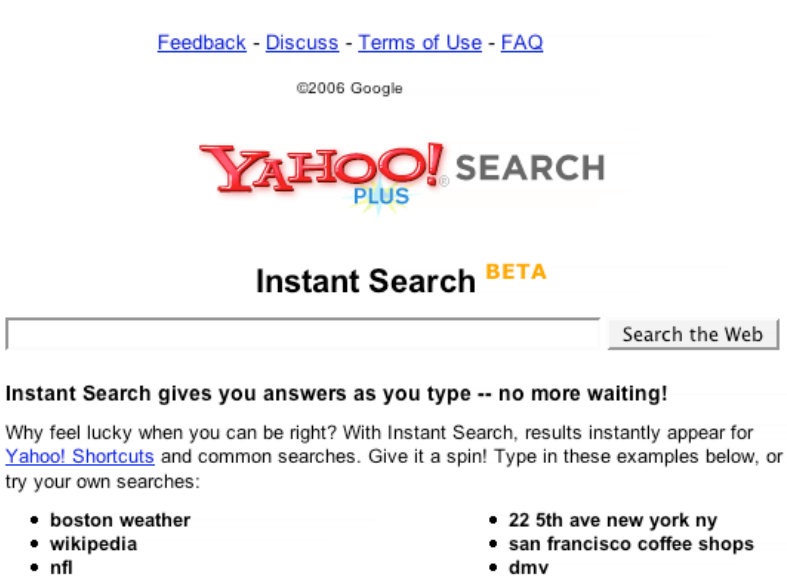
Operation. Lookup

I can get information when I need it

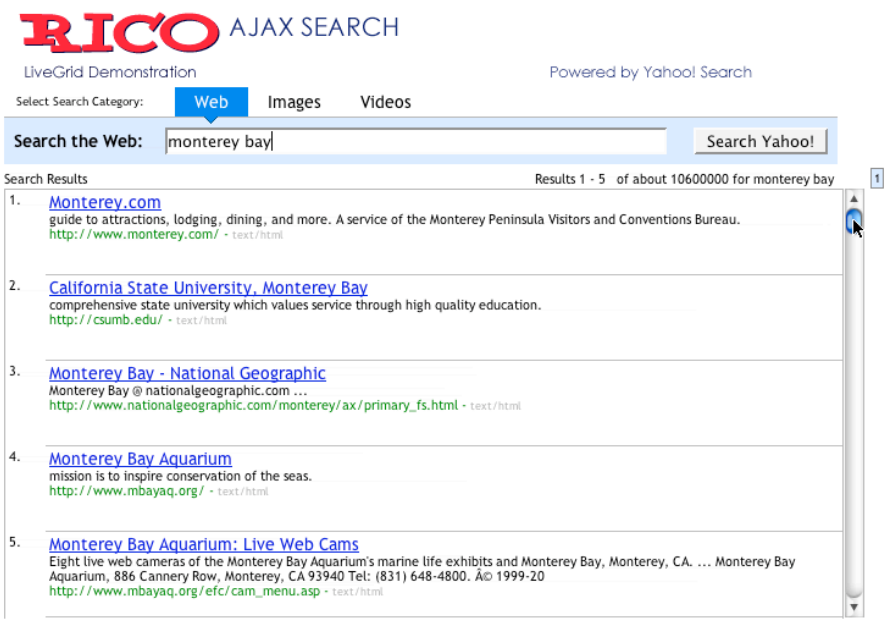
Auto Complete. Deferred Content Loading. Dynamic Goal. Narrowing Choices. Refining Search. Live Search. Dynamic Filter. Detail Zoom. Endless Scrolling. Expandable Paging Boundary. Fresh Content. Hover Detail. In Place Drill Down. On Demand Refresh. Periodic Refresh. Resizable Modules. Scrolling Modules. Auto Form Fill.

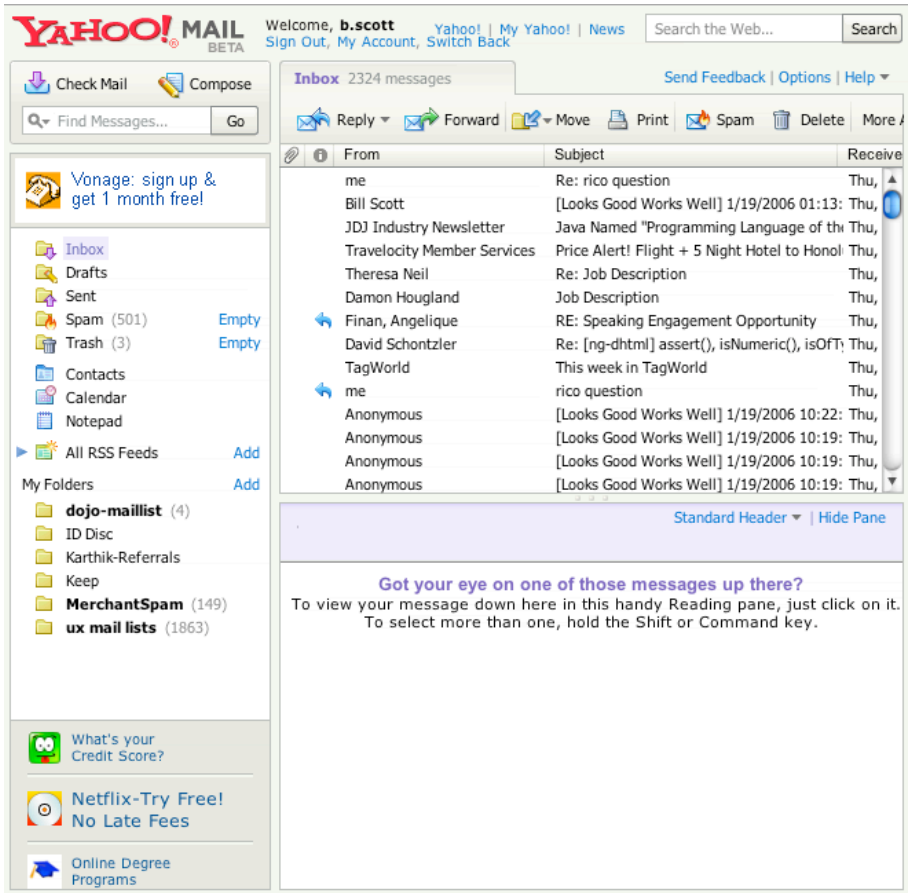
Lookup live search





Rico LiveGrid



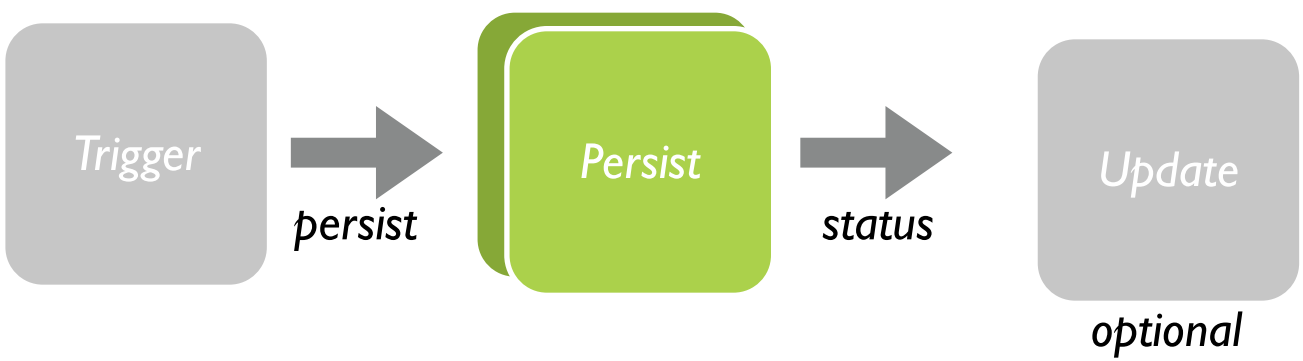


Yahoo! Mail Beta (Oddpost)

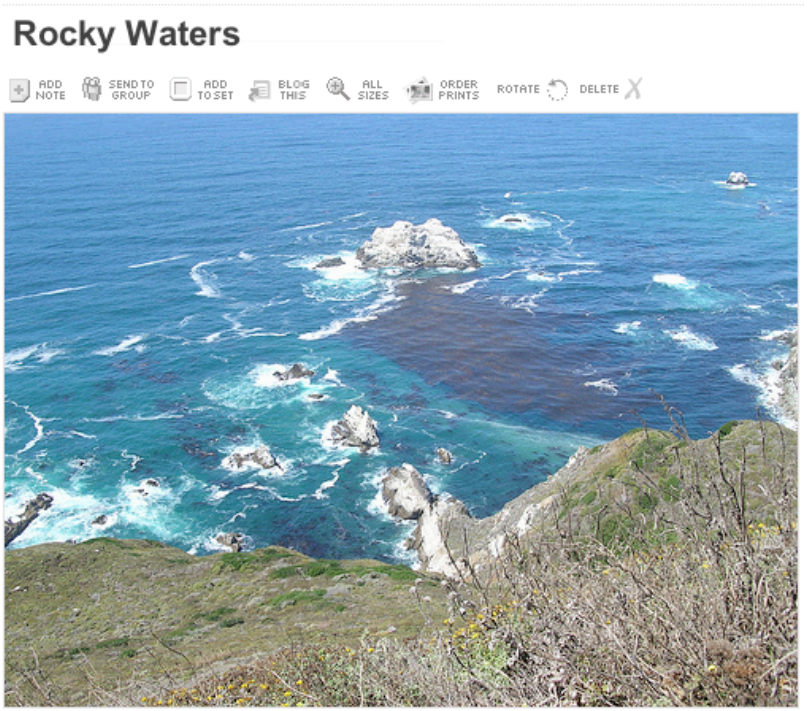
Operation. Persist

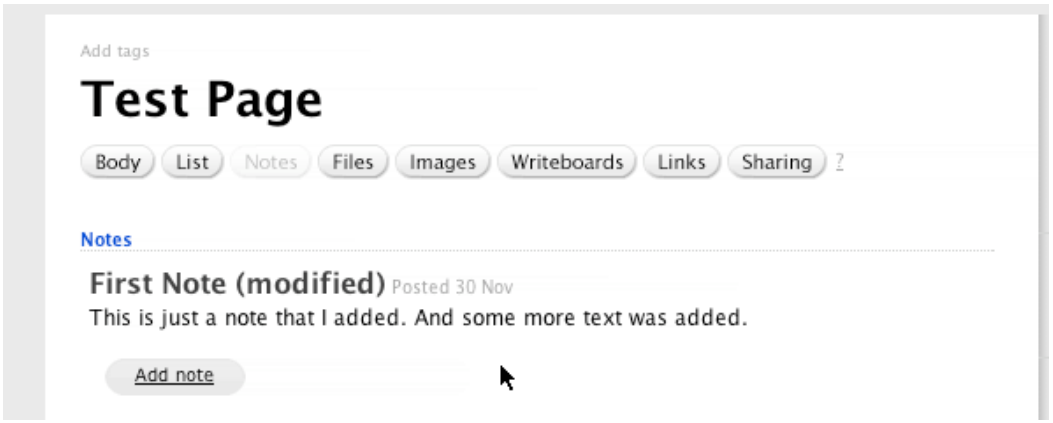
I can save in real-time

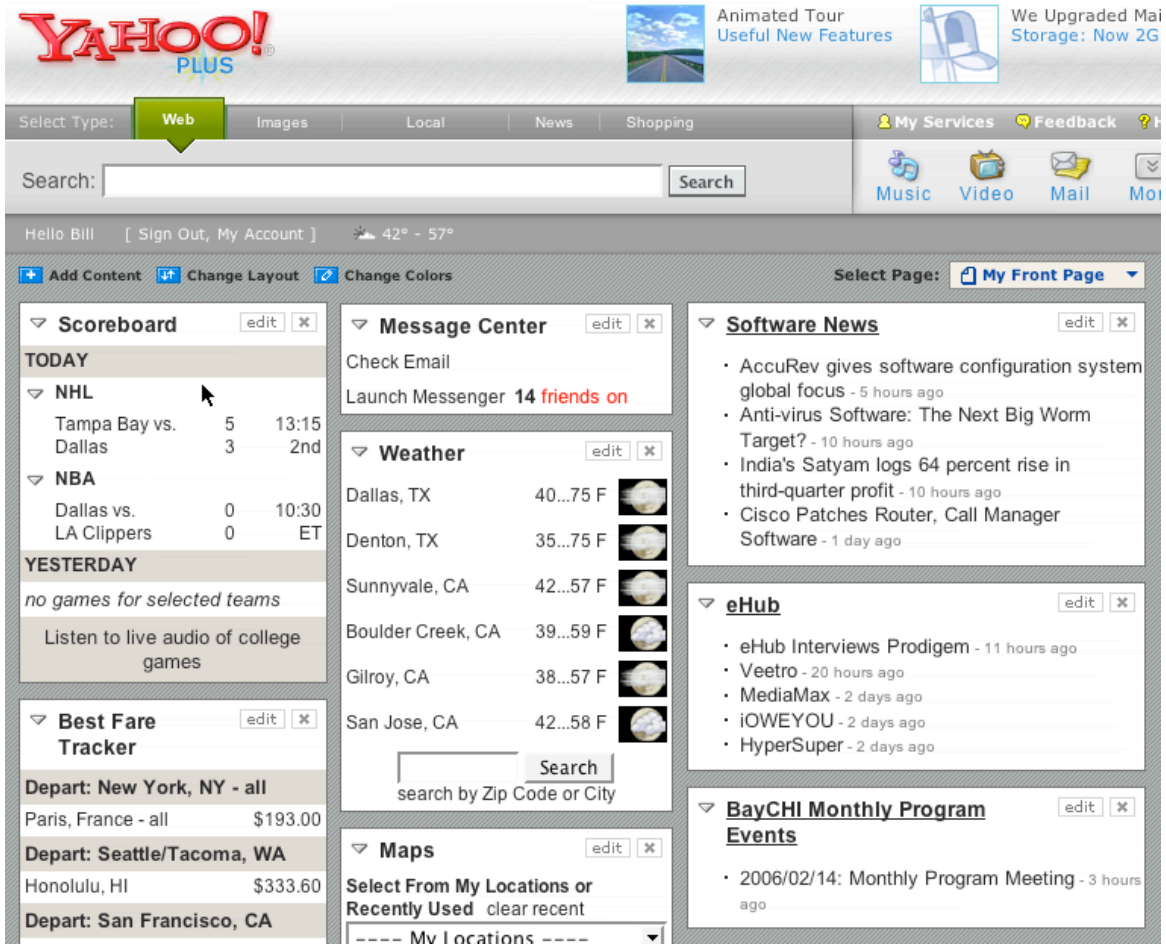
Drag and Drop Modules. In Page Editing. In Page Custom Editing. Direct State Editing. Grid Cell Editing. Inline Custom Editing. Inline Tag Editing. Popup Custom Editing. Slide-out Custom Editing. Inline Text Editing. Inline Reordering. Auto Save. Remembered Collection. Remembered Preferences. Rating an Object.



Persist inline edit






YAHOO! PLUS
Animated Tour
Useful New Features
We Upgraded Mai
Storage: Now 2G
Select Type: Web Images Local News Shopping
My Services Feedback
Search: Search
Music Video Mail
Hello Bill [Sign Out, My Account] 42° - 57°
Add Content Change Layout Change Colors
Select Page: My Front Page
Scoreboard edit
TODAY
NHL
Tampa Bay vs. 5 13:15
Dallas 3 2nd
NBA
Dallas vs. 0 10:30
LA Clippers 0 ET
YESTERDAY
no games for selected teams
Listen to live audio of college games
Best Fare Tracker edit
Depart: New York, NY - all
Paris, France - all $193.00
Depart: Seattle/Tacoma, WA
Honolulu, HI $333.60
Depart: San Francisco, CA
Message Center edit
Check Email
Launch Messenger 14 friends on
Weather edit
Dallas, TX 40...75 F
Denton, TX 35...75 F
Sunnyvale, CA 42...57 F
Boulder Creek, CA 39...59 F
Gilroy, CA 38...57 F
San Jose, CA 42...58 F
Search
search by Zip Code or City
Maps edit
Select From My Locations or Recently Used clear recent
Software News edit
AccuRev gives software configuration system global focus - 5 hours ago
Anti-virus Software: The Next Big Worm Target? - 10 hours ago
India's Satyam logs 64 percent rise in third-quarter profit - 10 hours ago
Cisco Patches Router, Call Manager Software - 1 day ago
eHub edit
eHub Interviews Prodigem - 11 hours ago
Veetro - 20 hours ago
MediaMax - 2 days ago
iOWEYOU - 2 days ago
HyperSuper - 2 days ago
BayCHI Monthly Program Events edit
2006/02/14: Monthly Program Meeting - 3 hours ago

Drop Module
persist
Persist
status
Drop Accepted

Operation. Validate

I can prevent errors early

per field validate keystroke validate
inline form validate invalid but suggest

Operation. Invoke

I can make things happen now

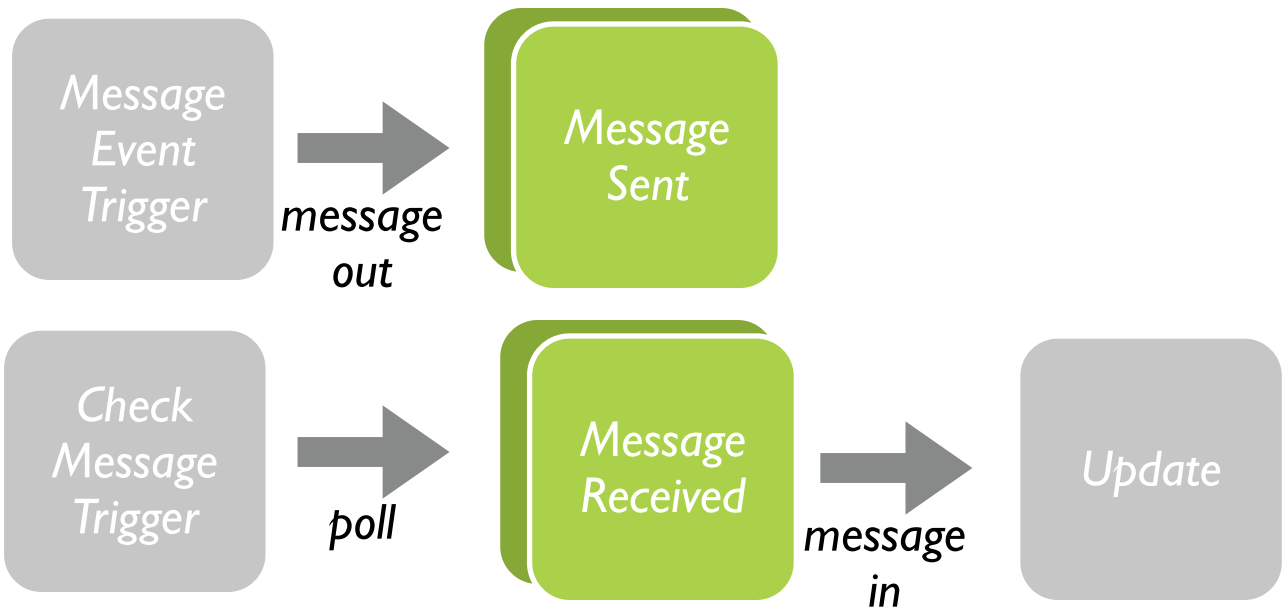
fire and forget lightweight action

Operation. Message

I can communicate instantly

two-way chat real-time auction

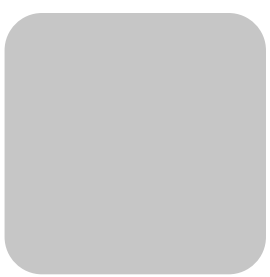
collaborative white board



Update

- User interface changes to reflect what is happening

slideout slideover dropdown popup balloon
busyicon mousecursor fade dim brighten selfhealing additems
finishedindicator remove show hide resize



Five Operations Define Ajax

- **Lookup.** I can get information when I need it
- **Persist.** I can save in real-time
- **Validate.** I can prevent errors early
- **Invoke.** I can make things happen now
- **Message.** I can communicate instantly

Steps to Understanding Ajax

- Use the **patterns**
- Understand XHR - **operations**
- Understand event management - **triggers**
- Know the tricks of the DOM - **update**
- Understand the **issues** that surround Ajax
  - *We will discuss these issues this afternoon*

Pattern-O-Matic

Ajax 101

A Look Under the Hood

adaptive path

Bill W. Scott, Y! Ajax Evangelist

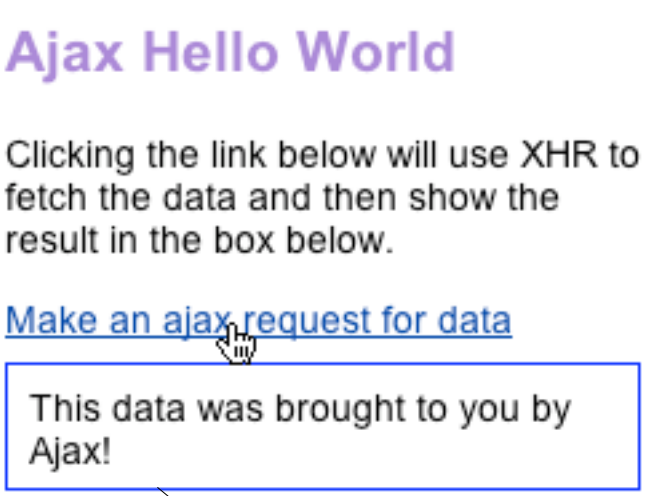
b.scott@yahoo.com

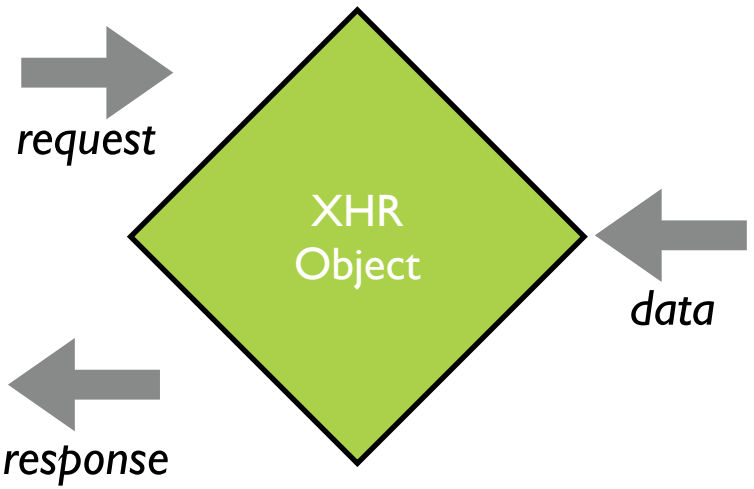
Operation. Using XHR

Operation. Using XHR

- The five operations are not built into XHR
- The simple send/response mechanism can be used to implement lookup, persist, validate, invoke and message
- To create these operations, one must understand how to use XHR
- A simple HelloWorld example will illustrate using XHR

Simple Ajax 'Hello World'






```
<?xml version="1.0" ?>
<root>
This data was brought
to you by Ajax!
</root>
```

/response.xml

```
<h2>Ajax Hello World</h2>
<p>Clicking the link below will use XHR to fetch the data and then show the
result in the box below.</p>

<span class="ajaxlink" onclick="makeRequest('test.xml')">
Make an ajax request for data
</span>

<div id="helloArea"></div>
```

/index.html

- Clicking the link makes an XHR request
- Response is inserted into the area outlined in blue

Ajax How To

1. Create a request object
2. Write a callback
3. Make the request
4. Parse the response

```
if browser is mozilla or safari or opera then
    create a new XMLHttpRequest

otherwise it is IE then
    create a new ActiveXObject
otherwise
    error - browser does not support XMLHttpRequest
```

- IE5, 5.5, 6 implements XHR as an ActiveX object (Msxm12.XMLHTTP/Microsoft.XMLHTTP)
- Mozilla 1.0+, Safari 1.2+, Opera 8+, IE7 provide XMLHttpRequest object natively
- All XHR objects have same methods & properties

XHR Methods

Method	Description
open("method", "url", [, asynchFlag [, "username" [, "password"]]])	Sets up the request object for sending a request
send(content)	Sends the request to the server. Can be null.
abort()	Stops the request.
getAllResponseHeaders()	Returns all response headers for the HTTP request as key/value pairs.
getReponseHeader("header")	Returns the string value of the specified header.
setRequestHeader("header", "value")	Sets the specified header to the supplied value.

Source: Foundations of Ajax - APress

XHR Properties

Property	Description
onreadystatechange	The event handler that fires at every state change.
readystate	The state of the request: 0=uninitialized, 1=loading, 2=loaded, 3=interactive, 4=complete
responseText	The response from the server as a text string
responseXML	The response from the server as an XML document
status	The HTTP status code from the server for the request object: 200: Ok; 201: Created; 400: bad request, 403: Forbidden; 500: internal sever error
statusText	The text version of the HTTP status code

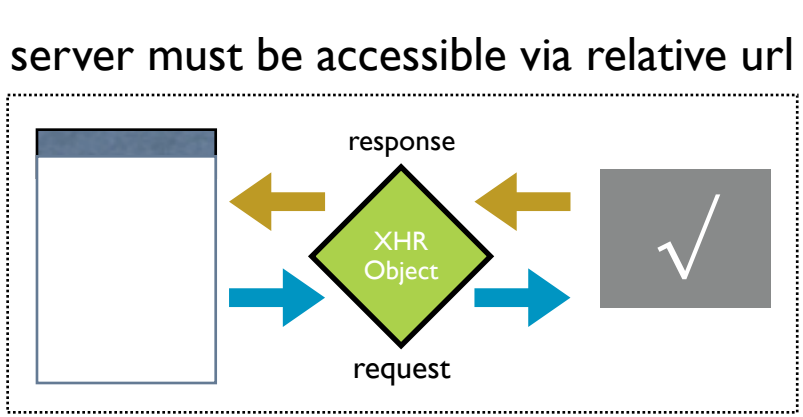
```
function handleAjaxResponse
begin
   do something with the data that is returned from XHR
end
```

- JavaScript callback function is invoked when the readystate changes on the XHR object

Ajax XHR

```
set onreadystatechange to callback function - handleAjaxResponse

open a request on the xhr object

send the request through the xhr object
```

- The JavaScript function *handleAjaxResponse* will be invoked when the readystate property changes on the XHR object
- Same site rule
- 'GET' or 'POST'
- Asynchronous flag



X

X

- Use **GET** for
  - For retrieve
  - REST services
  - When passing parameters
  - Idempotent URLs
  - Small amount of data
- Use **POST** for
  - Modification
  - Large amounts of data passed to server
  - Non-idempotent URLs

- Its just a normal HTTPRequest
  - Normal mechanism for getting request parameters
- Raw POST (xhr.send(someData))
  - Java/JSP: request.getInputStream() - read as raw post
  - Rails: @request.raw_post
  - PHP: $data = file_get_contents('php://input')

```
function handleAjaxResponse
begin
    if response is valid then
        get responseXML
        get rootNode
        get helloArea on the page
        stuff the rootNode value into the helloArea DIV
    endif
end
```

- **readystate** values
  - 0 – Uninitialized
  - 1 – Loading
  - 2 – Loaded
  - 3 – Interactive
  - 4 – Completed

- XML version must be first line

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

- Set up response header's Content-type

```
"Content-Type", "text/xml"
```

- Use well-formed XML

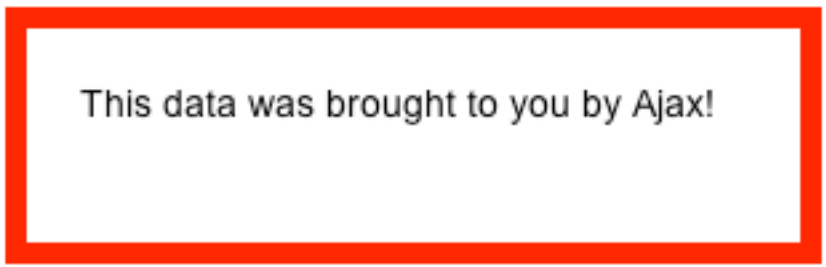
- Use XHR property **responseXML** to get the response as an XML DOM (XmlDocument)
- Use standard JavaScript DOM methods
  - Mozilla, Safari, Opera & IE support a common set of methods and properties
  - Watch out for IE only stuff (e.g., **children** property)
  - Watch out for whitespace issues
- Other options
  - Use XML library for JS (e.g., XML for <script>)

Property/Method	Description
documentElement	Returns the root element of the document
firstChild	Is the first element within another element (the first child of the current node)
lastChild	Is the last element within another element (the last child of the current node)
nextSibling	Is the next element in the same nested level as the current one
previousSibling	Is the previous element in the same nested level as the current one
nodeValue	The value of a document element
getElementsByTagName	Used to place all elements into an object ????????

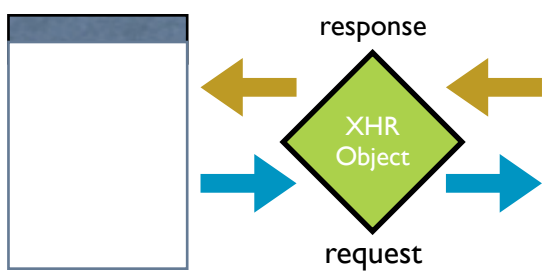
4. Parse the Response: A little more parsing.

```
<?xml version="1.0" ?>
<root>
   <contents>
    This data was brought to you by Ajax!
   </contents>
   <style>
   height:40px;width:250px;margin-top:20px;padding:20px;border:8px solid red;
   </style>
</root>
```

```
function alertContents() {

     if (xhr.readyState == 4) {
          if (xhr.status == 200) {
               var response = xhr.responseXML.documentElement;
               var content =
                 response.getElementsByTagName('contents')[0].firstChild.data;
               var style   =
                 response.getElementsByTagName('style')[0].firstChild.data;
               var helloArea = document.getElementById("helloArea");
               helloArea.innerHTML=content;
               helloArea.style.cssText=style;
          } else {
               alert('There was a problem with the request.');
          }
     }
}
```

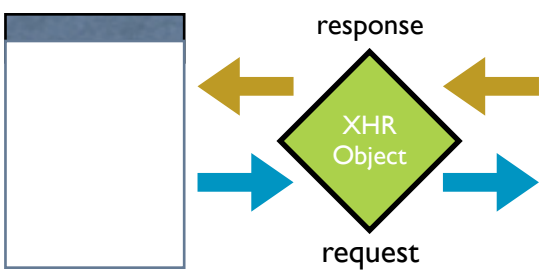


- Two types: text, xml
- Many formats



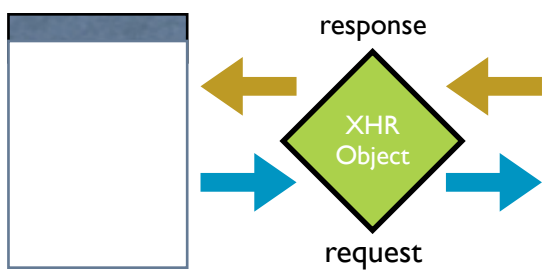
text	xml
data	data
html	html
JS	JS
json	

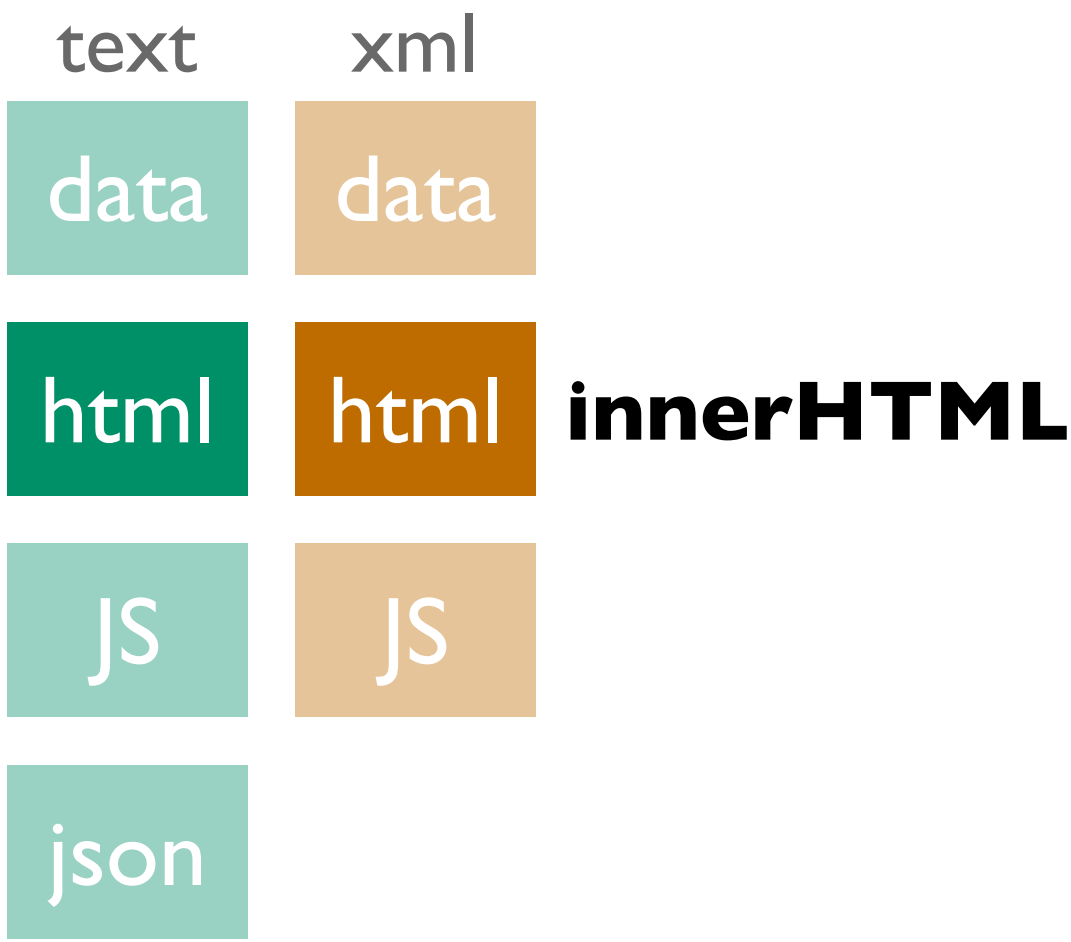
- JavaScript generated on server-side
- JavaScript is eval'ed on client side to generate GUI
- Dangerous!



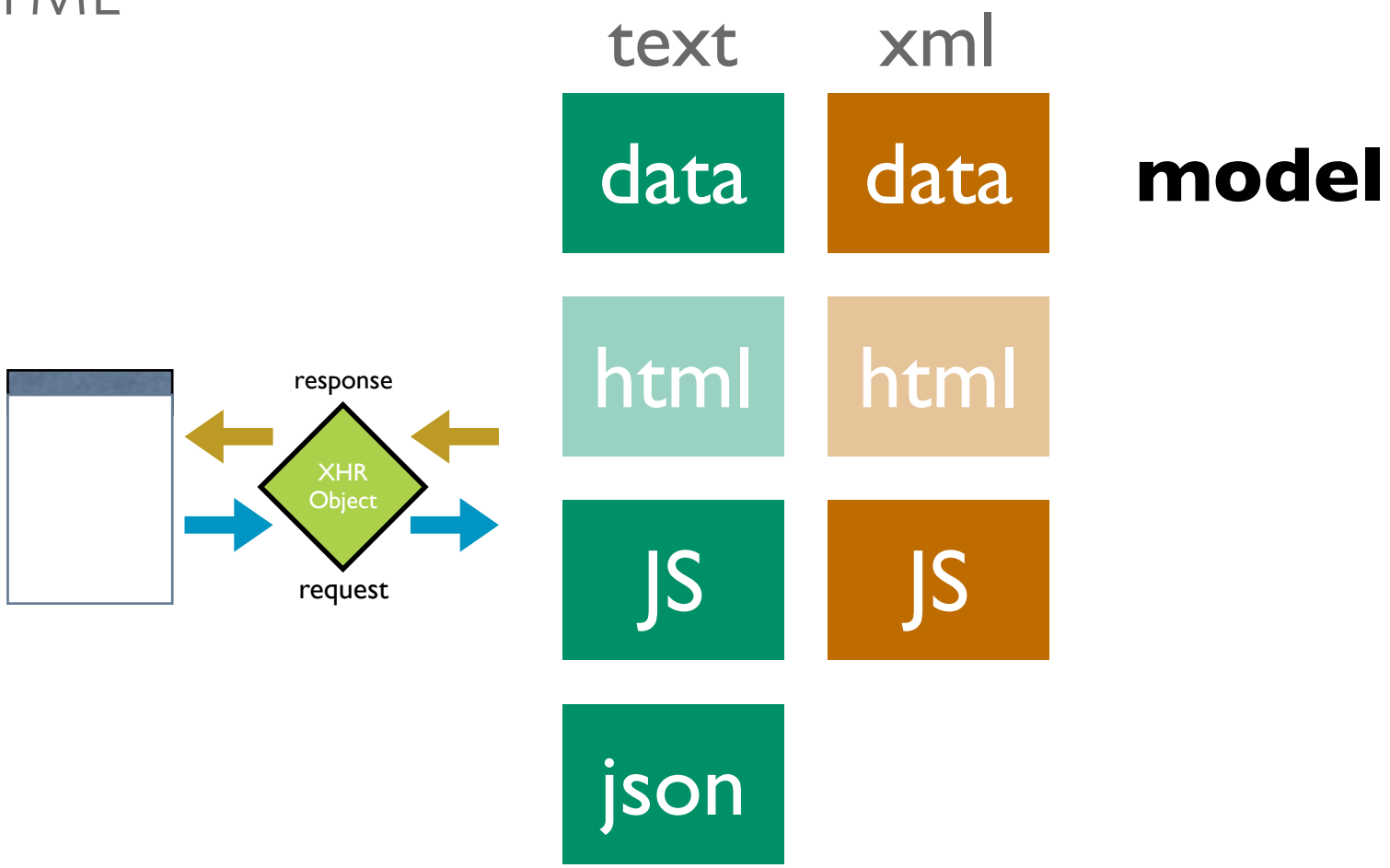
text	xml	
data	data	
html	html	
JS	JS	**interface code**
json		

- HTML generated on server side
- Stuffed into innerHTML of DOM element





- Model data generated on server side
  - w3c DOM elements created from model data
  - Substitute model data into HTML strings and insert via innerHTML



JSON

- JavaScript supports several string based notations
  - Allows the basic types to be represented as string literals
  - Strings are easy to pass over the wire
- JSON (JavaScript Object Notation - json.org)

{"**name**": "Jack B. Nimble", "at large": true, "grade": "A", "level": 3}

name	Jack B. Nimble
at large	true
grade	A
level	3

["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]

array of 7 named days

[

[0, -1, 0],

[1, 0, 0],

[0, 0, 1]

]

3x3 array

JSON Response: JSON is JavaScript

- JSON's simple values are the same as used in JavaScript
- No restructuring is requested: JSON's structures are JavaScript!
- JSON's object is the JavaScript object
- JSON's array is the JavaScript array

- Parsing is simple, native

- Obtain responseText
- Parse the responseText

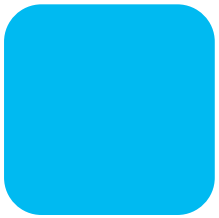
```
responseData = eval('(' + responseText + ')');
   OR
responseData = JSON.parse(responseText);
```

JSON Response: Y! JSON API

Ajax XHR

- http://api.search.yahoo.com/WebSearchService/V1/webSearch?appid=YahooDemo&query=finances&start=1&results=1&output=json


```
{
 "ResultSet":
 {
    "totalResultsAvailable":"69200000",
    "totalResultsReturned":"1",
    "firstResultPosition":"1",
    "Result":
    {
       "Title":"Yahoo! Finance",
       "Summary":"manage the market and your money with Yahoo! Finance. Includes
       stock market quotes, business news, mutual funds, online bill pay, banking
       tools, loans, insurance, retirement planning, and tax tips and advice.",
       "Url":"http:\/\/finance.yahoo.com\/",
       "ClickUrl":"http:\/\/finance.yahoo.com\/",
       "ModificationDate":"1137225600",
       "MimeType":"text\/html"
    }
 }
}
```



Trigger. JavaScript Events

Trigger. JavaScript Events

- Ajax interactions are kicked off by event & timer triggers
- There are issues with event management within the browsers
- You **do** need to understand these implications to write good Ajax applications

First, the Events

onAbort	onBlur
onChange	onClick
onDblClick	onDragDrop
onError	onFocus
onKeyDown	onKeyPress
onKeyUp	onLoad
onMouseDown	onMouseMove
onMouseOut	onMouseOver
onMouseUp	onMove
onReset	onResize
onSelect	onSubmit
onUnload	

Problem in a Nutshell

- Different event models
- Timing Issues
- Confusing madness around the **this** pointer
- Browser incompatibilities
- Memory leak issues

- Element Attributes
  - <element onclick="func()" >
- Element Properties (function reference)
  - element.onclick=func;
- Event Binding
  - w3c: addEventListener, removeEventListener
  - IE: attachEvent, detachEvent

- Element Attributes

```
<a href="#" onclick="clickHandler(this)">
function clickHandler(anchorDOMElem) {
   // this == window --> window owns the function
   // anchorDOMElem was set to anchor DOM element
   this = anchorDOMElem; // fix this pointer
}
```

- Element Properties

```
myAnchor.onclick=clickHandler;
function clickHandler() {
   //this == anchorDOMElem --> anchorDOMElem owns the function
}
```

- Event Binding

```
function AnchorLink(anchorDOMElem) {
   this.anchorDOMElem = anchorDOMElem;
   this.anchorDOMElem.onclick = this.clickHandler;
   this.anchorDOMElem.anchorLinkObj = this;
}
AnchorLink.prototype.clickHandler = function() {
   // this == anchorDOMElem, not AnchorLink object
   // confusing since this normally refers to AnchorLink
   // grab our normal this
   anchorLinkObj = this.anchorLinkObj;
}
```

- Supports arbitrary number of event handlers
- Provides a way to remove events
- The model defines capture & bubble flow
- But, no way to get list of handlers
- And remember the browsers play differently:

Internet Explorer	Mozilla (FF), Safari, Opera [W3C]
attachEvent() detachEvent()	addEventListener() removeEventListener()
this == window object	this == DOM event object

- Attempting to attach events before the page is loaded will cause problems
- Do it on the onload

- IE's garbage collection does simple reference counting
- Attaching events and not removing them when finished will cause memory leaks
- Can use an Observer style pattern to cache event handlers and at the end clean them up

- Standardizes event binding
  - Attach an event

```
var oElement = document.getElementById("elementid");
function fnCallback(e) { alert("click"); }
YAHOO.util.Event.addListener(oElement, "click", fnCallback);
```

  - Attach multiple events

```
var ids = ["el1", "el2", "el3"]; // array can contain object references, element ids, or both
function fnCallback(e) { alert(this.id); }
YAHOO.util.Event.addListener(ids, "click", fnCallback);
```

- Handles automatic scope correction
  - By default "this" refers to the DOM element that the event was attached to
  - Can be overridden
- First parameter is the event object
- Pass arbitrary objects to event handler

- Automatic Listener Cleanup
- Custom Events
  - For custom objects or overriding events on widgets
- onAvailable event
  - Define a function that will execute as soon as an element is detected in the DOM (during load)

```
<script type="text/javascript">
function TestObj(id) { YAHOO.util.Event.onAvailable(id, this.handleOnAvailable, this); }
TestObj.prototype.handleOnAvailable = function(me) { alert(this.id + " is available"); }
var obj = new TestObj("myelementid"); </script>
<div id="myelementid">my element</div>
```

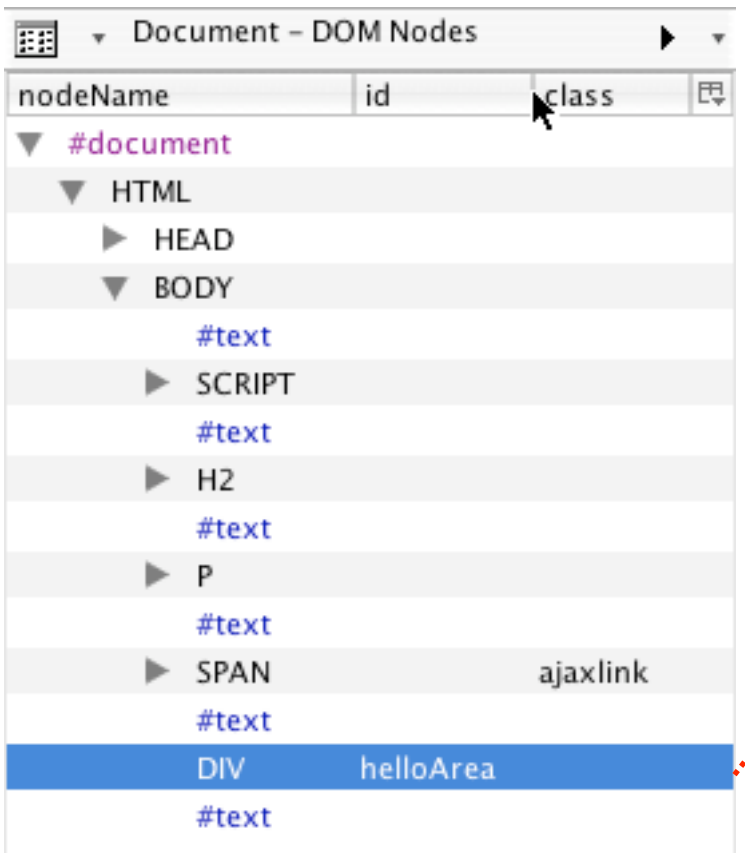
Update. The DOM

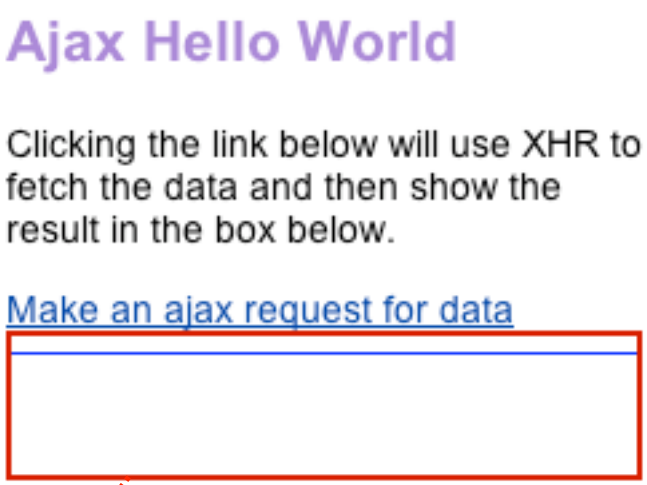
Update. The DOM

- Browsers represent the user interface as a set of objects (or elements)
- Since a page has a hierarchy of containment, each of these elements are related as parent-child or siblings
- This takes the shape of a tree
- The tree of elements is called the *document object model* or DOM. Specifically the Browser DOM.
- Any change to the DOM structure or a DOM element is reflected immediately on the web page

DOM Example

- Represented as a tree of nodes





Using the DOM

- JavaScript is the DOM manipulator
- Use it to
  - Find DOM elements
  - Add new elements
  - Remove elements
  - Modify element attributes
    - Position elements
    - Style elements

Finding DOM Elements

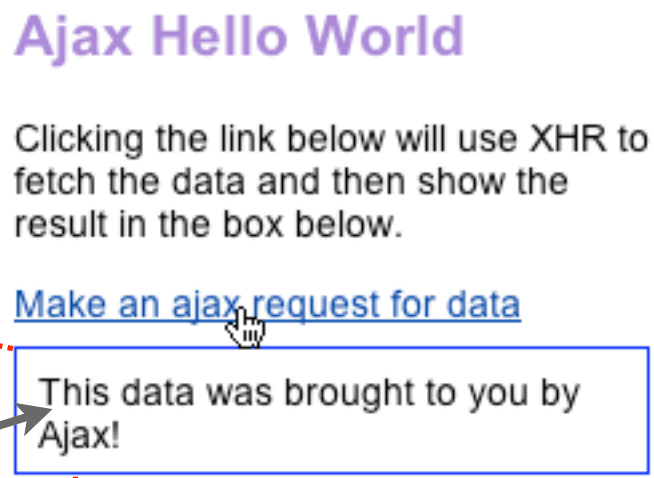
- document.getElementById
  - Prototype library shortcut: $("idName") or $(id)
  - Yahoo! library shortcuts:
    - YAHOO.util.Dom.get("idName")
    - getElementByClassName
- parentNode
- childNodes
  - Yahoo! library has ways to find ancestors

DOM Manipulation

- Creating new interface elements
  - innerHTML, createElement(), createTextNode(), appendChild()
- Changing element styles
  - Visual attributes
  - Geometry
- Y!UI positioning, region detection
- Y!UI classname & style normalization

Example of DOM Manipulation (Hello World)

```
<h2>Ajax Hello World</h2>
<p>Clicking the link below will use XHR to fetch the data and
then show the result in the box below.</p>

<span class="ajaxlink" onclick="makeRequest('response.jsp')">
Make an ajax request for data
</span>

<div id="helloArea"></div>
```




```
var helloArea = document.getElementById("helloArea");
helloArea.innerHTML=rootNode.firstChild.data;
```

Keeping it Clean

- Separate presentation style from content with CSS
  - Supports degradability
  - Supports accessibility
  - Simplifies maintenance
- http://www.mezzoblue.com/css/cribsheet/
-

DHTML DOM

- Avoid these elements:
  - b, big, hr, i, small, sub, sup, tt
  - basefont, center, dir, font, isindex, menu, s, strike, u, tfoot
  - br is ok for content breaks; not layout
- Make your HTML meaningful
  - List structures as ul, li
  - Tree structures as nested ul, li
  - Containers as div
- Use css selectors (contextual)
  - Reduces the number of classes
  - Avoid inline styles

Semantic Markup Example

DHTML DOM

```
<div id="weather">
 <div id="current">
   <div id="currentHeader" class="accordionTabTitleBar">
     Current Conditions
   </div>
   <div class="weatherTabContentBox">
     <div class="weatherPanel" id="ccInfo">
     </div>
   </div>
 </div>

 <div id="moon">
   <div id="moonHeader" class="accordionTabTitleBar">
     Moon
   </div>
   <div class="weatherTabContentBox">
     <div class="weatherPanel" id="moonInfo">
     </div>
   </div>
 </div>

 <div id="sun">
   <div id="sunHeader"  class="accordionTabTitleBar">
     Sunlight Summary
   </div>
     <div class="weatherTabContentBox">
     <div class="weatherPanel" id="sunInfo">
     </div>
   </div>
 </div>
</div>
```

```
new Rico.Accordion( 'weather', {panelHeight:220, expandedBg:'#39497b'}
```

Keeping it clean: Example

DHTML DOM

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head><title>Make this page more semantic</title></head>
<body>
<div class="page-title">Make this page more semantic</div>
<div>
<div class="module-title">Module Title</div>

<span class="time"><i>12:40pm Wed Mar 29</span></i>
<div class="para">Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci </
div>
</div>
<p><div><i>ìWeb users ultimately want to get at data quickly and easily. They don't care as much about
attractive sites and pretty design.î</i></div> says <b>Tim Berner's Lee</b>, inventor of the World Wide Web,
although I'm sure many visual designers would beg to differ.  <u>"The web isn't print!"</u> - Chanel.</p>


<table>
   <tr>
      <td class="heading">column 1</td>
      <td class="heading">column 2</td>
      <td class="heading">column 3</td>
   </tr>
      <td class="data">data cell 1</td>
      <td class="data">data cell 2</td>
      <td class="data">data cell 3</td>
   </tr>
</table>
</body>
</html>
```

Keeping it clean: Example

DHTML DOM

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head><title>Make this page more semantic</title></head>
<body>
<h1>Make this page more semantic</h1>

<h2>Module Title</h2>

<em class="time" title="time stamp">12:40pm Wed Mar 29</em>
<blockquote>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt
ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci </blockquote>
<blockquote cite="Tim Berner's Lee">

<p>&#39;Web users ultimately want to get at data quickly and easily. They don't care as much about attractive
sites and pretty design.&#39;</p></blockquote> says <cite>Tim Berners Lee</cite>, inventor of the World Wide
Web, although I'm sure many visual designers would beg to differ.  <q>The web isn't print!</q> -
<cite>Chanel</cite>.

<table summary="this month's numbers">
   <caption>Q1 Financials</caption>
   <thead> <tr>
      <th scope="col">column 1</th>
      <th scope="col">column 2</th>
      <th scope="col">column 3</th>
   </tr> </thead>
   <tbody>
      <td class="data">data cell 1</td>
      <td class="data">data cell 2</td>
      <td class="data">data cell 3</td>
   </tr>
   </tbody>
</table>
</body>
</html>
```

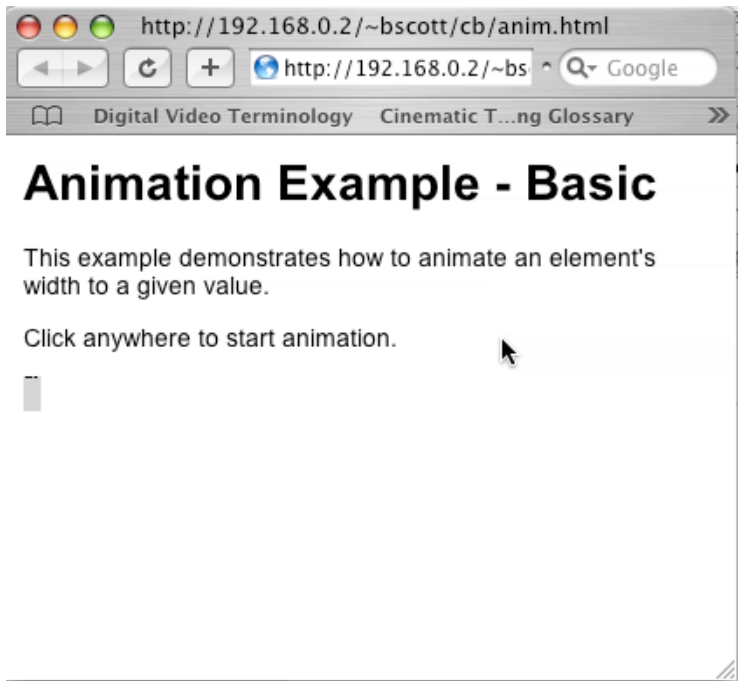
DHTML DOM

```
...
<script type="text/javascript">
YAHOO.example.DDApp = function() {
    var dd;
    return {
        init: function() {
            var dd = new YAHOO.util.DD("dragDiv");
        }
    }
} ();
YAHOO.util.Event.addListener(window, "load", YAHOO.example.DDApp.init);
</script>

<style type="text/css">
#dragDiv {
    background:url(images/macosxlogo.gif) 0 0 no-repeat;
    height:100px;
    width:70px;
}
</style>
</head>

<body>
<h2>Drag and Drop</h2>
<p>The Mac logo is draggable</p>

<div id="dragDiv"></div>

</body>
...
```

Advanced DOM Manipulation: Animation

DHTML DOM

```
<html>
<head>
<style>
#anim { background:#ccc;width:10px;height:20px;font-size:10%;}
</style>
<script type="text/javascript">
YAHOO.namespace('example.anim');
YAHOO.example.anim.init = function() {
   var myAnim = new YAHOO.util.Anim('anim', {
                        width: {to: 200},
                        height: {to: 150},
                        fontSize: {from: 20, to: 120, unit: '%'},
                        opacity: {from: 0.25, to: 1.0 }},
                        1,
                        YAHOO.util.Easing.easeOut);
   var animate = function(e) {
      myAnim.animate();
      return false;
   }
   YAHOO.util.Event.addListener(document, 'click', animate);
}
YAHOO.util.Event.addListener(window, 'load', YAHOO.example.anim.init);
</script>
</head>
<body>
<h1>Animation Example - Basic</h1>
<p>This example demonstrates how to animate an element's width to a given value.</p>
<p>Click anywhere to start animation.</p>

<div id="anim">Lorem ipsum dolor</div>

</body>
</html>
```



Learning Ajax

- Learning Ajax, means understanding
  - **Triggers** (event or timer)
  - **Operations** (ajax xhr)
  - **Updating** (dom)

Summary

Ajax Architecture

- The Ajax DNA (events, data, presentation) is just a restatement of Model-View-Controller software pattern
- It is an architectural approach that separates these three concerns
  - **Model**: keep it pure, avoid presentation artifacts
  - **View**: keep HTML clear of style. Keep style in CSS.
  - **Controller**: don't mix events in HTML markup, use observer style of event management

Pulling It All Together

- Two Projects that
  - Shows XHR in action
  - Explores XML vs JSON

Ajax & Web Services

XML and JSON Examples

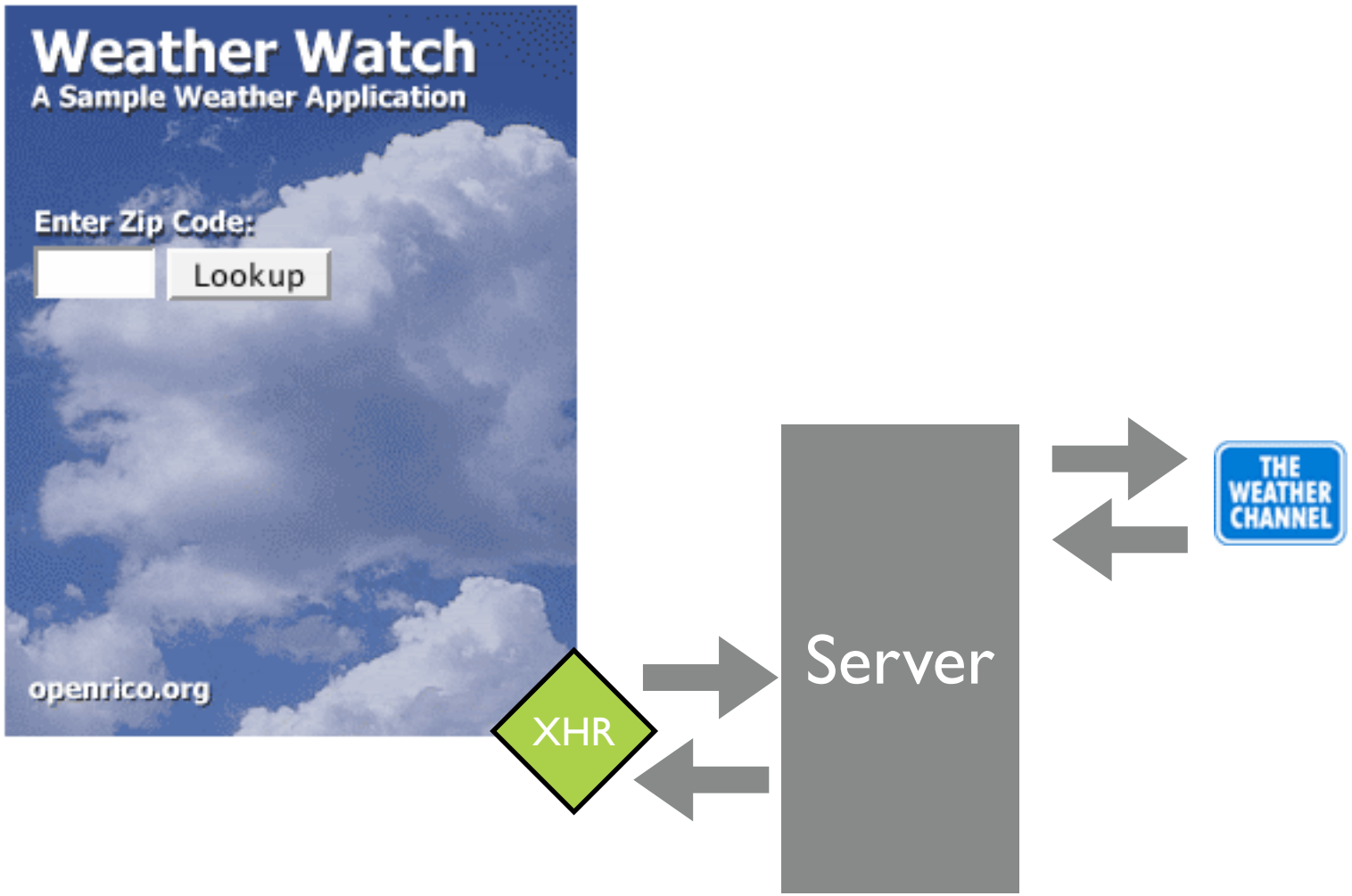
adaptive path

Bill W. Scott, Y! Ajax Evangelist

b.scott@yahoo.com

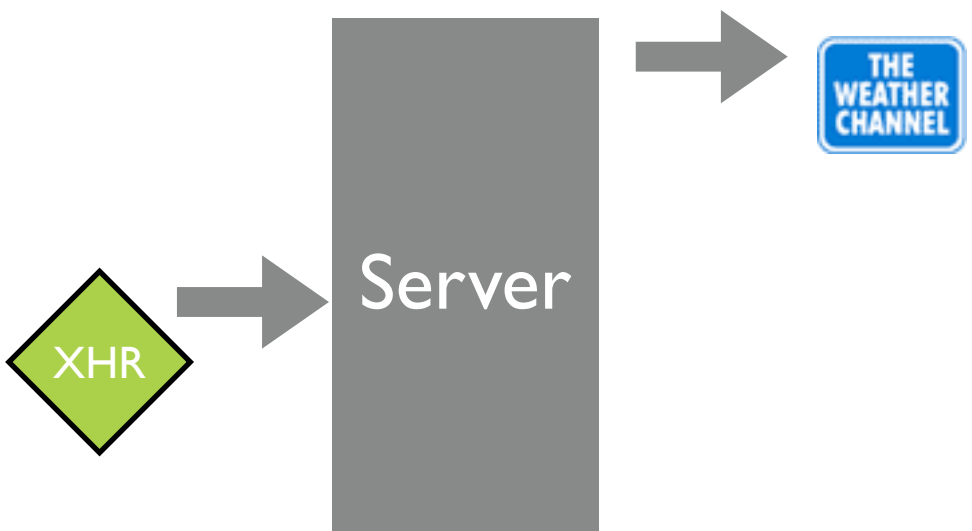
Building an Ajax Weather Widget

Using an XML Service



weather.com exposes its weather service via an URL that includes a partner key and license key

```
http://xoap.weather.com/weather/local/95132?
cc=*&link=xoap&prod=xoap&par=PARTNER_KEY&key=LICENSE_KEY
```



XML Response from weather.com

Server

```
<?xml version="1.0" encoding="ISO
-8859-1"?>
<weather ver="2.0">
  <head>
    <locale>en_US</locale>
    <form>MEDIUM</form>
    <ut>F</ut>
    <ud>mi</ud>
    <us>mph</us>
    <up>in</up>
    <ur>in</ur>
  </head>
  <loc id="95132">
    <dnam>San Jose, CA (95132)</
dnam>
    <tm>8:37 PM</tm>
    <lat>37.4</lat>
    <lon>-121.86</lon>
    <sunr>7:18 AM</sunr>
    <suns>5:20 PM</suns>
    <zone>-8</zone>
  </loc>
```

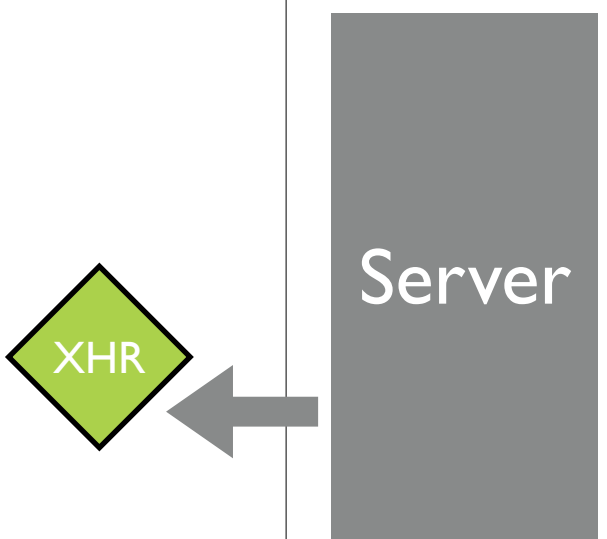
```
  <cc>
    <lsup>1/21/06 7:53 PM PST</lsup>
    <obst>San Jose, CA</obst>
    <tmp>49</tmp>
    <flik>46</flik>
    <t>Mostly Cloudy</t>
    <icon>27</icon>
    <bar>
      <r>30.27</r>
      <d>rising</d>
    </bar>
    <wind>
      <s>7</s>
      <gust>N/A</gust>
      <d>350</d>
      <t>N</t>
    </wind>
    <hmid>80</hmid>
    <vis>10.0</vis>
    <uv>
      <i>0</i>
      <t>Low</t>
    </uv>
    <dewp>43</dewp>
    <moon>
      <icon>21</icon>
      <t>Waning Gibbous</t>
    </moon>
  </cc>
</weather>
```

weather.com responds with an XML response that describes the weather's current conditions

Rico/Prototype Response

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ajax-response>
  <response type="element" id="ccInfo">
  <div class="weatherTitle">San Jose, CA (95132)</div>
  <div>
  <span><image id="ccImg" src="/images/weather/27.png"></image></span>
  <span class="weatherTemp">49&#176;F</span>
  </div>
  <div class="weatherDescr">Mostly Cloudy</div>
  <div>
  <table cellspacing="0" cellpadding="0" border="0">
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Humidity:</td>
    <td>80&#37;</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Barometer:</td>
    <td>30.27&quot;</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Wind:</td>
    <td>From N at 7 mph</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle"></td>
    <td>gusting to N/A mph</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Dewpoint:</td>
    <td>43&#176;F</td></tr>
    <tr class="weatherDetail">
    <td class="weatherDetailTitle">Heat Index:</td>
    <td>46&#176;F</td></tr>
  </table>
  </div>
  </response>
</ajax-response>
```

My server code translates the weather.com XML into a Rico/Prototype ajax-response. Notice the response is an HTML code snippet. The response is of type element, and mapped to id="ccInfo"



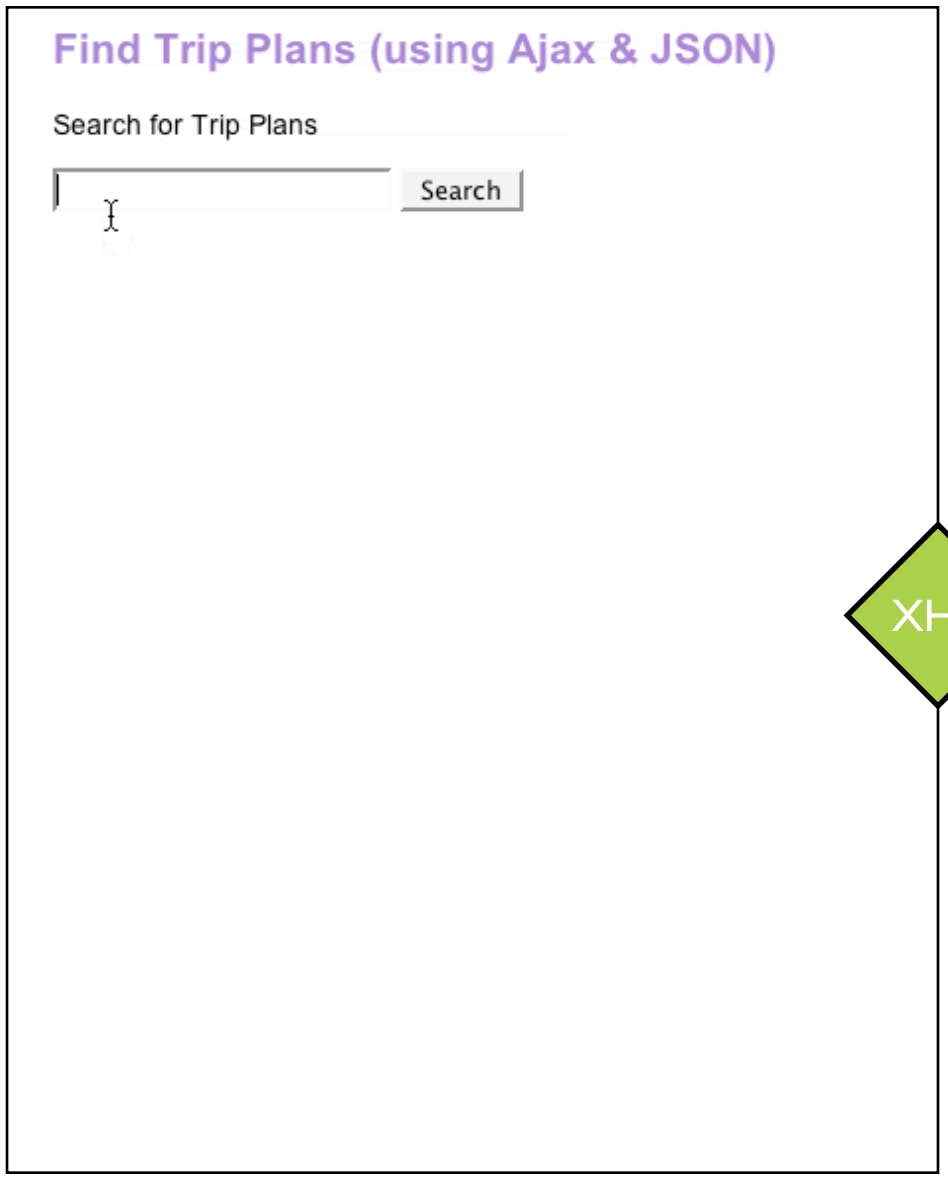
Simple DIV Structure Defines the Accordion

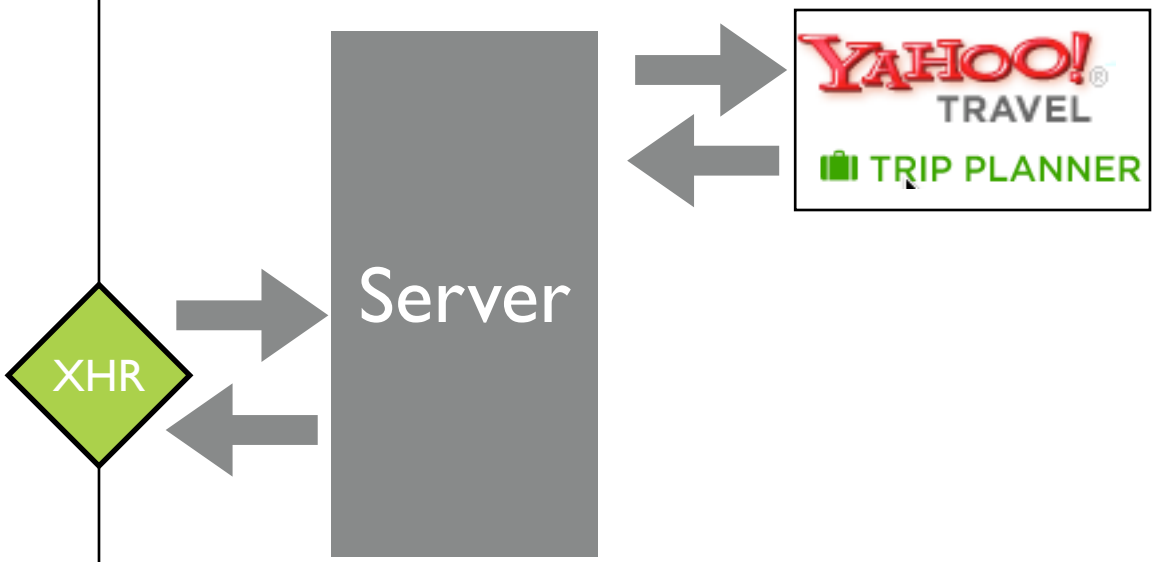
```
<div id="weather">
 <div id="current">
   <div id="currentHeader" class="accordionTabTitleBar">
     Current Conditions
   </div>
   <div class="weatherTabContentBox">
     <div class="weatherPanel" id="ccInfo">
     </div>
   </div>
 </div>

 <div id="moon">
   <div id="moonHeader" class="accordionTabTitleBar">
     Moon
   </div>
   <div class="weatherTabContentBox">
     <div class="weatherPanel" id="moonInfo">
     </div>
   </div>
 </div>

 <div id="sun">
   <div id="sunHeader"  class="accordionTabTitleBar">
     Sunlight Summary
   </div>
     <div class="weatherTabContentBox">
     <div class="weatherPanel" id="sunInfo">
     </div>
   </div>
 </div>
</div>
```

```
<script>
onloads.push( accord );
function accord() {
  new Rico.Accordion( 'weather',
      {panelHeight:220, expandedBg:'#39497b'} );
}
</script>
```

```
function registerAjaxStuff() {
   ajaxEngine.registerRequest( 'getWeatherInfo', 'ajax_weather_info' );
   ajaxEngine.registerAjaxElement( 'ccInfo' );
   ajaxEngine.registerAjaxElement( 'moonInfo' );
   ajaxEngine.registerAjaxElement( 'sunInfo' );
   ajaxEngine.registerAjaxElement( 'sponsoredLinks' );
   $('zip').onkeydown = handleEnterKey.bindAsEventListener($('zip'));
}
function getWeatherInfo() {
   $('checkanother').style.visibility='visible';
   new Rico.Effect.Position( $('zipinput'), 200, null, 100, 10);

   new Rico.Effect.FadeTo( 'frontdoor', 0, 100, 10,
         {complete:function() {$('frontdoor').style.display = 'none';}}
         );

   ajaxEngine.sendRequest( 'getWeatherInfo', "zip=" + $('zip').value);
}
function resetWeather() {
   $('zipinput').style.left = '12px';
   $('checkanother').style.visibility='hidden';

   $('frontdoor').style.display = ''
   $('zip').focus();
   new Rico.Effect.FadeTo( 'frontdoor', .99999, 100, 10, {complete:emptyContents});
}
```

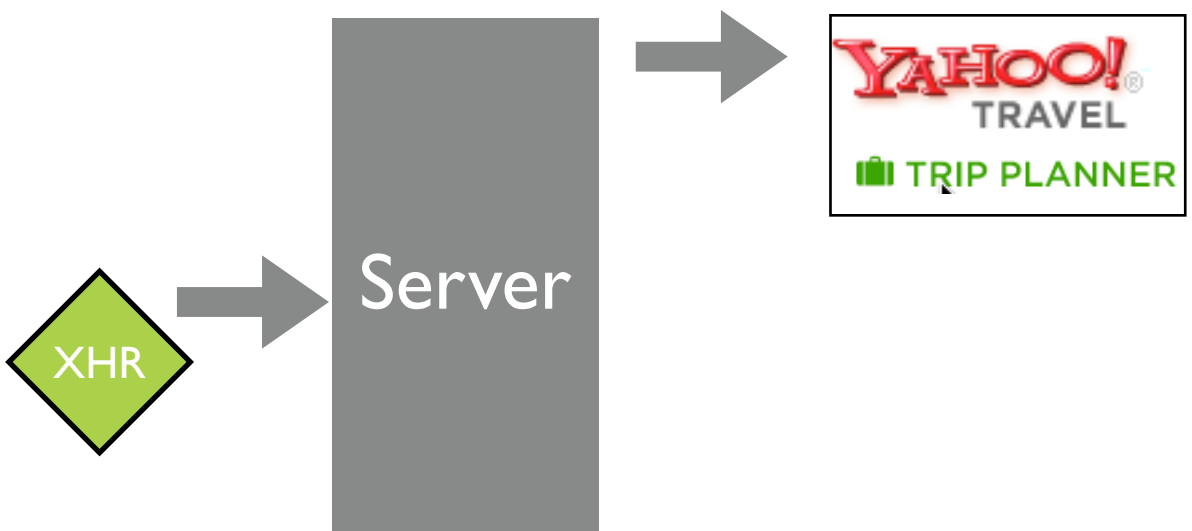
JSON Trip Finder





Yahoo! Trip Planner exposes its service via an URL that requires a free partner key. Notice output=json.

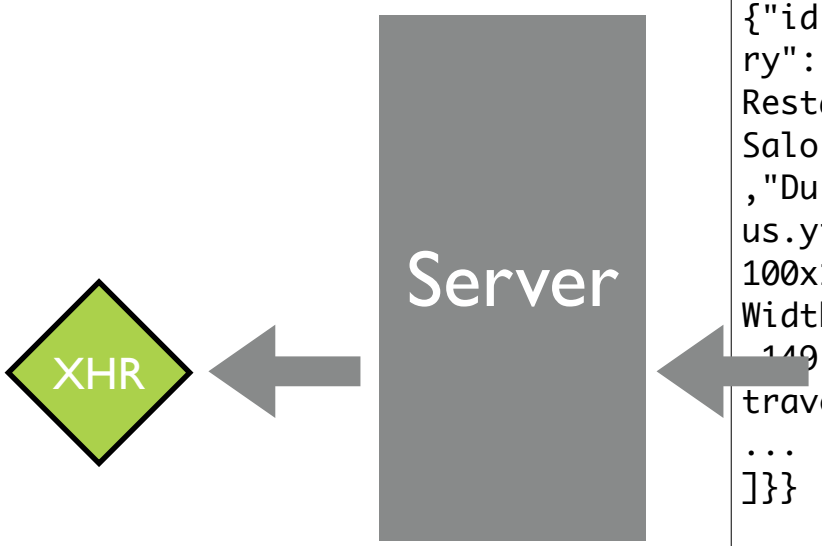
```
http://api.travel.yahoo.com/TripService/V1/tripSearch?
appid=PARTNER_KEY&query=alaska&output=json
```



JSON Response


```
{"ResultSet":{"firstResultPosition":
1,"totalResultsAvailable":"16","totalResultsReturned":
10,"Result":[

{"id":"351102","YahooID":"jk_95661","Title":"Cruise to Alaska
from Vancouver","Summary":"Things to do: Whistler Blackcomb -
Hiking,... Hotel: Hyatt Regency Vancouver, Howard
Jho...","Destinations":"Vancouver, Anchor
Vancouver, Ketchikan,
Se...","CreateDate":"1130437928","D
{"Url":"http:\/\/us.i1.yimg.com\/u
lp\/cd\/
100x100_cde24409e413d4d6da27953f6a
"Width":"66"},"Geocode":{"Latitude"
-123.120499","precision":"not availa
travel.yahoo.com\/trip\/?pid=351102

{"id":"400333","YahooID":"brdway_grl","Title":"Alaska","Summa
ry":"Things to do: Moose's Tooth Pub and Pizzer...
Restaurant: Orso, Simon's & Seafort's
Salo...","Destinations":"Anchorage","CreateDate":"1134676973"
,"Duration":"10","Image":{"Url":"http:\/\/us.i1.yimg.com\/
us.yimg.com\/i\/travel\/tg\/poi\/ca\/
100x100_ca605a44b47e30a731d3093b94afa37e.jpg","Height":"75","
Width                {"Latitude":"61.190361","Longitude":"
-149                 :"not available"},"Url":"http:\/\/
trave                ?pid=400333&action=view"},
...
]}}
```


Y! service responds with a JSON text string, valid JavaScript object notation. This is passed directly back as the XHR result



adaptive path

Designing and Building with Ajax

89

The Ajax/JSON Code

```
<script>
function showResponse() {
     if (xhr.readyState == 4) {
          if (xhr.status == 200) {
               var jsontext = xhr.responseText;
               var helloArea = document.getElementById("helloArea");
               var theTrip = eval( '(' + jsontext + ')' );
               var tripPlanHTML = "";
               for(var i=0; i<theTrip.ResultSet.totalResultsReturned; i++) {
                    var result = theTrip.ResultSet.Result[i];
                    tripPlanHTML = tripPlanHTML + '<div style="padding:4px;
border:1px solid gray;width:'+result.Image.Width+';"><img
src="'+result.Image.Url+'" width="'+result.Image.Width+'"
height="'+result.Image.Height+'"></img></div>'+
'<div ><a href="'+result.Url+'"><span style="font-weight:bold;font-size:
18px;">'+result.Title+'</span></a></div><div>'+result.Summary+'</div><br/>';
               }
               helloArea.innerHTML=tripPlanHTML;
          } else {
                alert('There was a problem with the request.');
          }
     }
}
</script>
<h2>Find Trip Plans (using Ajax &amp;amp; JSON)</h2>
<p>Search for Trip Plans</p>
<input id="searchCriteria" type="text"> <input value="Search" type="button"
onclick="makeRequest('response_trip_planner_mult_json.jsp',
          document.getElementById('searchCriteria').value)">
<div style="height:300px;width:420px;margin-top:8px;padding:8px;"
id="helloArea"></div>
```

Ajax Toolkits

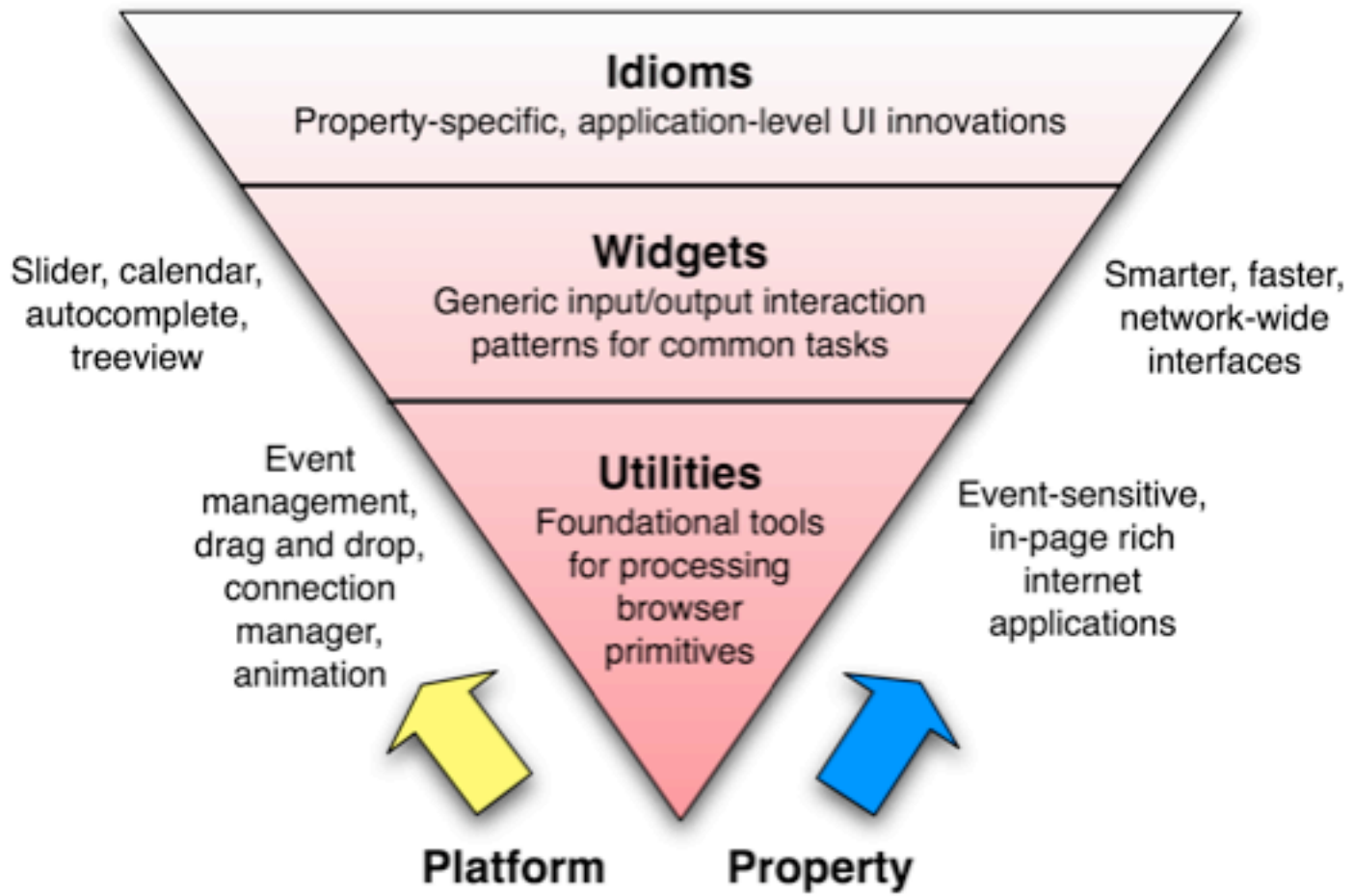
Designing and Building Ajax Applications with Ajax Toolkits

adaptive path

Bill W. Scott, Y! Ajax Evangelist

b.scott@yahoo.com

What a Toolkit Provides



Evaluating Toolkits

- What language is it targeted for?
- Are you looking for XHR abstraction or high level?
- Will you use XML or JSON?
- Is it $$ or open source?
- Does it simplify general JS development?
- Community of support?
- Likelihood it will be around next year?

Ajax Toolkits: Open Source JavaScript Libraries

Prototype	Inspired by Ruby, simplifies DOM manipulation, events, Ajax connectivity. Several frameworks based on it. Notable for the $() function among many other shortcuts. OO extends.
Scriptaculous	Built on Prototype, nice effects, animations, drag and drop, widgets. Nicely documented
Rico	Built on Prototype, nice effects, animations, behaviours (widgets), drag and drop, nice Ajax engine abstraction, style effects. Lots of demos.
Behaviour	Built on Prototype, separates event management from HTML using CSS selectors
Dojo	Widget library, drag and drop, effects, widget framework, event management, animation, bookmarkability, manipulating location history, extensive deployment support
Zimbra	Extensive set of widgets (data list, wizard, button, rich text editor, tree, menus, etc.). MVC, debugging facilities
Microsoft Atlas	In developer release. Support for drag and drop, animation, full XML specification for wiring widgets together, behavior abstraction, will be integrated in full suite of MS dev tools.
MochiKit	Widgets, painless DOM manipulation, task management, JSON-like notation, logging. Active community
Yahoo! UI	Official Yahoo! DHTML/Ajax library. Contains event mgmt, DOM manipulation, animation, drag and drop, CSS grid, CSS fonts, CSS resets, calendar, tree, autocomplete, windowing, menus, etc.

TIBCO GI	Very extensive framework with full IDE. Dozens of widgets, vector based charting package, support for SOAP. IDE has WYSIWYG GUI layout, step through debugging as well as code completion
Bindows	Been around since 2001, lots of great widgets and library routines. Not always browser independent
JackBe	Focused at corporate environments. Able to emulate a desktop application. Very fast performance even on a dial-up line. Uses a proprietary "screen" format that gets interpreted by a JavaScript runtime engine.
Active Widgets	Rich set of JavaScript widgets. Rich grid widget is a high spot.

AjaxAC	Separates events from HTML, subrequest support, event management, easily remotes to PHP methods, start of a widget framework
Cajax	Low level. Simplifies server side programming, limits client side JS required, suggest widget, plug-in for form submission.
CakePHP	OO, modeled after Ruby on Rails, integrated CRUD, fast, flexible templating, nice docs
CPAINT	OO library for Ajax connectivity. Local & remote functions, single or multiple XHR, supports both POST & GET
JPSpan	PHP remoting via JavaScript
XAjax	Low level. PHP remoting via JavaScript. Simplified DOM updates on XHR response.
XOAD	Uses JSON and native PHP serialized objects, security emphasis, server side events, HTML manipulation. Tutorials and documentation.

Ajax Toolkit: Java Frameworks

Ajax JavaServer Faces Framework	Converts JSF applications to Ajax
Ajax JSP Tag Library	Set of tags for Ajax. Autocomplete, callout, select, toggle, update fields
DWR	For remoting Java methods from JavaScript code. Contains a number of widgets.
Echo2	Auto generate HTML & JavaScript from server side. Write in pure Java.
Guise ($$)	Application framework that uses the desktop UI approach to components and events (hides HTML generation and DOM manipulation.) Controls, dialogs, flyovers,
JSP Controls Tag Library	For portlet style JSP components. Supports Ajax and non-Ajax modes
SWATO	JS library based on Prototype. Uses JSON for marshalling. Direct remoting for POJOs. Widgets include auto suggest, template, logger.
AjaxAnywhere	Turns existing JSP/JSF/Struts/Spring components into Ajax Aware components
WebOrb for Java	Ties together Ajax, Flash and Java objects with XML services

Prototype

```
function makeRequest(url) {
    var xhr = new Ajax.Request(url, {method:'get', onComplete: showResponse})
}
function showResponse(xhr) {
    $('helloArea').innerHTML = xhr.responseText;
}
```

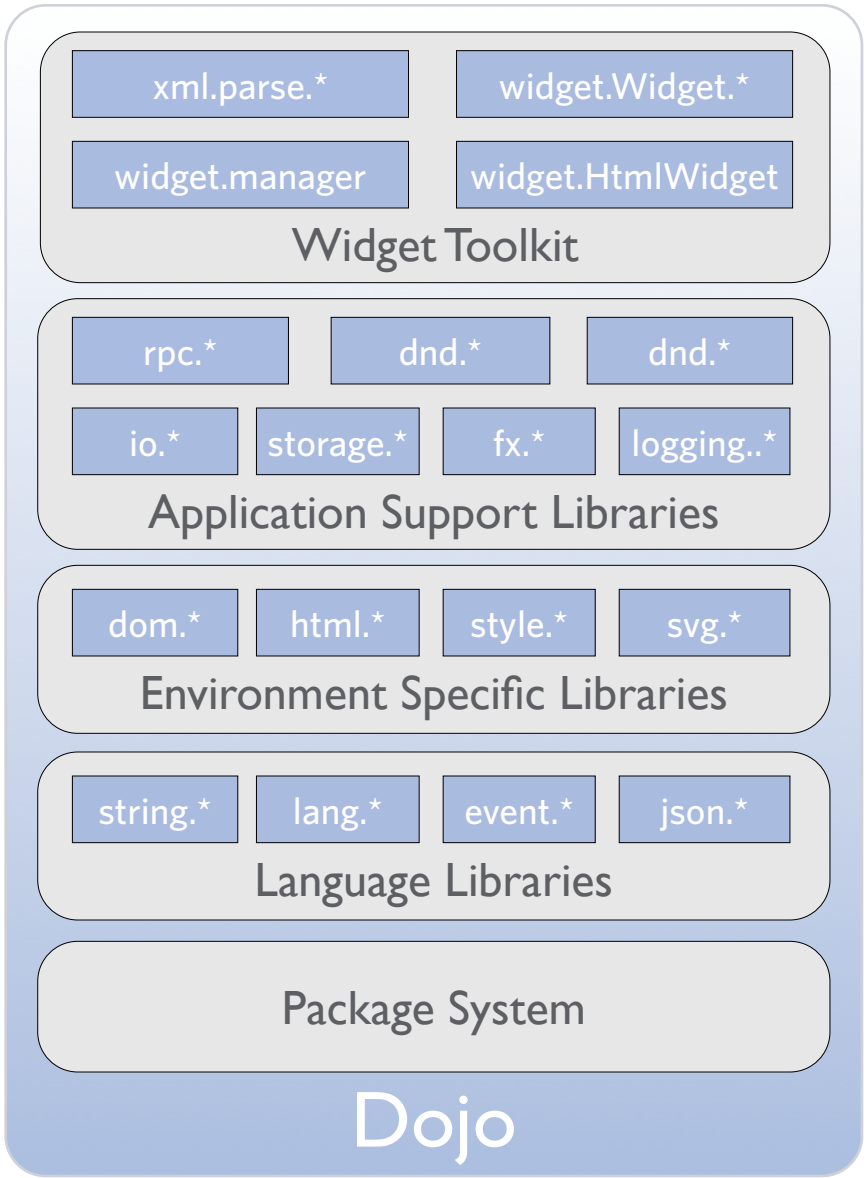
- $()
- $F()
- Try.these()
- Ajax.request()
- Ajax.Updater()
- Element.show/hide/toggle/remove
- Object-oriented extensions

- Makes Ajax request/response simple

```
function makeRequest(url) {
    var xhr = new Ajax.Request(url,
                  { method:'get', onComplete: showResponse });
}

function showResponse(xhr) {
   $('helloArea').innerHTML = xhr.responseText;
}
```

- Rico allows multiple targets in the response

Dojo



- Events: multiple listeners, browser abstraction, memory leak protection
- Aspect-oriented
- I/O: common interface for multiple transports, single/multi response
- Built-in packaging system
- Widget building w/HTML & CSS fragments, automatic reuse

Microsoft Atlas Features

- Extensible core adds lifetime mgmt, inheritance, multicast event handler, and interfaces
- Base class library: strings, timers, tasks
- UI framework for attaching behaviors to HTML
- Simple connectivity
- Set of rich UI controls (auto complete, popup panels, animation, and drag and drop
- Browser compatibility layer

Microsoft Atlas: Interesting Concepts

- Update panels: mark region for auto update
- Behaviors: encapsulate actions to associate with DHTML events [floating, hover, popup, autocomplete]
- Validators [required, type, range, regex, etc.]
- Data binding: connect controls & manage flow between them [data entered, reflected elsewhere]
- Bindings transformers: support transform event as part of a binding [ToString, invert, etc.]

Microsoft Atlas

- Two models: XML scripting & JS API
- Will be fully supported by Microsoft tools

Advanced Topics

Problems and Challenges with Building Ajax Applications

adaptive path

Bill W. Scott, Y! Ajax Evangelist

b.scott@yahoo.com

How Ajax Changes Things

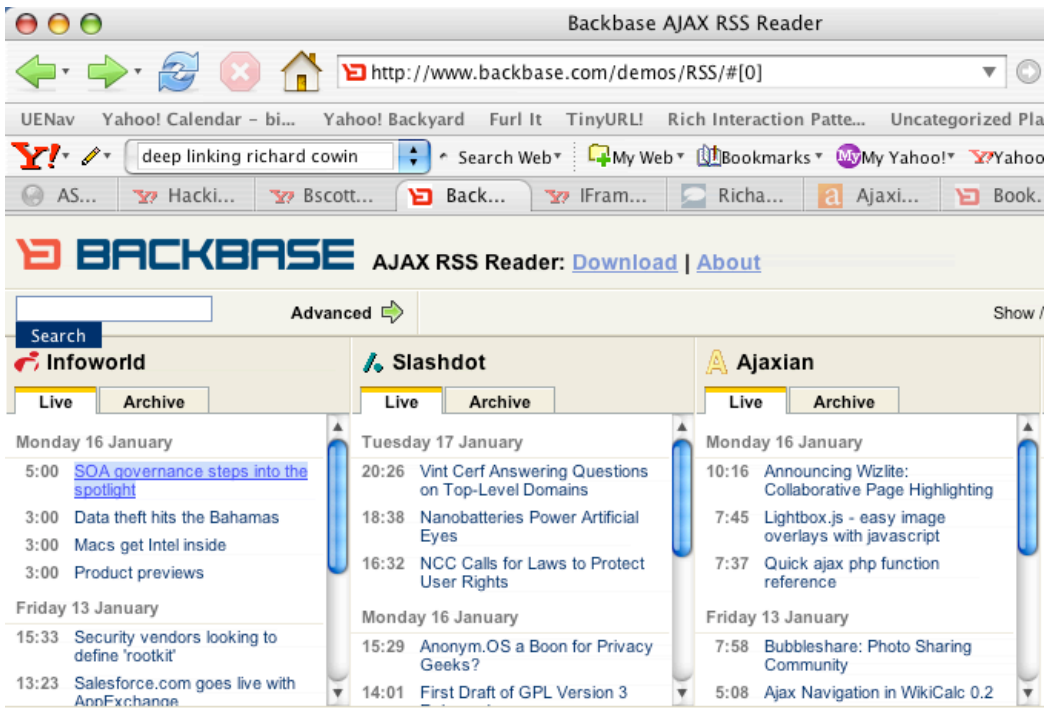
Classic Web	Ajax/Web 2.0	Problems/Challenges
Page to page navigation	Micro content	Back button, SEO, bookmarking, accessibility, security
URL/Link = User's location	Application state	Back button, SEO, bookmarking, accessibility
Browser history	Application history	Back button, bookmarking, accessibility
Back button = Undo	Is unpredictable	Back button, bookmarking, accessibility
Little JavaScript	More JavaScript	Accessibility, degradability, security, memory, performance, debug, obsfucation, error handling
Document model	Application model	Back button, SEO, bookmarking, accessibility
Static content	Dynamic content	SEO, bookmarking, accessibility
Course-grained events	Micro states	Back button
Synchronous	Asynchronous	Error handling
Browser chrome	Application controls	Back button, bookmarking, accessibility
Page Services	Web Services	Security, XML vs. JSON

Back Button

- Problem: Ajax application state changes, URL does not. No browser history of state changes/navigation
- What does the user expect?
  - Often confused with Undo
  - True context view changes should be part of history
    - Navigation tabs; but not content tabs?
    - Steps in a process
    - Tree selection when views change
  - Impact of tabbed browsers?

Back Button Solution: Fragment Identifier

- URL hash, fragment identifier (**http://a.com#loc** does not trigger a page reload
- Fragment identifier (string after '#') can be used to record state changes in the URL



- Yahoo! Maps Beta also uses this technique
- Bottom line: tricky to implement
  - Dojo, Backbase provide direct support
  - One approach: http://www.contentwithstyle.co.uk/Articles/38/fixing-the-back-button-and-enabling-bookmarking-for-ajax-apps
    - All links have fragment
    - Clicking link changes URL, not page
    - Timer monitors window.location.href & updates

- Technique: use iframes to control browser history for recording state changes
- Tricky issues abound
  - Fragile across browsers
  - onload issues
  - audible transition (on IE if sound enabled)
- Bottom line: problematic

Search Engine Optimization (Deep Linking)

- All the content may not be statically available for search engine crawlers
  - Won't find content to index your pages correctly
- Possible solutions
  - **Lightweight Indexing**: leverage existing tags such as meta, title and h1
  - **Extra Links**: extra links are placed on the site.
  - **Secondary Site**: a secondary site is fully accessible to the search engine. (See degraded experience)

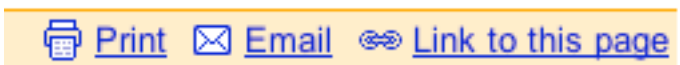
source: backbase.com

Bookmarking

- Since we have broken the history and URL paradigm, bookmarking become problematic
- What does the user expect?
  - Do they expect to bookmark application state? content viewed?
  - Desktop apps often support bookmarking. It is always content based.

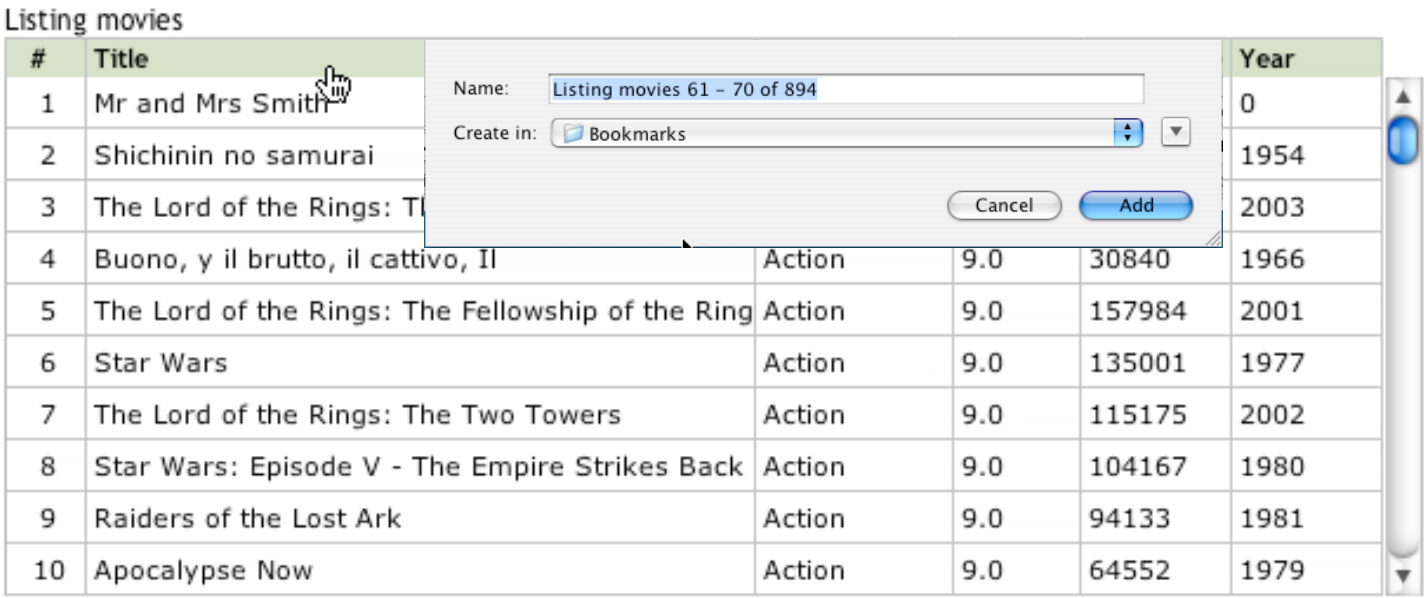
- Allow the user to save a bookmark at an interesting moment in an Ajax application
- Perhaps dynamically generate a link for bookmarking
- The URL generated for the bookmark is sufficient to restore the state

- Google Maps
  - Link is generated on each new map address
  - Link contains URL parameters to return to the page



- # OpenRico LiveGrid

http://richardcowin.typepad.com/blog/2005/07/there_has_been_.html



#	Title				Year
1	Mr and Mrs Smith				0
2	Shichinin no samurai				1954
3	The Lord of the Rings: T				2003
4	Buono, y il brutto, il cattivo, Il	Action	9.0	30840	1966
5	The Lord of the Rings: The Fellowship of the Ring	Action	9.0	157984	2001
6	Star Wars	Action	9.0	135001	1977
7	The Lord of the Rings: The Two Towers	Action	9.0	115175	2002
8	Star Wars: Episode V - The Empire Strikes Back	Action	9.0	104167	1980
9	Raiders of the Lost Ark	Action	9.0	94133	1981
10	Apocalypse Now	Action	9.0	64552	1979

```
http://openrico.org/rico/livegrid.page?
data_grid_index=60&data_grid_sort_col=rating&data_grid_sort_dir=ASC
```

Accessibility

DHTML Provides	Accessibility Expects	Problem
JavaScript enabled markup, new user interface controls	Simple markup	Markup has more meaning than expected. How does the assistive technology understand this is a tree control? How does it understand the state of a control?
Dynamic pages & content	Fairly static pages	How do I announce dynamic content?
Weak support for keyboard navigation	Keyboard navigation	Sight-impaired users will not be using rich interactions; they will use the keyboard (or another device that simulates the keyboard) But how to tab key, arrow key, select items with the keyboard?

- IBM/Mozilla Accessible DHTML API/Spec
  - Direct support for keyboard traversal
    - Supports tab key to a container, arrow keys to navigate inside the container and enter key to select item
    - Setting tabindex=-1 allows focus to be set on a object without it being in the tab order
  - A way to add metadata to HTML markup for assistive technologies to understand:
    - Widget roles, properties, values and events
  - Working with assistive technology vendors to make them aware of this API (funded development)
- Microsoft's plans

tabindex value	Purpose	Tab key navigable?
Not set	Accept default behavior	Default behavior Form elements
tabindex = "-1"	For child elements (nodes in a tree)	No. You must set the focus with JavaScript
tabindex = "0"	To tab in HTML code order	Tabbing moves you field to field in the order they are added to the page
tabindex > "0"	To specify an exact order for each field	Value describes the tab order

- Roles

```
<span tabindex="0" xhtml2:role="wairole:checkbox"
      property:checked="true"
      onkeydown="return checkBoxEvent(event);"
      onclick="return checkBoxEvent(event);">
  Any checkbox label
</span>
```

- Provides clues to assistive technologies

Progressive Enhancement

- Degradability - Managing the user experience as you move down in device/browser capability
- At Yahoo! we grade the browser by experience we will provide (A, B, X grade)
  - A-grade experience (majority of our users; greatest visual fidelity and richest interactions
  - B-grade experience (almost A-grade, but either they are on bleeding edge, low adoption or low economic value). Not considered in development or testing
  - X-grade experience (baseline of support; content but little style or interaction). Crawlers and old browsers get this experience.

Graded Browser Support @ Yahoo!

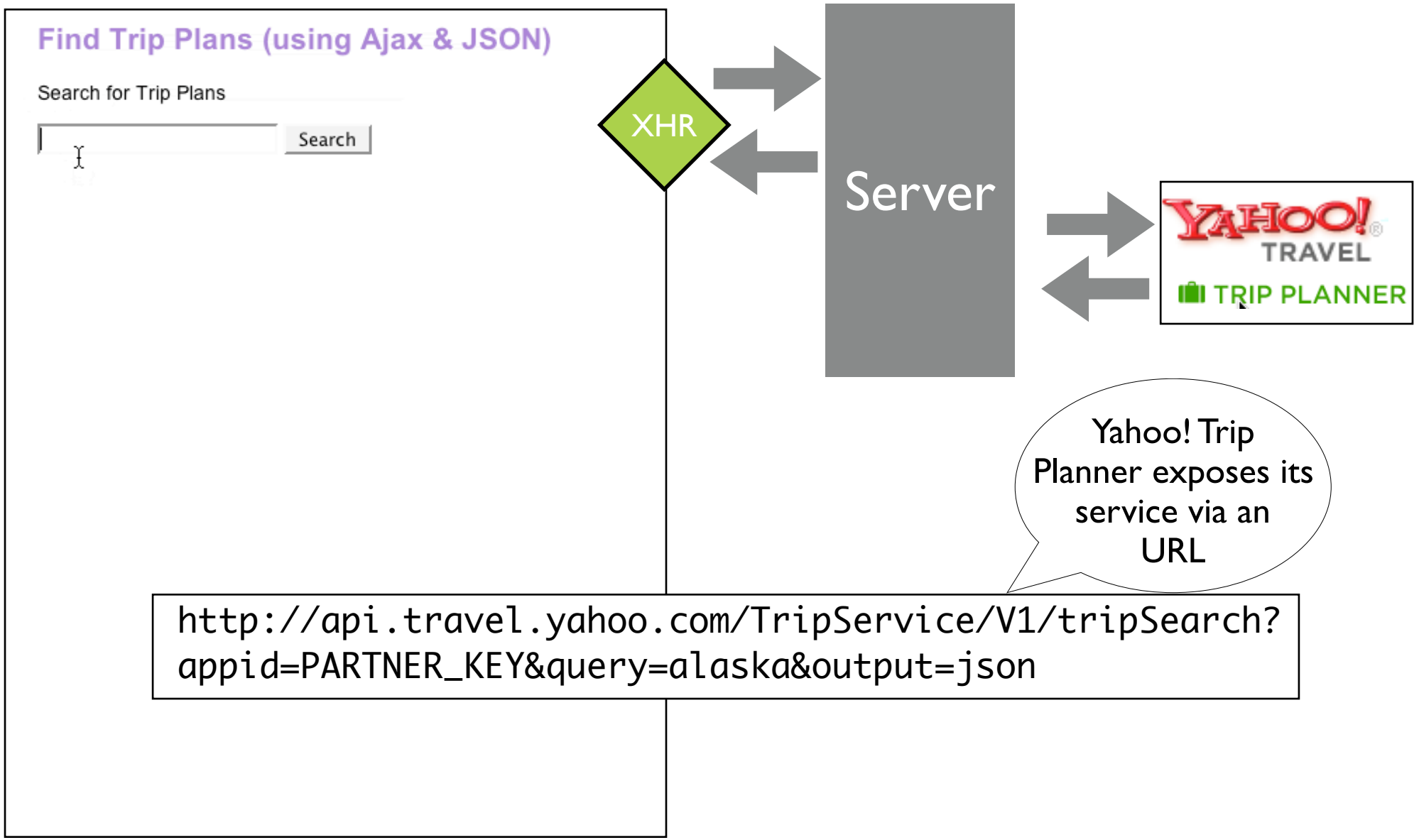
	Win 98	Win 2000	Win XP	Mac 10.0	Mac 10.2	Mac 10.3	Mac 10.3.x	Mac 10.4
IE 7.0	n/a	n/a	A-grade	n/a	n/a	n/a	n/a	n/a
IE 6.0	A-grade	A-grade	A-grade	n/a	n/a	n/a	n/a	n/a
IE 5.5	A-grade	A-grade	n/a	n/a	n/a	n/a	n/a	n/a
IE 5.0	C-grade	C-grade	n/a	C-grade	C-grade	C-grade	C-grade	C-grade
Netscape 8.0	X-grade	X-grade	A-grade	n/a	n/a	n/a	n/a	n/a
Firefox 1.5	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade
Firefox 1.0.7	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade
Mozilla 1.7.12	X-grade	X-grade	A-grade	X-grade	X-grade	X-grade	X-grade	X-grade
Opera 8.5	X-grade	X-grade	A-grade	C-grade	C-grade	C-grade	X-grade	X-grade
Safari 1.0	n/a	n/a	n/a	X-grade	n/a	n/a	n/a	n/a
Safari 1.1	n/a	n/a	n/a	X-grade	X-grade	n/a	n/a	n/a
Safari 1.2	n/a	n/a	n/a	X-grade	X-grade	X-grade	n/a	n/a
Safari 1.3	n/a	n/a	n/a	n/a	n/a	X-grade	A-grade	n/a
Safari 2.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	A-grade

- Pre-emptive nag bar
- Semantic Markup
  - What's good for accessibility is good for degradability
- Design Issues
  - What is the experience for C-grade browsers?
  - What is the experience for non-browsers?

Web Services

- Web 2.0 - Web as a Platform. Lots of web services!
- YAHOO! DEVELOPER NETWORK
  - del.icio.us, Finance, flickr, HotJobs,
    Maps, Merchant Solutions, Music, Design Patterns, RSS Feeds, Search, Search Marketing, Shopping, Travel, Traffic, upcoming.org, UI library, weather, webjay, widgets, JS Developer Center, PHP Developer Center
- Google™ Google Web APIs (beta)
  - maps, search, desktop, sitemaps,
    adwords, finance, calendar

Web Services Example



- JSON means
  - Trivial parsing
  - Faster than XML
  - Friendlier for developer
  - Runs everywhere
  - It's JavaScript, simpler programming model
  - Very stable... never will change!
  - Strong community, wide acceptance
- Yahoo! behind it

Security: Same Site Rule

- The domain of the URL request destination must be same as one that serves up page containing script
- Why? XHR requests to a different site would carry cookies. Cookies would spoof authentication.
- Solutions
  - Proxies
  - <script> hack
  - Other remote scripting tricks
  - JSONRequest (coming)

- I want to access multiple services (from different domains) without setting up a separate server (scalable, simpler to implement)
- Solution: Protocol that POSTs JSON text, gets response, parses the response into JavaScript value
  - No cookies sent
  - No other file formats accepted. Must be valid JS
  - JSON is safe JavaScript (data not functions)
  - Little or no error information provided to a miscreant
  - Accumulates random delays before trying again to frustrate denial of service attacks
  - Can support duplex!

Memory Management

- Hey, Internet Explorer is leaky!
  - Its memory garbage collector is the culprit
    - Uses reference counting; JS uses Mark & Sweep
    - Often closures are blamed
- Common problem that causes memory leaks
  - DOM <--> Application object reference each other
  - Event handlers left hanging around
- Tool for IE: Drip

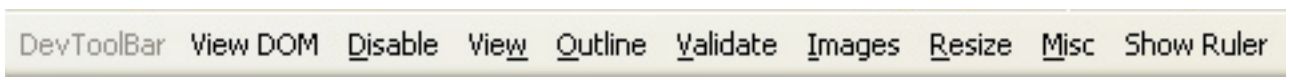
JavaScript Performance

- JavaScript requires common sense
  - Cache repeatedly accessed objects
  - Always use var for local variables
  - Use numbers for number and strings for strings
  - Avoid using eval() for accessing properties
    eval("obj."+propName) --> obj[propName]
  - Look to your loops
- And a few surprises
  - Build DOM trees downward
  - Use array.join for lengthy string concatenations

source: Adam Platti

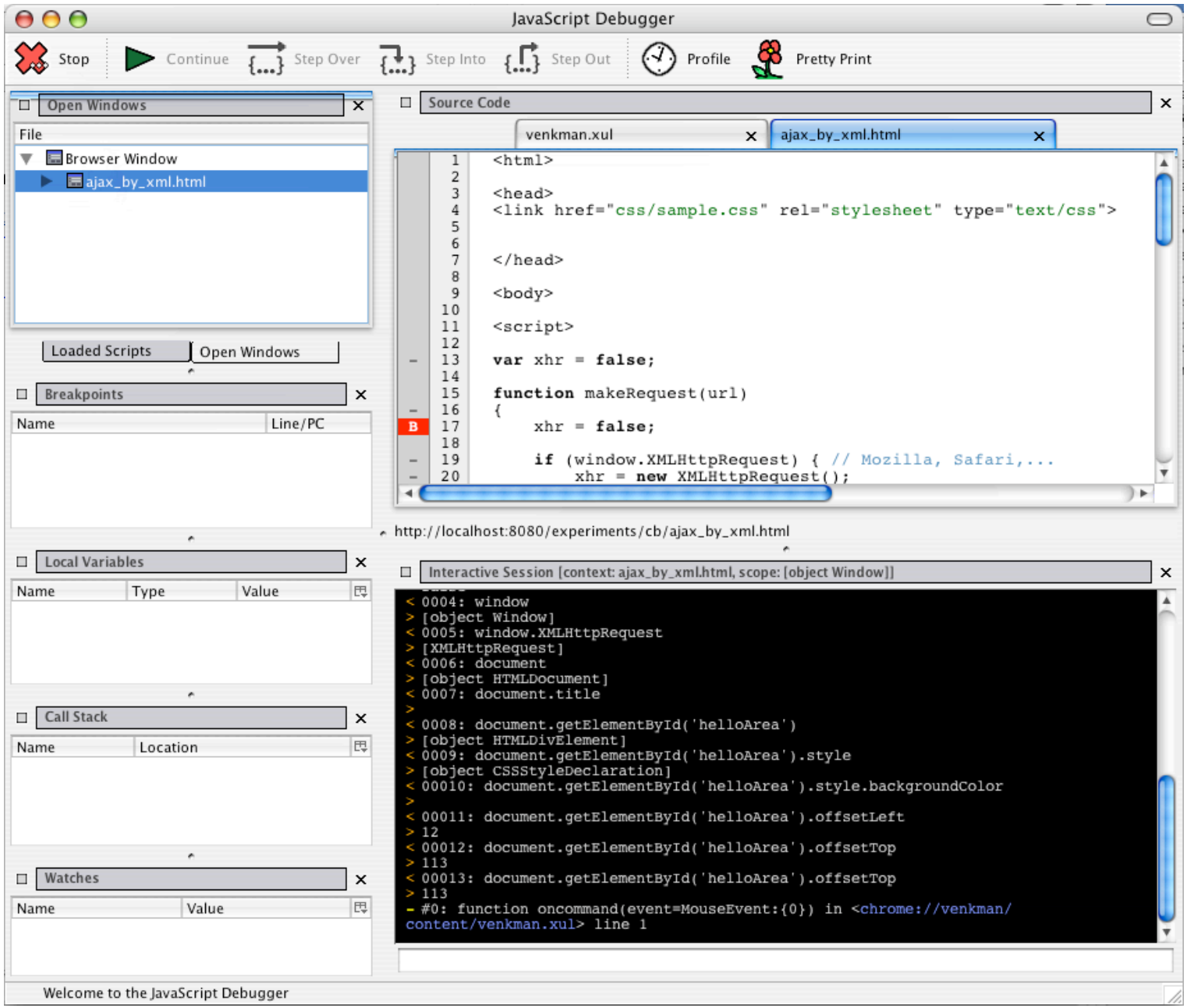
Debugging Tools

- Microsoft Script Editor
- Instant Source (IE Plugin)
- IE Dev Toolbar



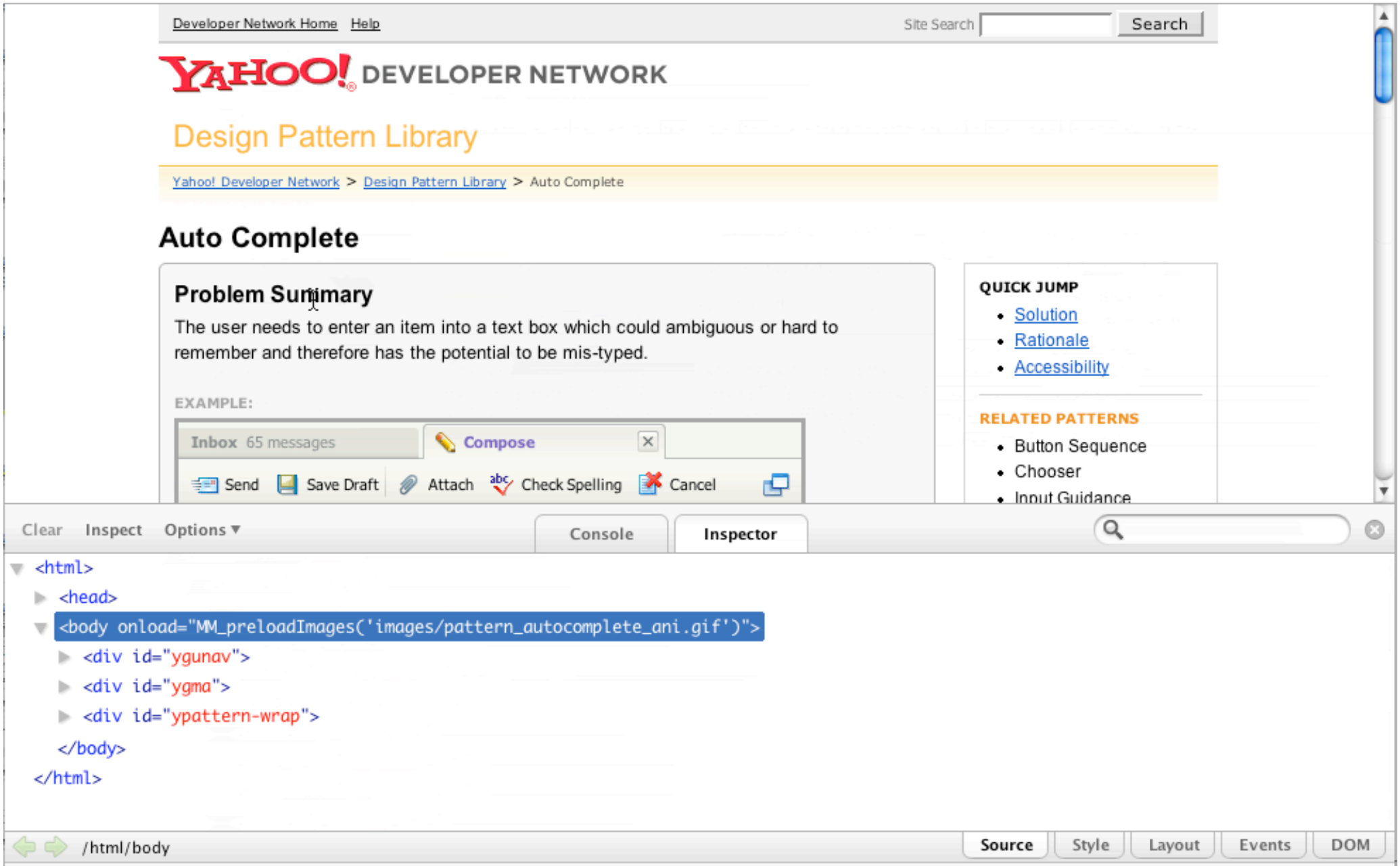
- Venkman (Mozilla)
- Firebug (Mozilla)
- DOM Inspector (Mozilla)
- Web Developer Tools (Mozilla)
- Safari JavaScript Console
- JSLint

Debugging: Venkman



- Breakpoints
- Local variables
- Watches
- Call Stack
- Profiling
- Source view
- Debug toolbar

Debugging: Firebug



- Lint are syntax checkers and validators
- JavaScript needs lint
- http://www.crockford.com/jslint/lint.html
- Scans source file & looks for common syntax problems
  - Nice way to catch silly mistakes
  - Try it at: http://jslint.com

Obsfucation or Minification

- JavaScript is easy to see with a Save or a <ctrl>-U
- JavaScript can increase page size & page load
  - Obsfucators mangle the code into unreadable form
  - Minifiers strip white space & comments
- Obsfucators go too far
  - Makes development way too hard
- Minification & compression do just enough
  - JSMin (http://www.crockford.com/javascript/jsmin.html)

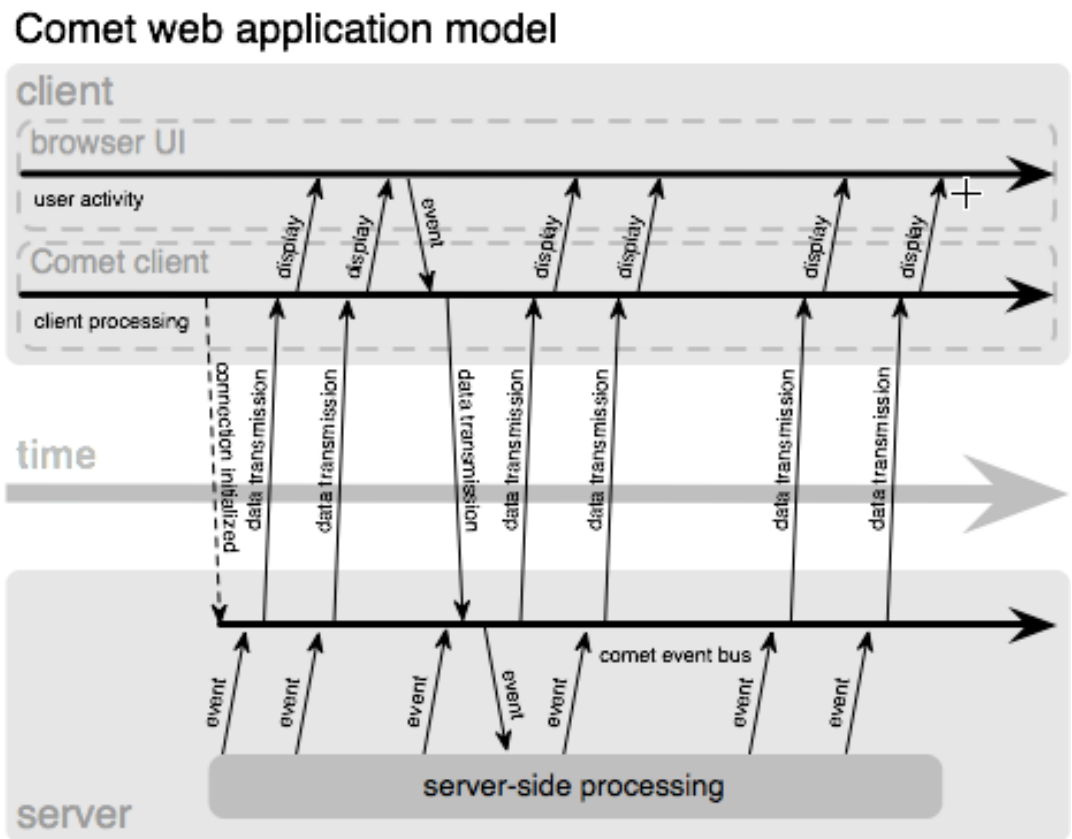
Error Handling

- Asynchronous error handling
  - Keep design simple (do you need multiple requests going at once?)
  - Will increase implementation issues
- Normal error handling
  - Check for error codes (!=200)
  - Roll your own HTTP status error codes
- Minimize server-side errors with intentional validation (error prevention)

- Difficult to correct XML, JSON or even HTML over the wire. XML & JSON are slightly harder. Server issue, but how to handle
- Use error handlers provided by Ajax frameworks
  - Prototype try.these
  - JavaScript try/catch
  - DWR has global error handler
  - Dojo error handlers

Comet

- Can deliver data to client any time
  - Delivered over single, previously opened connection
- Example applications
  - gmail's GTalk integration
  - Jot Live
  - Renkoo
  - Meebo



What about Flash?

- Adobe Flash/JS Bridge (finance.google.com)
- Adobe Flex/Ajax Bridge (8.5)
- Flash sites (maps.yahoo.com/beta)
- Flex: 2.0 is serious environment (Flash 8.5; free)
- Laszlo: Generates either Flash or DHTML/Ajax
- Communication Bridge
- Storage Mechanism
- Event Extender
- Rich Media Extender (sound, video, streaming)

Questions?