

DNA of Ajax Applications

Designing and Building Ajax Applications with Patterns



Bill W. Scott, Y! Ajax Evangelist

b.scott@yahoo.com

Ajax at Yahoo!

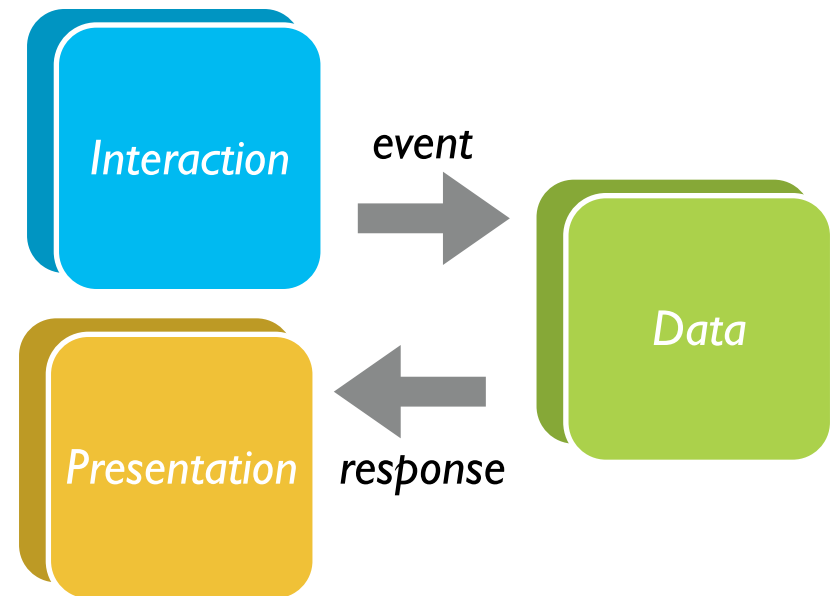
- Acquisition of Oddpost led to mission to evangelize *Ajax goodness throughout Yahoo*
- Summer 2005 - Ajax Evangelism Team Founded
 - Doug Crockford - JSON, Y! DHMTL Architect
 - Iain Lamb - Founder of Oddpost/Y!Mail Beta
 - Adam Platti - Engineer Oddpost/Y!Mail Beta
 - Me - Ajax Evangelist, Interaction Designer for Y! Widgets, Keeper of the Y! Interaction Design Patterns



Richness

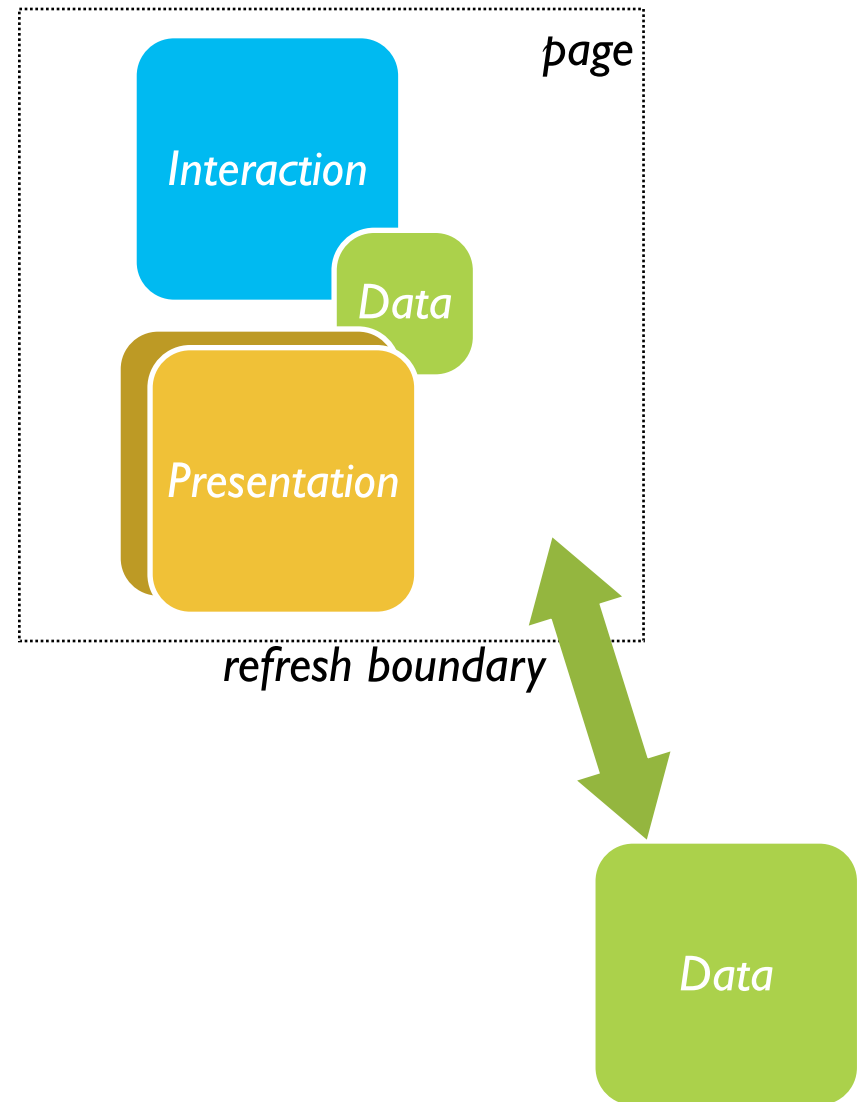
Defining Richness: Web in 3D

- Interaction Dimension
- Data Dimension
- Presentation Dimension

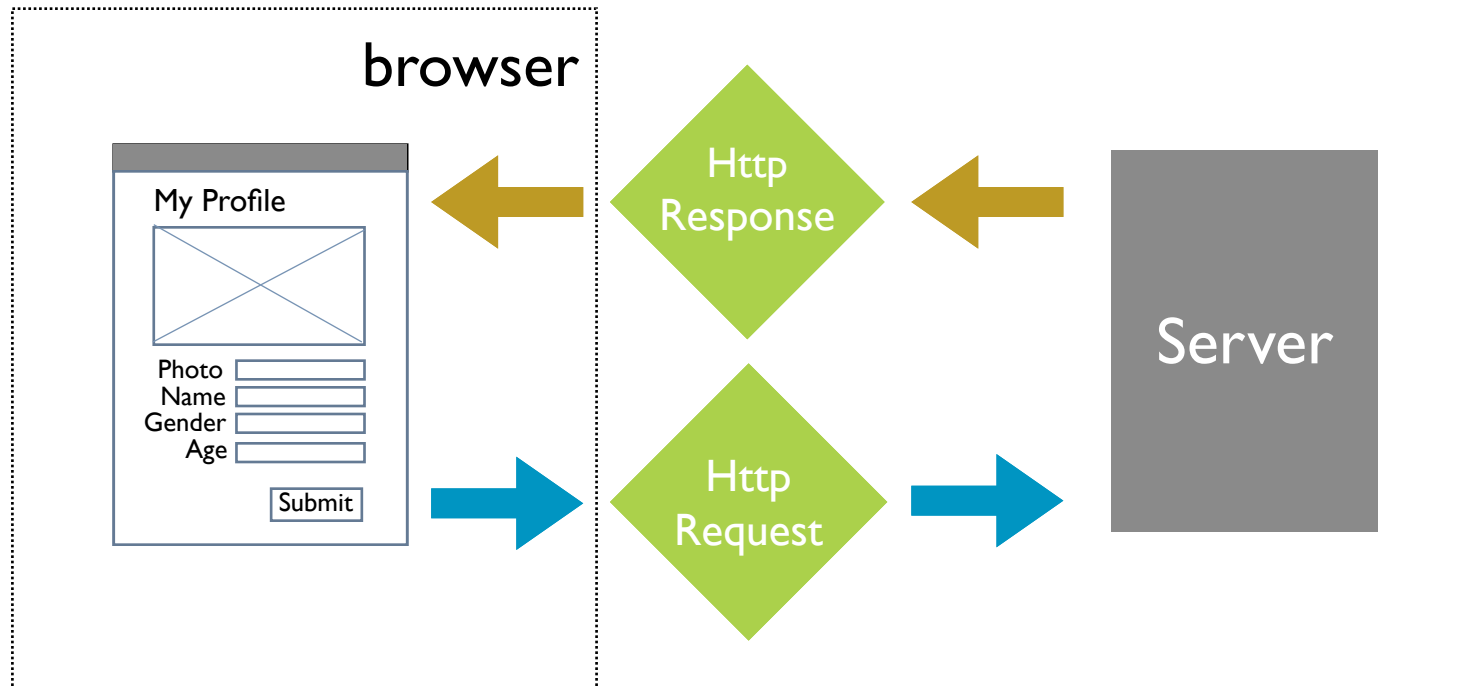


Classic Web

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



Classic Web Illustrated



Ajax = Just-in-Time. Immediacy.

Most Popular

Most Emailed | Most Viewed | Most Recommended | My Sources

- Lonely elephants, edgy apes baffled by post-Katrina life at zoo Knight Ridder - Sun Oct 16, 4:40 PM ET
 - Scientists Study Gorilla
 - Inflation Soars Higher
 - Rick Springfield's Soap
 - Feds Want Banks to Stop
- » All Most Emailed



NEW ORLEANS - An AWOL alligator has resurfaced, elephants are forlorn and apes are agitated at Audubon Zoo, one of the nation's most renowned

animal sanctuaries, left by Hurricane Katrina both broken and broke.

World

Just-in-time data



Web | Images | Groups | News | Froogle | Local ^{New!} | more »

tom cruise	
tom cruise	6,670,000 results
tom clancy	1,990,000 results
tom cruise movies	2,670,000 results
height	215,000 results
	5,780,000 results
books	145,000 results
filmography	730,000 results
ne	552,000 results
	347,000 results

Just-in-time presentation

Students to Enroll

Crew Id	Last Name	First Name
600922	Peterson	Shelley
349394	Craft	Adrian
432939	Douglas	Elayne
468370	Hill	Petra
999300	Barnes	James
344935	Smith	James
139204	Miller	Gordon

Students to Enroll

Crew Id	Last Name	First Name
600922	Peterson	Shelley
349394	Craft	Adrian
432939	Douglas	Elayne
468370	Hill	Petra
999300	Barnes	James
344935	Smith	James
139204	Miller	Gordon

Course Calendar

DEC 2005

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Course Calendar

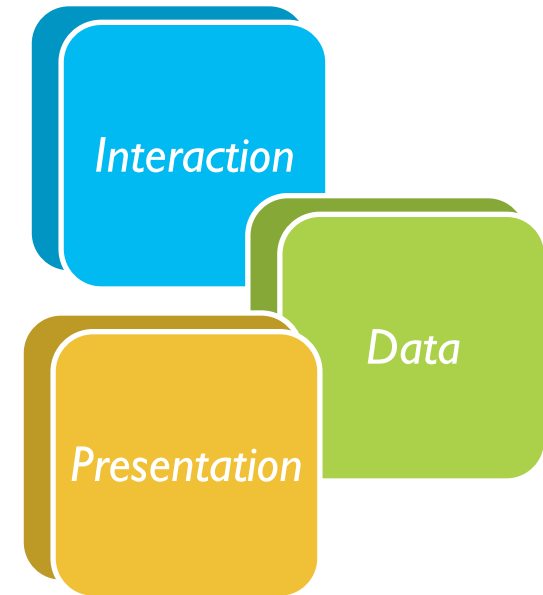
DEC 2005

S	M	T	W	T	F	S
				1	2	
4	5	6	7	8	9	
11	12	13	14	15	16	
18	19	20	21	22	23	
25	26	27	28	29	30	

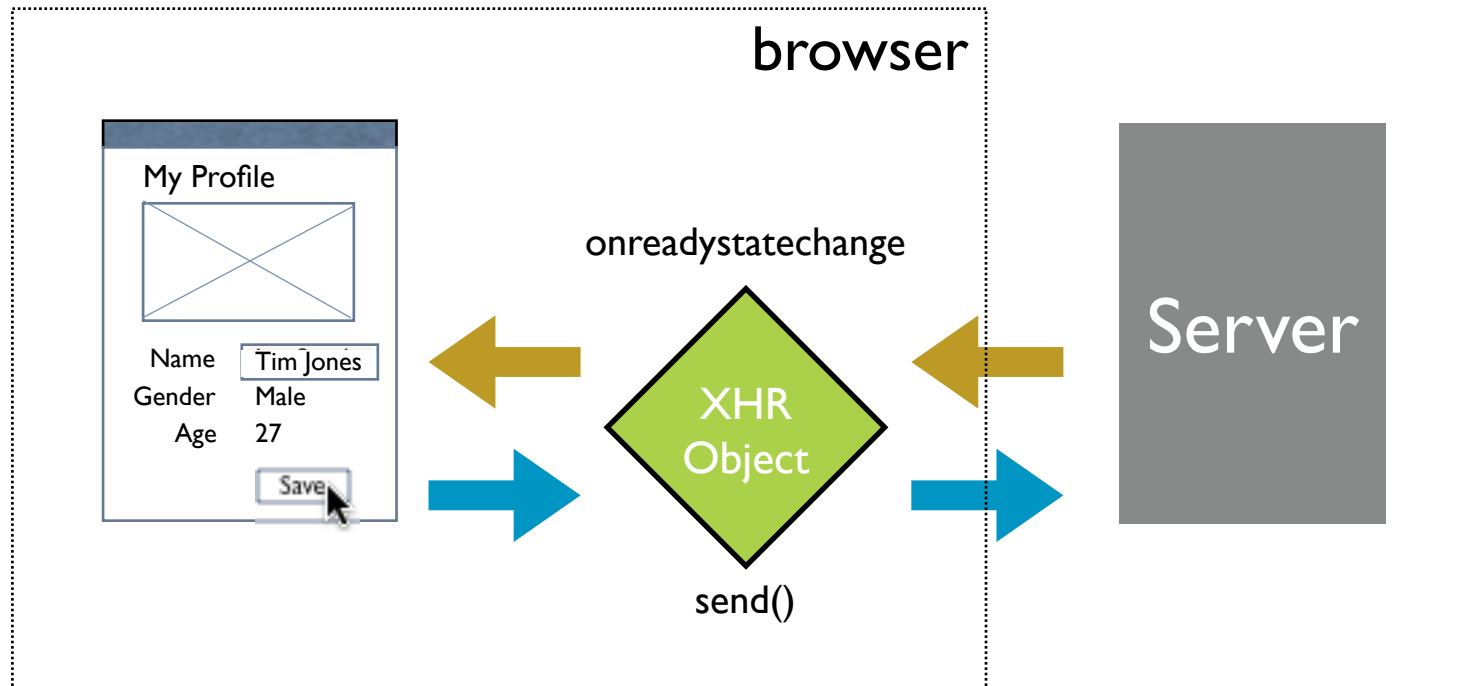
Just-in-time logic

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

Other Remote Scripting Techniques

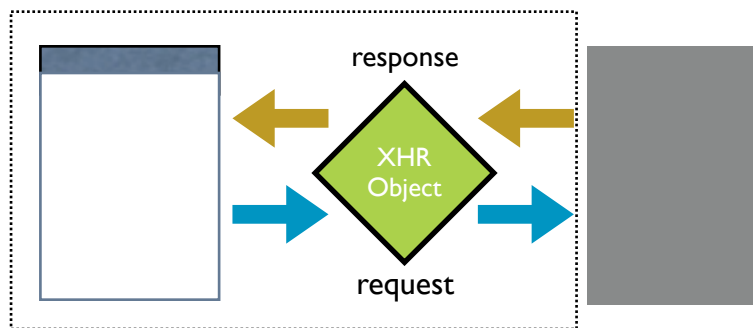
- Hidden IFrame
- ` src`
- `<script> src` hack
- CSS href hack
- JS to faceless Java applets
- JS to faceless Flash
- NO CONTENT Response
- Cookies



Patterns

So What Can XHR Do?

- Make a request
- Return a response
- And do it asynchronously



Ajax Enables New Design Patterns

- Look what such a small thing has wrought!

endless scrolling **dynamic detail** live pre-fill narrowing
actions refining search deferred loading dependent choices dynamic
updates **live search auto complete** blind content toggle on-
demand update look before you leap **auto refresh** dynamic filter

hover spy **inline edit** inline tagging rating persistent preferences
inline grid editing inline rearrange silent submit remembered
collection **persistent portals**

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

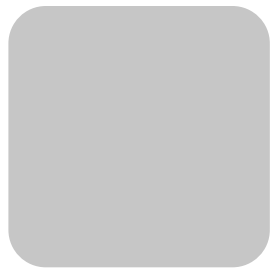
mouseout hover keypress keydown mousedown
drop filter choices mouseup drag click mousedown select
focus blur resize move timeout



Update

- User interface changes to reflect what is happening

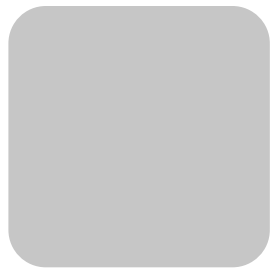
slideout slideover dropdown popup balloon
busyicon mousecursor fade dim brighten selfhealing additems
finishedindicator remove show hide resize



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

endless scrolling dynamic detail form auto-fill narrowing
actions refining search deferred loading dependent choices dynamic
updates live search suggestion blind content toggle on-demand
update look before you leap periodic refresh dynamic filter





type, Google will offer suggestions. Use the arrow keys to navigate the results. [L](#)

[Feedback](#) - [Discuss](#) - [Terms of Use](#) - [FAQ](#)

©2006 Google



Instant Search BETA



Instant Search gives you answers as you type -- no more waiting!

Why feel lucky when you can be right? With Instant Search, results instantly appear for [Yahoo! Shortcuts](#) and common searches. Give it a spin! Type in these examples below, or try your own searches:

- boston weather
- wikipedia
- nfl
- 22 5th ave new york ny
- san francisco coffee shops
- dmv



Rico LiveGrid

RICO AJAX SEARCH

LiveGrid Demonstration

Powered by Yahoo! Search

Select Search Category: **Web** Images Videos

Search the Web:

Search Results Results 1 - 5 of about 10600000 for monterey bay

1. [Monterey.com](http://www.monterey.com/)
guide to attractions, lodging, dining, and more. A service of the Monterey Peninsula Visitors and Conventions Bureau.
<http://www.monterey.com/> • text/html
2. [California State University, Monterey Bay](http://csumb.edu/)
comprehensive state university which values service through high quality education.
<http://csumb.edu/> • text/html
3. [Monterey Bay - National Geographic](http://www.nationalgeographic.com/monterey/ax/primary_fs.html)
Monterey Bay @ nationalgeographic.com ...
http://www.nationalgeographic.com/monterey/ax/primary_fs.html • text/html
4. [Monterey Bay Aquarium](http://www.mbayaq.org/)
mission is to inspire conservation of the seas.
<http://www.mbayaq.org/> • text/html
5. [Monterey Bay Aquarium: Live Web Cams](http://www.mbayaq.org/efc/cam_menu.asp)
Eight live web cameras of the Monterey Bay Aquarium's marine life exhibits and Monterey Bay, Monterey, CA. ... Monterey Bay Aquarium, 886 Cannery Row, Monterey, CA 93940 Tel: (831) 648-4800. Â© 1999-20
http://www.mbayaq.org/efc/cam_menu.asp • text/html

YAHOO! MAIL BETA Welcome, **b.scott** [Yahoo!](#) | [My Yahoo!](#) | [News](#)

Inbox 2324 messages [Send Feedback](#) | [Options](#) | [Help](#)

	From	Subject	Receive
	me	Re: rico question	Thu, 1/19/2006 01:13: Thu
	Bill Scott	[Looks Good Works Well] 1/19/2006 01:13: Thu	
	JDJ Industry Newsletter	Java Named "Programming Language of the Year"	Thu, 1/19/2006 01:13: Thu
	Travelocity Member Services	Price Alert! Flight + 5 Night Hotel to Honolulu	Thu, 1/19/2006 01:13: Thu
	Theresa Neil	Re: Job Description	Thu, 1/19/2006 01:13: Thu
	Damon Hougland	Job Description	Thu, 1/19/2006 01:13: Thu
	Finan, Angelique	RE: Speaking Engagement Opportunity	Thu, 1/19/2006 01:13: Thu
	David Schontzler	Re: [ng-dhtml] assert(), isNaN(), isNumeric(), isObject()	Thu, 1/19/2006 01:13: Thu
	TagWorld	This week in TagWorld	Thu, 1/19/2006 01:13: Thu
	me	rico question	Thu, 1/19/2006 01:13: Thu
	Anonymous	[Looks Good Works Well] 1/19/2006 10:22: Thu	
	Anonymous	[Looks Good Works Well] 1/19/2006 10:19: Thu	
	Anonymous	[Looks Good Works Well] 1/19/2006 10:19: Thu	
	Anonymous	[Looks Good Works Well] 1/19/2006 10:19: Thu	

[Standard Header](#) | [Hide Pane](#)

Got your eye on one of those messages up there?
To view your message down here in this handy Reading pane, just click on it.
To select more than one, hold the Shift or Command key.

What's your Credit Score?

Netflix-Try Free! No Late Fees

Online Degree Programs

Yahoo! Mail Beta (Oddpost)



Operation. Persist

I can save in real-time

hover spy **in page edit** inline tagging rating persistent preferences
inline grid editing inline rearrange silent submit remembered collection
persistent portals



Rocky Waters

ADD NOTE SEND TO GROUP ADD TO SET BLOG THIS ALL SIZES ORDER PRINTS ROTATE DELETE X



Add tags

Test Page

Body List Notes Files Images Writeboards Links Sharing ?

Notes

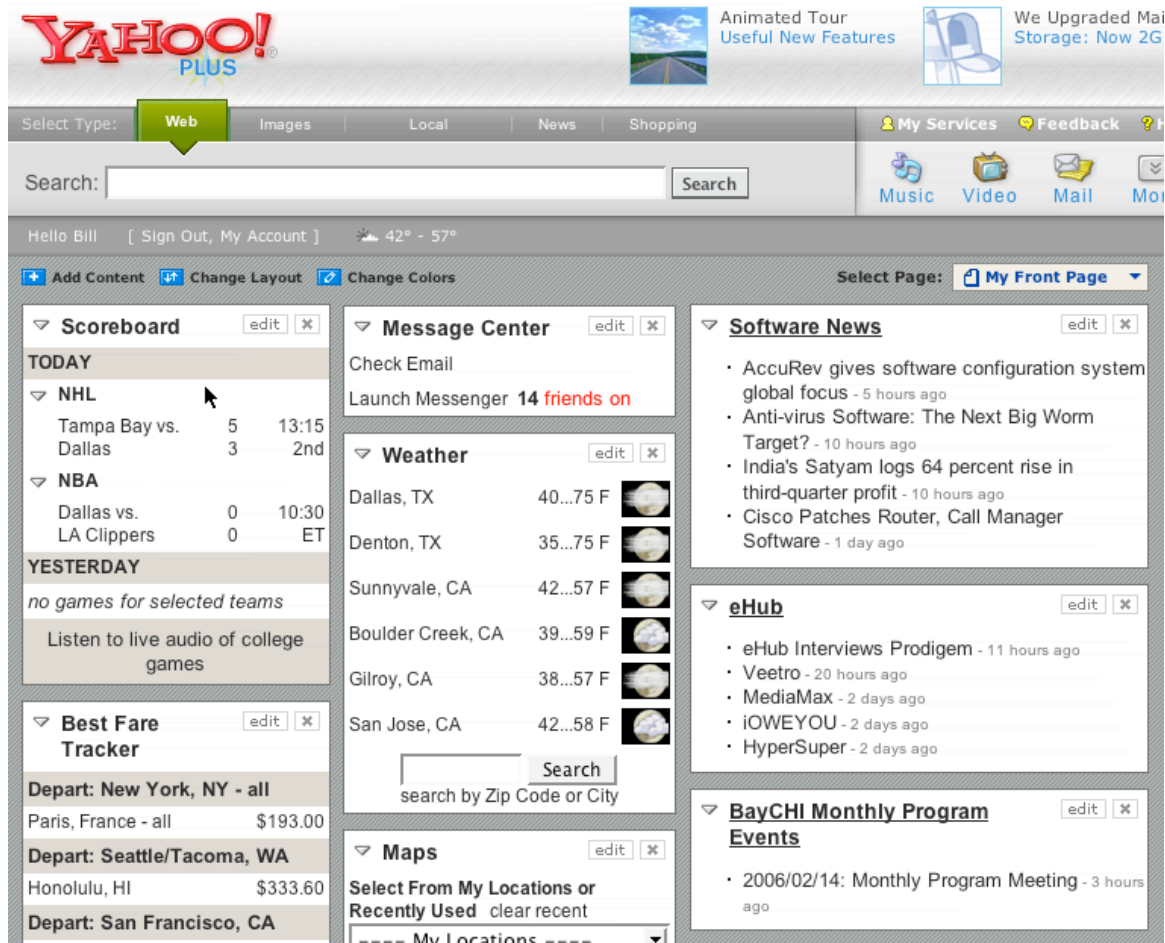
First Note (modified) Posted 30 Nov

This is just a note that I added. And some more text was added.

[Add note](#)



Persist persistent portals



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

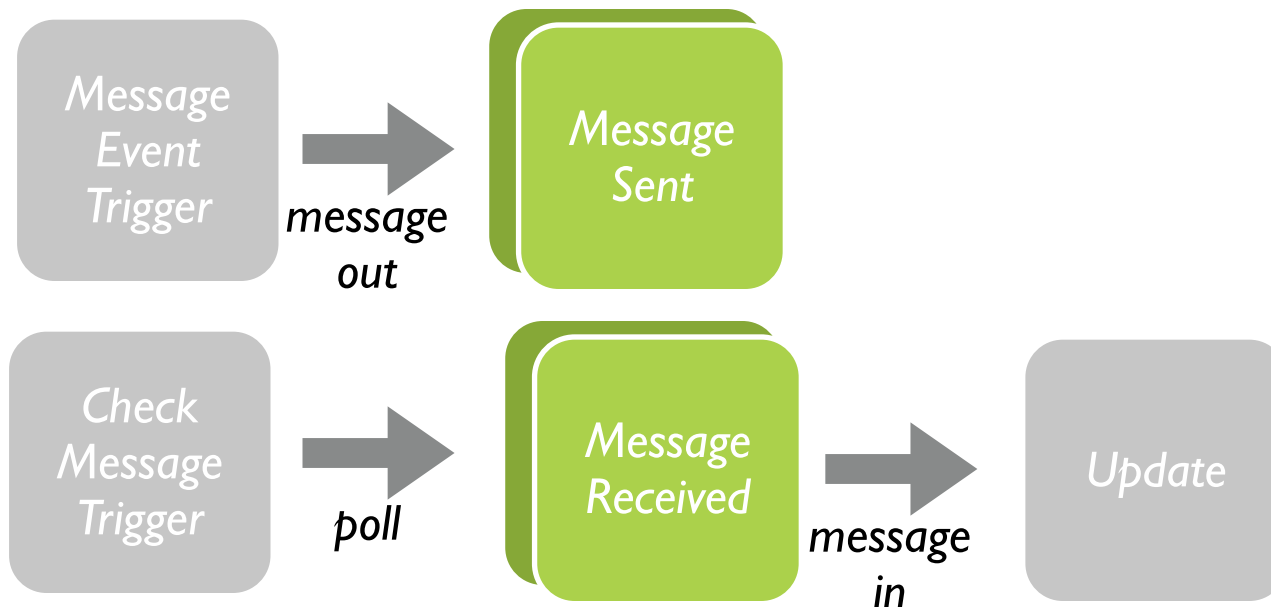
fire and forget lightweight action



Operation. Message

I can communicate instantly

two-way chat real-time auction
collaborative white board



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**
- Know how to perform basic Ajax **operations**
- Know how to work with JavaScript **triggers** (events, timers)
- Know how to **update** the DOM
- Understand the **issues** that surround Ajax
 - *We will discuss these issues this afternoon*

Pattern Ingredients

- Trigger + Operation + Update defines Ajax patterns
- So, let's make some patterns

Pattern-O-Matic





Questions?

Ajax 101

Where we get down and dirty with Ajax operations, JavaScript events and the DOM



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

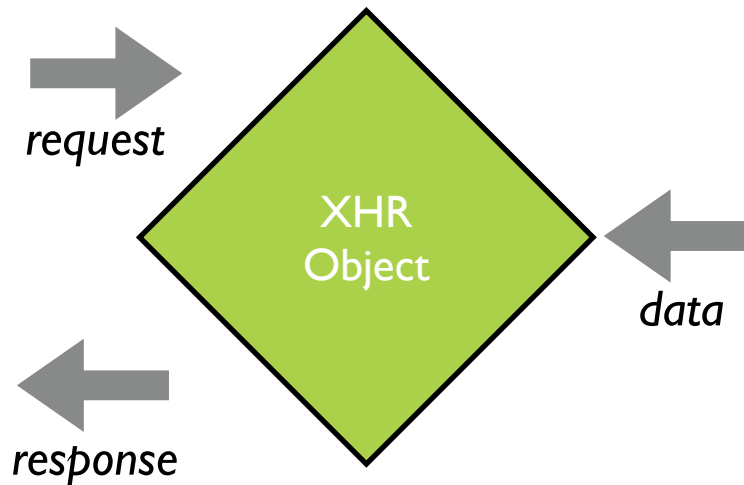
Simple Ajax 'Hello World'

Ajax Hello World

Clicking the link below will use XHR to fetch the data and then show the result in the box below.

[Make an ajax request for data](#)

This data was brought to you by Ajax!



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

/response.xml

- 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+ implements XHR as an ActiveX object
- Mozilla 1.0+, Safari 1.2+, Opera 8+, IE7 provide an XMLHttpRequest object in their API
- All XHR objects have the same methods & properties

XHR Methods

Method	Description
<code>open("method", "url", [, asyncFlag [, "username" [, "password"]]])</code>	Sets up the request object for sending a request
<code>send(content)</code>	Sends the request to the server. Can be null.
<code>abort()</code>	Stops the request.
<code>getAllResponseHeaders()</code>	Returns all response headers for the HTTP request as key/value pairs.
<code>getResponseHeader("header")</code>	Returns the string value of the specified header.
<code>setRequestHeader("header", "value")</code>	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 function is invoked when the readystate changes on the XHR object

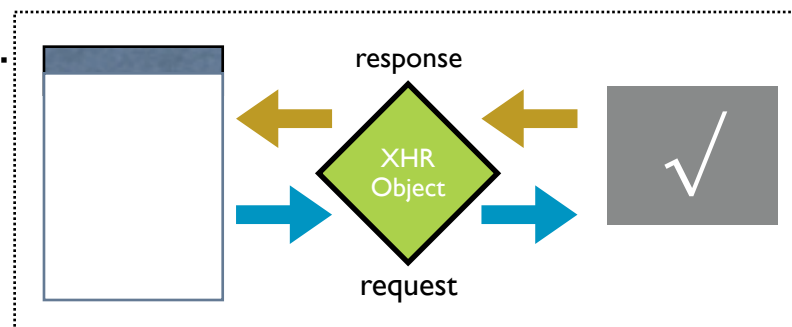
set `onreadystatechange` to callback function - `handleAjaxResponse`

open a request on the xhr object

send the request through the xhr object

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

server must be accessible via relative url



- 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

3. Make the Request: Handling the Request on the Server Side

- 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

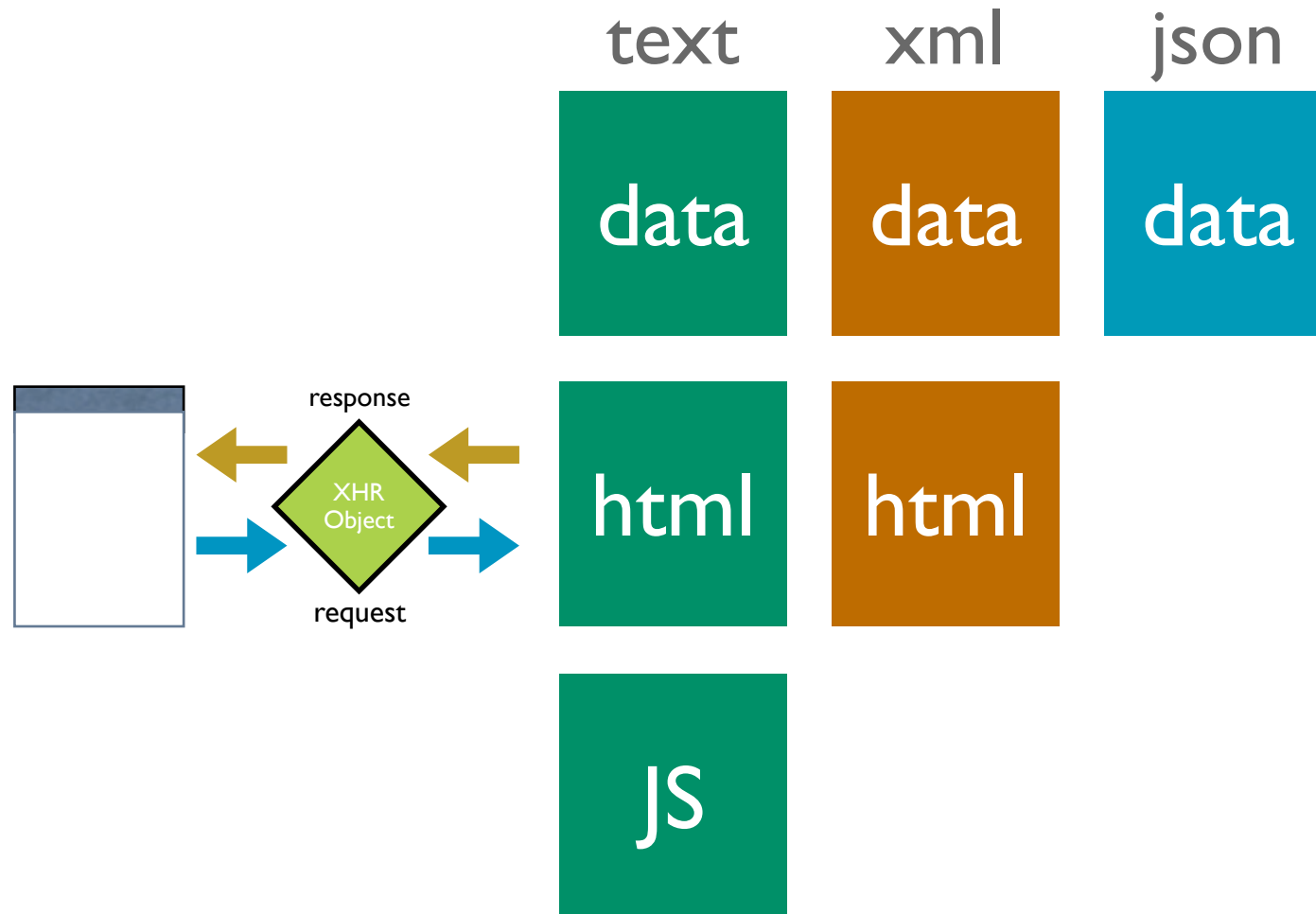
- XML version must be first line
`<?xml version="1.0" encoding="ISO-8859-1"?>`
- Set up response header's Content-type
“Content-Type”, “text/xml”
- Use well-formed XML

- 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	Is the value of a document element
getElementsByTagName	Used to place all elements into an object

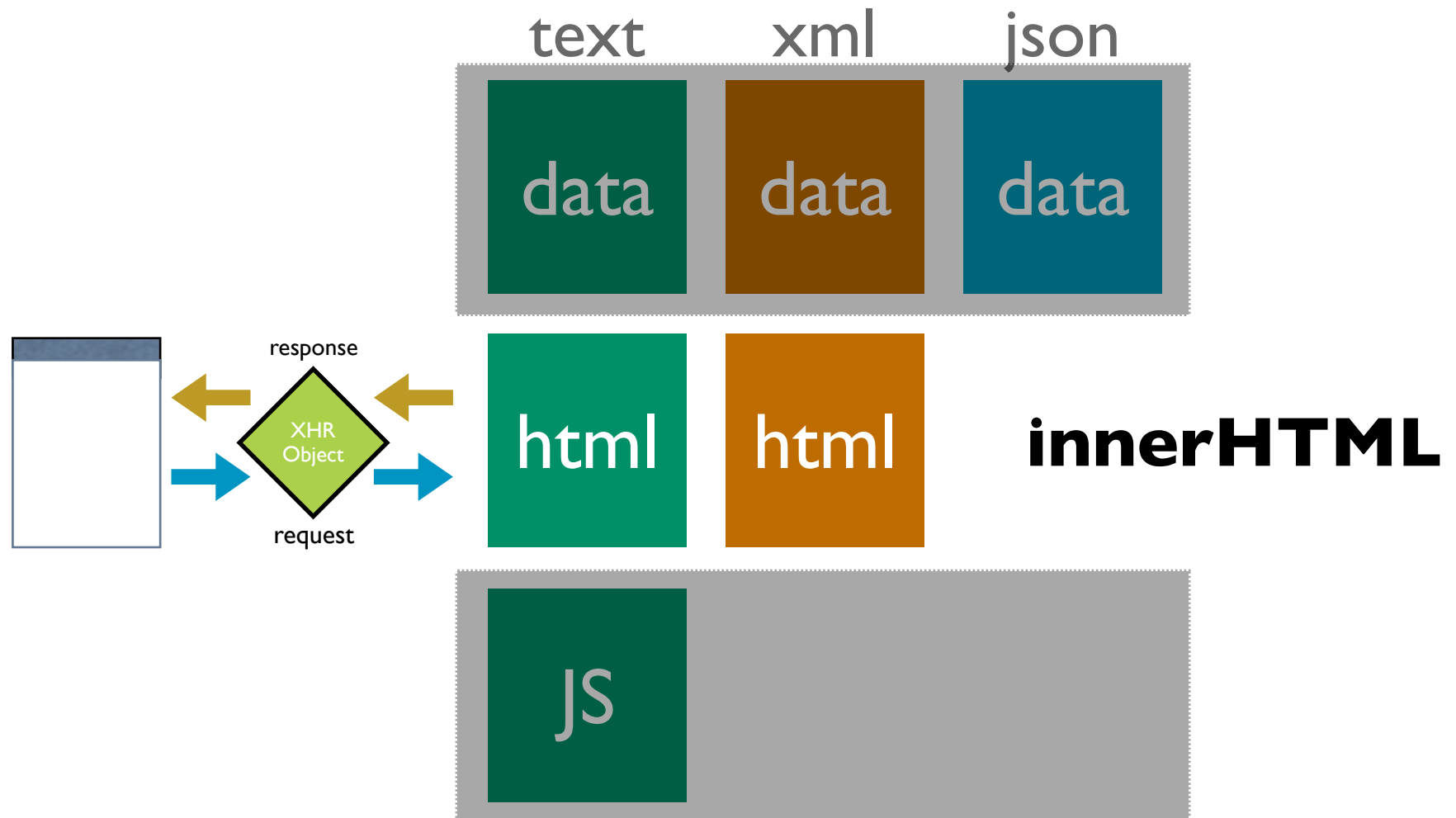
Response Options

- Different types, different formats



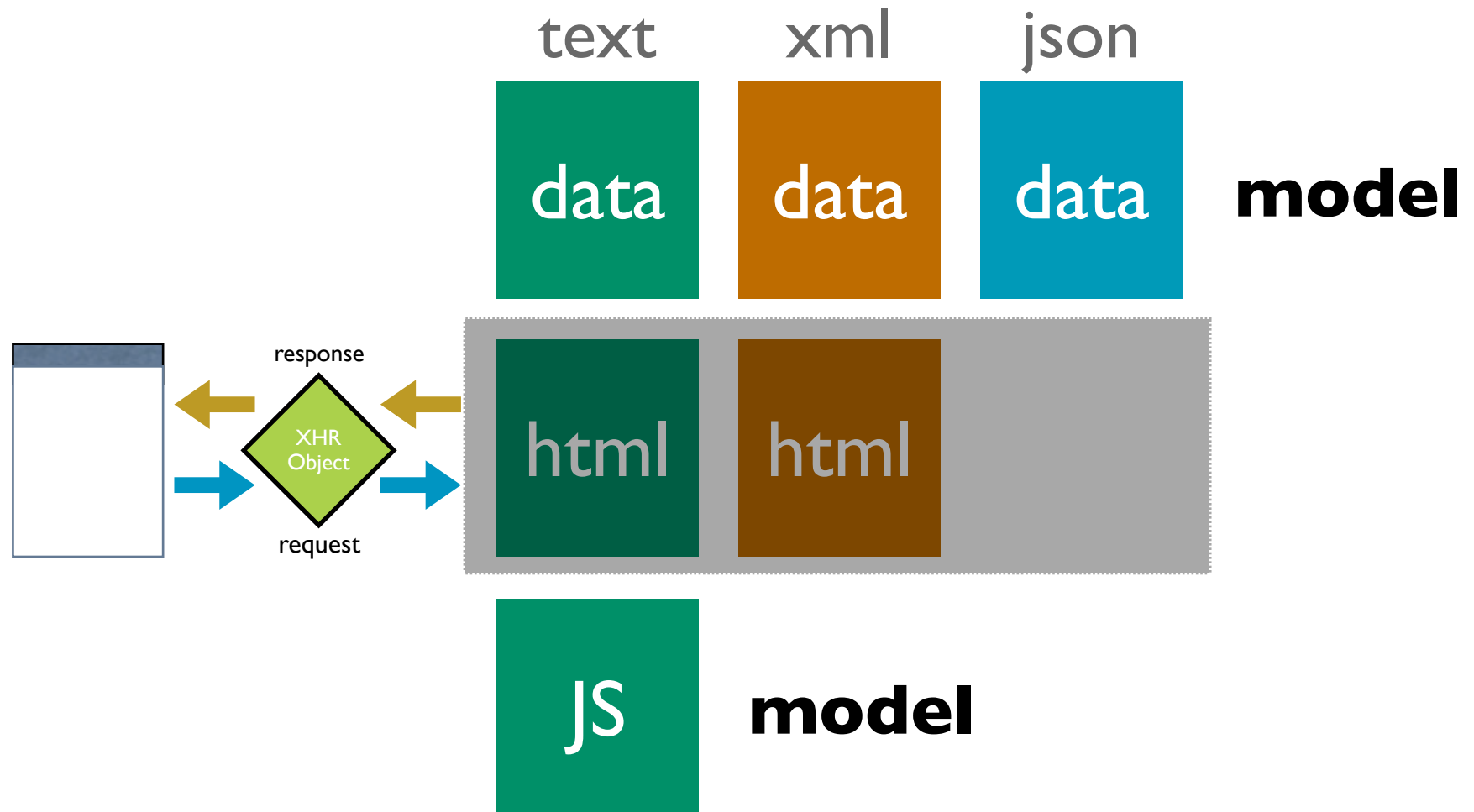
Response Options

- Snippets of HTML returned, stuffed into innerHTML



Response Options

- Data returned from the server, interface built from it



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'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);
```


- <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"
    }
  }
}
```

Ajax Operations

- Lookup
- Persist
- Validate
- Invoke

Ajax Lookup

- Use 'GET' for lookups
- Check for error codes
 - Can use other HTTP status codes to signal errors
 - Asynchronous issues
- Avoid POST

Ajax Persist

- Use GET or POST
 - By-the-book, use POST for idempotent style URLs
 - Use POST for parameter lists > 512 bytes
 - Use request URLs for either GET or POST
- Raw POST
 - Can POST raw data (`xhr.send(value)`)
 - Good for large amounts of client data (large client data sets being updated)
- Check for status & flag errors

Ajax Validate

- Use GET or POST
 - Since validate is non-idempotent GET is probably preferred (you are getting status)
 - Don't validate too early or too often
 - Once status is returned, show it in context in a non-intrusive manner
- Usually won't need a Raw Post since validation is normally on small amounts of data

Ajax Invoke

- Use POST. Invoke is normally idempotent
- URL parameters or Raw Post is ok
- Example: delete
 - Determine how you will handle delete failing. Best approach is to use In Context Busy and don't reflect the change to server state until it is valid. This will avoid confusing the user (who thinks it worked, but then it fails)



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
onDbClick	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

- Two Models (actually three!)
 - Classic style
 - `<element onclick=func() >`
 - `element.onclick=func;`
 - W3C event model
 - `addEventListener`, `removeEventListener`
 - IE event model
 - `attachEvent`, `detachEvent`

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

- Inline registration

```
<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
}
```

- Event handler with function

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

- Event handler with Object prototype

```
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;
}
```

- Arbitrary number of event handlers
- Way to remove events
- Defines capture & bubble flow
- No way to get list of handlers
- And the browsers play differently:

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

- Surprise! They do it differently

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

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



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
- 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

Document - DOM Nodes			
nodeName	id	class	
#document			
HTML			
HEAD			
BODY			
#text			
SCRIPT			
#text			
H2			
#text			
P			
#text			
SPAN		ajaxlink	
#text			
DIV	helloArea		
#text			

Ajax Hello World

Clicking the link below will use XHR to fetch the data and then show the result in the box below.

[Make an ajax request for data](#)

Using the DOM

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

Finding DOM Elements

- `document.getElementById`
 - Prototype library shortcut: `$("#idName")` or `$(id)`
- `parentNode`
- `childNodes`

DOM Manipulation

- Creating new interface elements
 - innerHTML, createElement(), createTextNode(), appendChild()
- Changing element styles
 - Visual attributes
 - Geometry

```
<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>
```

Ajax Hello World

Clicking the link below will use XHR to
fetch the data and then show the
result in the box below.

[Make an ajax request for data](#)

This data was brought to you by
Ajax!

```
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
- This is called good semantic markup


```
<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'}
```

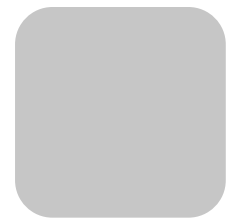
Learning Ajax

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

*Triggering
Events &
Timers*

*Using XHR
for Ajax
Operations*

*Updating
the 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 Toolkits

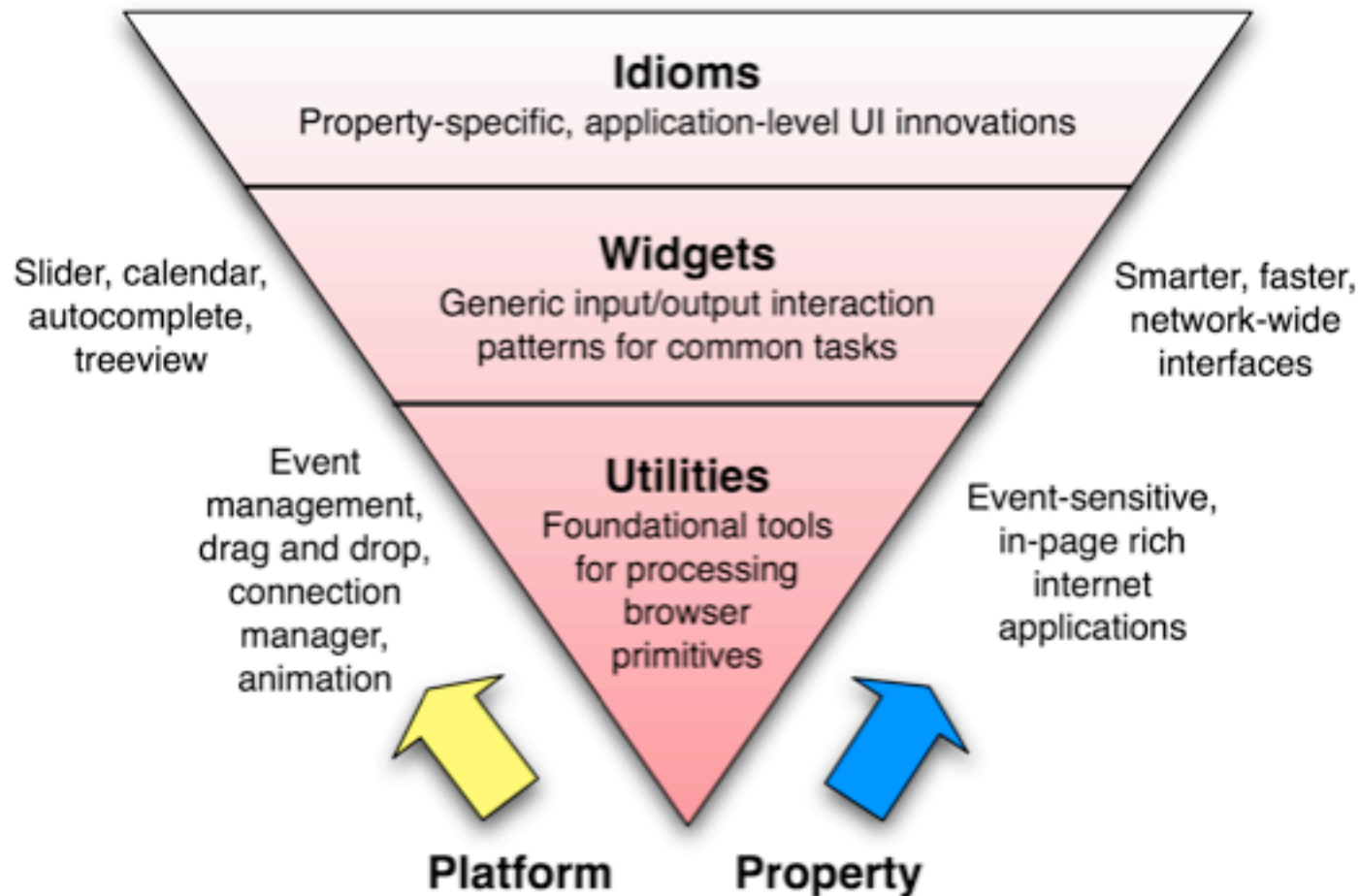
Designing and Building Ajax Applications with Ajax Toolkits



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?

Prototype	Inspired by Ruby, simplifies DOM manipulation, events, Ajax connectivity. Several frameworks based on it. Notable for the <code>\$()</code> 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
Sarissa	Encapsulates XML functionality in browser-independent calls

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, plugin 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 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 remotng 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 remotng 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
- Big Q? Tied to ASP.NET?
 - Language on the site is confusing
 - Similar to how XHR got lost!
- But I have been assured they intend the Atlas Client Scripting Framework to be totally independent of Microsoft technologies (although they want you to use them!)

Advanced Topics

Problems and Challenges with Building Ajax Applications



Bill W. Scott, Y! Ajax Evangelist

b.scott@yahoo.com

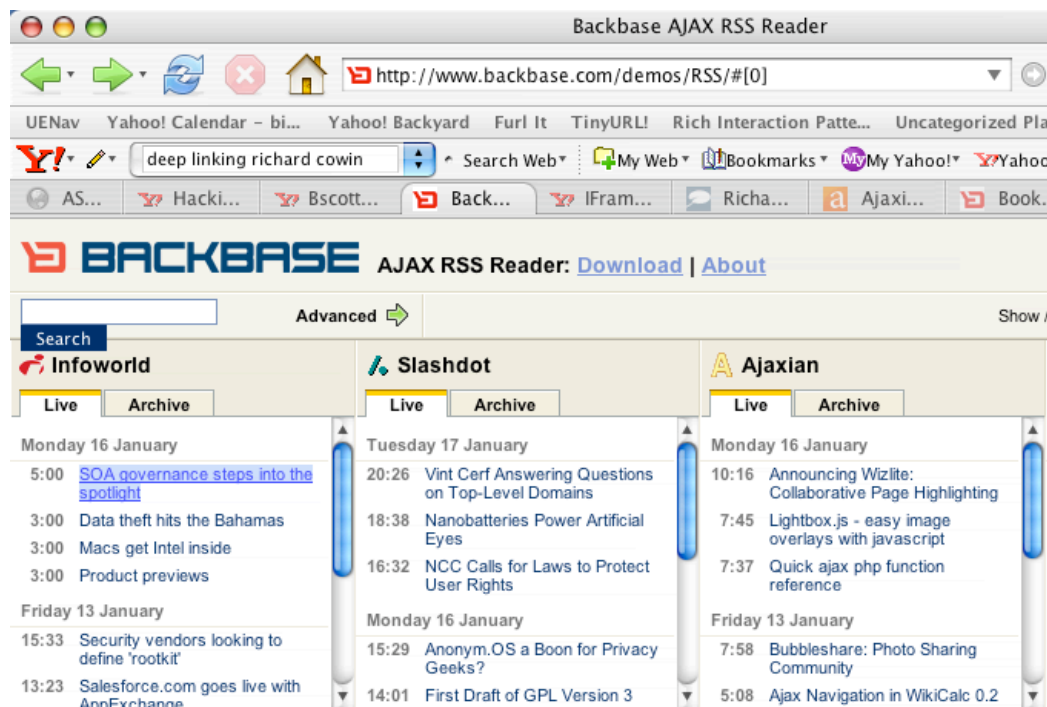
How Ajax Changes Things

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, obfuscation, 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?

- 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

 [Print](#)  [Email](#)  [Link to this page](#)

- OpenRico LiveGrid

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

Listing movies

#	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](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

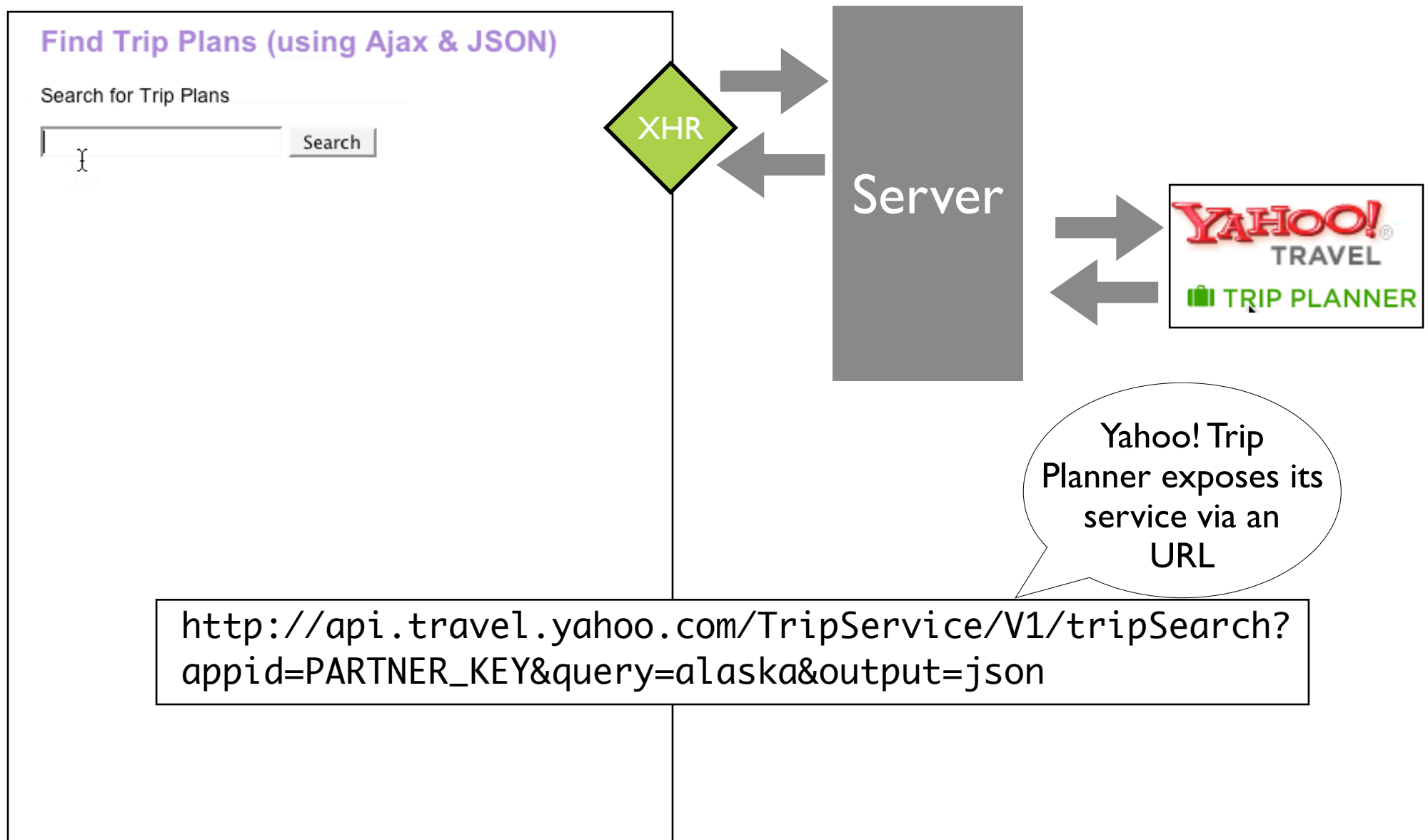
Degradability

- 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, C 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
 - C-grade experience (baseline of support; content but little style or interaction). Crawlers and old browsers get this experience.

- 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, Financ, Flickr, HotJob, Maps, Merchant Solutions, Music, RSS Feeds, Search, Search Marketing, Shopping, Travel, Traffic, upcoming.org, weather, webjay, widgets
-  **Google Web APIs (beta)**
 - maps, search, desktop, sitemaps, adwords



- 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
 - XMLHttpRequest (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

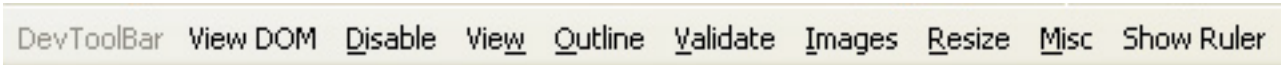
- Hey, Internet Explorer is leaky!
 - Its memory garbage collector is the culprit
 - Uses reference counting; JS uses Mark & Sweep
 - Often closures are blamed
- Common problem that causes memory leaks
 - DOM <--> Application object reference each other
 - Event handlers left hanging around
- Tool for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

JavaScript Performance

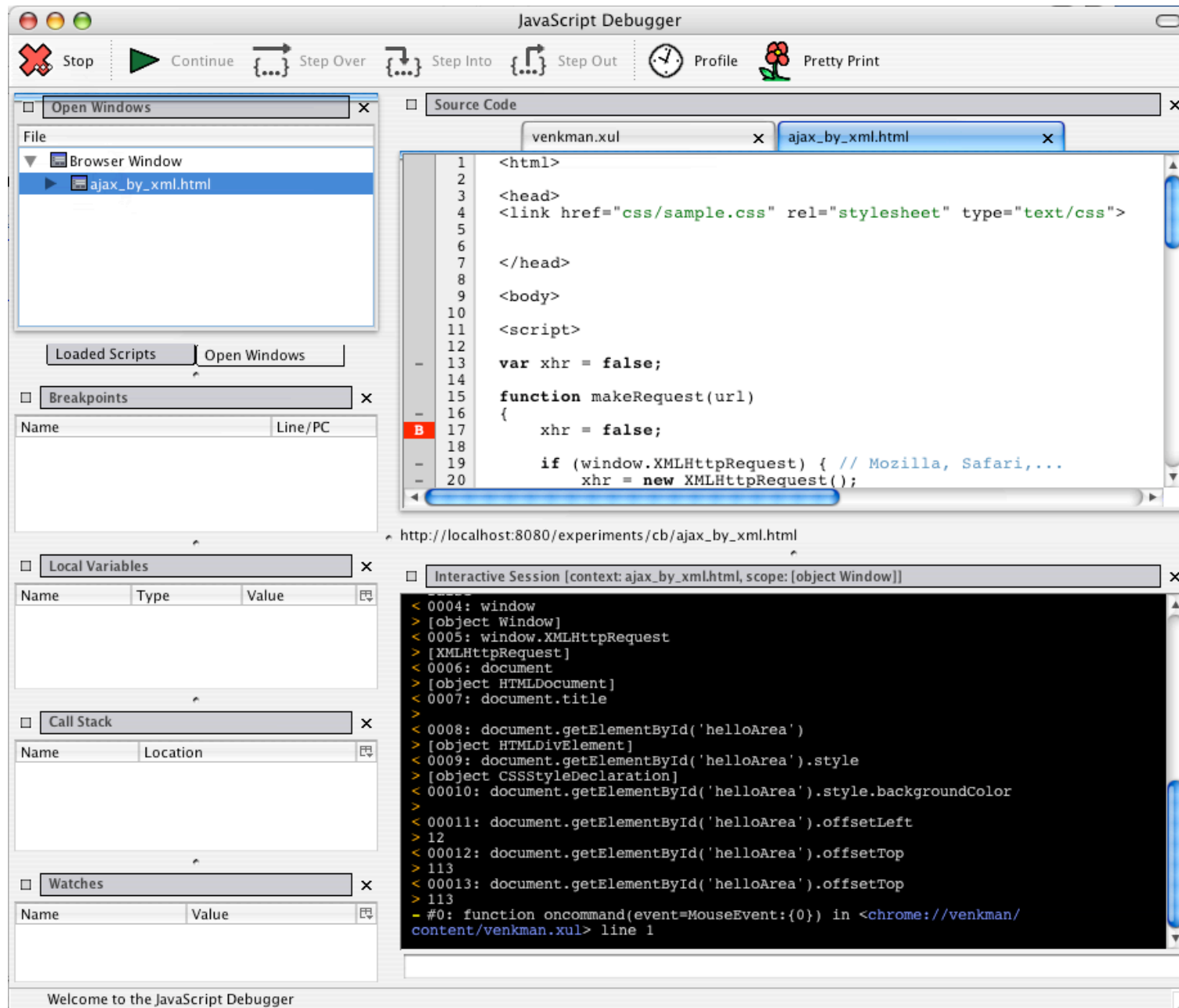
- JavaScript requires common sense
 - Cache repeatedly accessed objects
 - Always use var for local variables
 - Use numbers for number and strings for strings
 - Avoid using eval() for accessing properties
eval("obj."+propName) --> obj[propName]
 - Look to your loops
- And a few surprises
 - Build DOM 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)
- DOM Inspector (Mozilla)
- Web Developer Tools (Mozilla)
- Safari JavaScript Console
- JSLint

Debugging: Venkman



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

- 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>

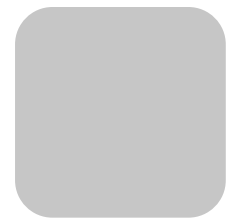
Obsfucation or Minification

- 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 ($\neq 200$)
 - 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



Questions?

Ajax & Web Services

XML and JSON Examples

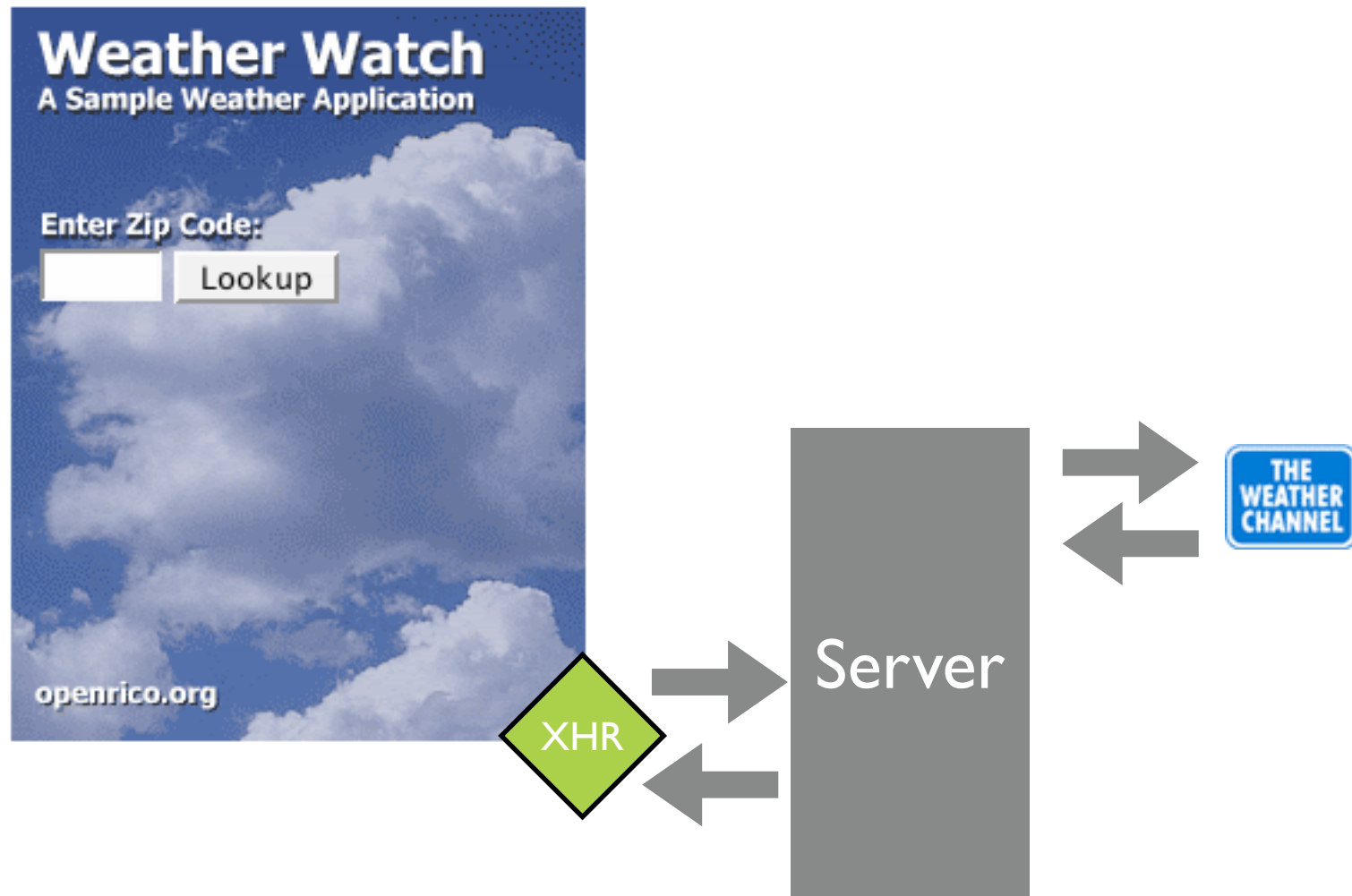


Bill W. Scott, Y! Ajax Evangelist

b.scott@yahoo.com

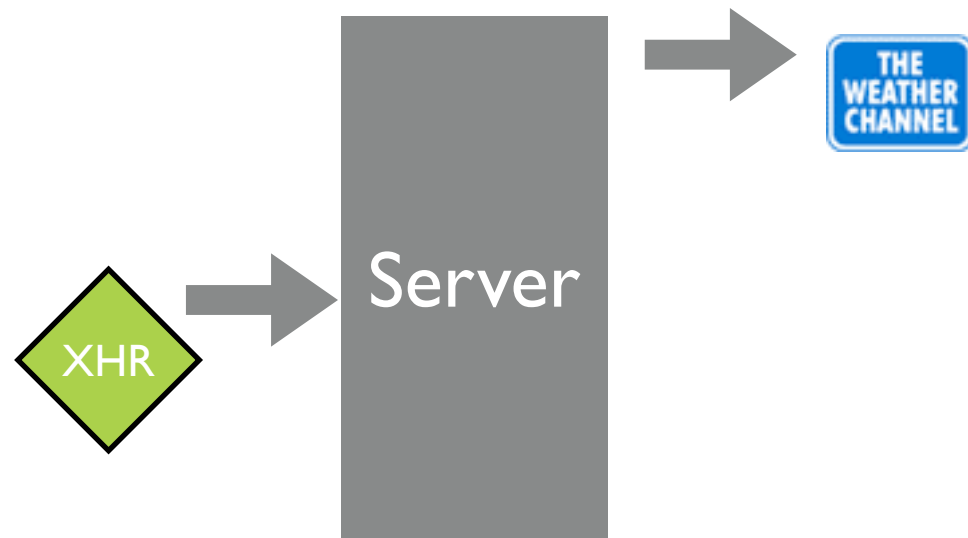
Building an Ajax Weather Widget

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`



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 cellpadding="0" cellspacing="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"



Server

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 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>
```

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

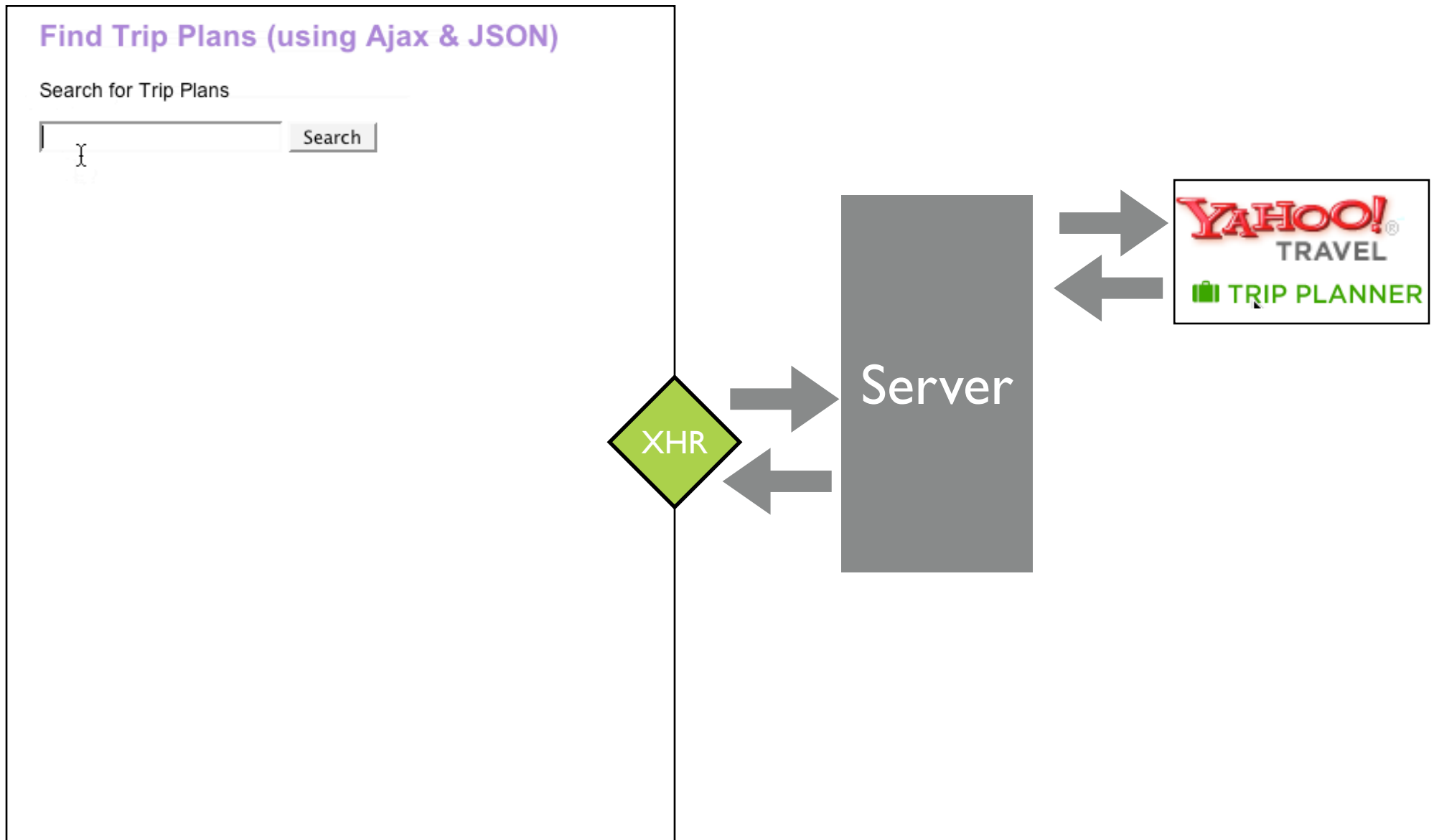
```
function registerAjaxStuff() {
    ajaxEngine.registerRequest( 'getWeatherInfo', 'ajax_weather_info' );
    ajaxEngine.registerAjaxElement( 'ccInfo' );
    ajaxEngine.registerAjaxElement( 'moonInfo' );
    ajaxEngine.registerAjaxElement( 'sunInfo' );
    ajaxEngine.registerAjaxElement( 'sponsoredLinks' );
    $('zip').onkeydown = handleEnterKey.bindAsEventListener($('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';

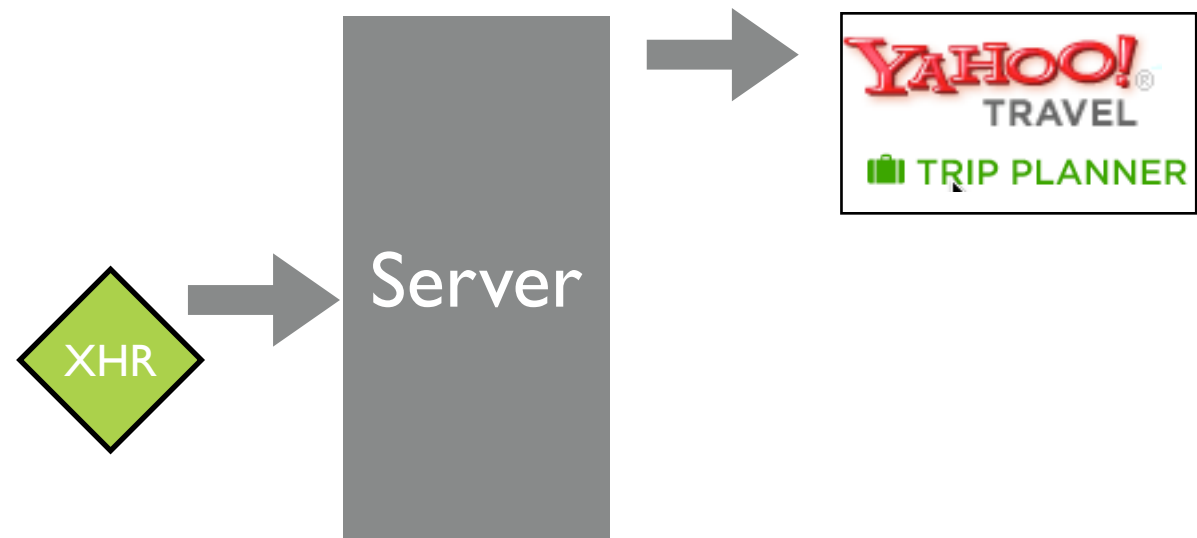
    $('frontdoor').style.display = ''
    $('zip').focus();
    new Rico.Effect.FadeTo( 'frontdoor', .99999, 100, 10, {complete:emptyContents});
}
```


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`





```
{
  "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, Anchorage, Seattle, Vancouver, Ketchikan, Sitka, Juneau, Skagway, Se...",
  "CreateDate": "1130437928",
  "Display": "100x100_cde24409e413d4d6da27953f6a51130437928",
  "Width": "66",
  "Geocode": {
    "Latitude": "49.28276",
    "Longitude": "-123.120499",
    "precision": "not available"
  },
  "Url": "http://us.i1.yimg.com/us.i1.yimg.com/travel.yahoo.com/trip/?pid=351102"
}
```

```
{
  "id": "400333",
  "YahooID": "brdway_grl",
  "Title": "Alaska",
  "Summary": "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-cc605e44b47e20e731d3093b94afa37e.jpg",
    "Height": "75",
    "Width": "100",
    "Latitude": "61.190361",
    "Longitude": "not available",
    "Url": "http://?pid=400333&action=view"},
  ...
}]}
```

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></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.');
```