

CS 683 Emerging Technologies
Fall Semester, 2006
Doc 23 Rails 7 AJAX
Nov 16, 2006

Copyright ©, All rights reserved. 2006 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

script.aculo.us, Common Ajax Javascript library, <http://script.aculo.us/>

Surveying open-source AJAX toolkits, Peter Wayner, Inforworld, July 31, 2006, http://www.inforworld.com/article/06/07/31/31FEajax_1.html

Proprietary AJAX toolkits: The other side of the coin, Peter Wayner, Inforworld, July 31, 2006, http://www.inforworld.com/inforworld/article/06/07/31/31FEajaxsb_1.html

Ajax/DHTML Library Scorecard:How Cross Platform Are They? Musings from Mars ,March 4th, 2006, <http://www.musingsfrommars.org/2006/03/ajax-dhtml-library-scorecard.html>

Wikipedia, http://en.wikipedia.org/wiki/Main_Page

Agile Web Development with Rails 2nd Ed Bl.16 October 25, Thomas & Hanson, The Pragmatic Bookshelf, PDF

Rails API, <http://api.rubyonrails.org/>

Some Ajax Reading

Why Ajax Sucks (Most of the Time) Jacob Nielsen <http://www.usabilityviews.com/ajaxsucks.html>

Ajax SWik <http://swik.net/Ajax>

Ajax Mistakes

How to use XMLHttpRequest

Places to use Ajax

Web Browsers suck

for developing applications

HTML is very limited as UI

Delay in response from server

Reducing Suckiness

CSS

Javascript

XMLHttpRequest

Cascading Style Sheets - CSS

stylesheet language that describes layout of HTML & XML

Provides HTML better
control over layout
UI widgets

Spec maintained by
World Wide Web Consortium (W3C)

```
body { background-color: #fff; color: #333; }
```

```
body, p, ol, ul, td {  
  font-family: verdana, arial, helvetica, sans-serif;  
  font-size: 13px;  
  line-height: 18px;  
}
```

```
pre {  
  background-color: #eee;  
  padding: 10px;  
  font-size: 11px;  
}
```

Javascript

Not related to Java

JScript

Microsoft's version

Scripting language used in

Web browsers

Adobe Acrobat

Mac OS X Widgets

Photoshop

ECMAScript

Standard version of Javascript

Multiple versions and implementations

Different browsers have different implementations

Different versions of a browser have different versions of javascript

Sample Javascript

```
<html>
<head>
  <title>Sample</title>
  <script defer="defer" type="text/javascript">
function go()
  {
  var message='Where do you wish to go?';
  var url = prompt(message, 'http://www.sdsu.edu/');
  window.open(url, 'ObjectNameGoesHere', 'resizable=no');
  }
  </script>
</head>

<body>
<a href="http://www.sdsu.edu" onmouseover="go()">
  Try Me
</a>

</body>
</html>
```

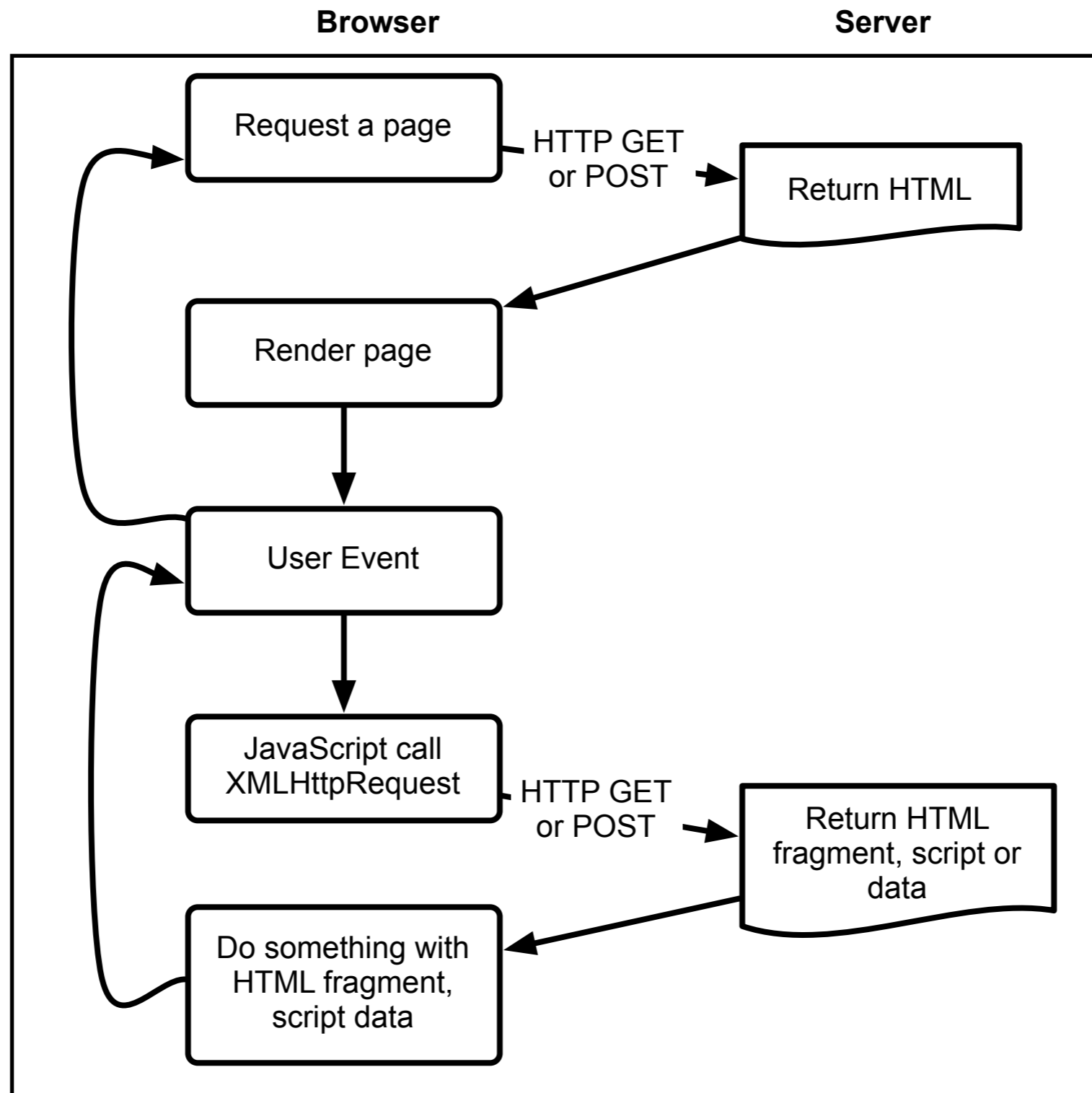
XMLHttpRequest

Set of API function calls allowing web browser scripts to transfer XML data from/to a web server

```
abort()  
getAllResponseHeaders()  
getResponseHeader( headerName )  
open( method, URL )  
open( method, URL, async )  
open( method, URL, async, userName )  
open( method, URL, async, userName, password )  
send( content )  
setRequestHeader( label, value )
```

AJAX

AJAX - Asynchronous JavaScript and XML



Google Maps demo

Issues

HTML is very limited as UI

XMLHttpRequest is low level

Each version of a browser implements different Javascript

AJAX Libraries

Rails Supported

Prototype

Script.aculo.us

RJS templates

Open Source Reviewed in InfoWorld

Dojo

Zimbra Kabuki AJAX Toolkit

Google Web Toolkit

Open Rico

Prototype

Microsoft Atlas (free but not open source)

Yahoo AJAX Library

Commercial Reviewed in InforWorld

General Interface

JackBe

Backbase

IceSoft

Nexaweb

Kapow

Musings from Mars Grade A libraries

Dojo Toolkit

Echo 2

Javascript/Ajax Toolbox

Jitsu

jQuery

jsLINB

MochiKit

Moo.fx

Prototype

Sardalya

Script.aculo.us

Spry

Tacos

TurboWidgets

TwinHelix

Wicket

Yahoo! User Interface Library

Zapatec Ajax Suite

Rails Ajax Support

Prototype

Makes AJAX calls using XMLHttpRequest
Interacts with page DOM

Script.aculo.us

Uses Prototype
Provides UI effects

RJS templates

Retrieves & execute Javascript from server

public/javascripts/

File	Lines
controls.js	815
dragdrop.js	913
effects.js	958
prototype.js	2006

Prototype Example

app/controllers/ajax_controller

```
class AjaxController < ApplicationController
  def time_now
    render :layout => false
  end

  def remoteLink
  end
end
```

Views

app/views/layout/ajax.rhtml

```
<html>
<head>
  <title>Ajax Examples</title>
  <%= javascript_include_tag "prototype" %>
</head>
<body>
  <%= @content_for_layout %>
</body>
</html>
```

app/views/ajax/remoteLink.rhtml

```
<%= link_to_remote( "Click on me",
  :update => 'changeDiv',
  :url => { :action => :time_now}
)%>
<div id="changeDiv">Hi mom</div>
```

app/views/ajax/time_now.rhtml

Hello Ajax

```
<p>The time is now <%= Time.now%>
</p>
<p>Session ID is <%= session.session_id %>
</p>
```

Views Rendered

[Click on me](#)

Hi mom

[Click on me](#)

Hello Ajax

The time is now Thu Nov 09 16:02:02 PST 2006

Session ID is f3c3847b5d45679eb8ac268ecb6c0823

How Does This Work?

app/views/ajax/remote_link.rhtml

```
<%= link_to_remote( "Click on me", :update => 'changeDiv', :url => { :action => :time_now} )%>  
<div id="changeDiv">Hi mom</div>
```

Function Definitions

```
def link_to_remote(name, options = {}, html_options = {})  
  link_to_function(name, remote_function(options), html_options)  
end
```

```
def link_to_function(name, function, html_options = {})  
  content_tag( "a", name,  
    { :href => "#", :onclick => "#{function}; return false;" }.merge(html_options.symbolize_keys))  
end
```

Web Page as Sent to Browser

```
<html>
<head>
  <title>Ajax Examples</title>
  <script src="/javascripts/prototype.js" type="text/javascript"></script>
</head>
<body>
<a href="#" onclick="new Ajax.Updater('changeDiv', '/ajax/time_now',
      {asynchronous:true, evalScripts:true}); return false;">Click on me</a>
<div id="changeDiv">Hi mom</div>

</body></html>
```

Prototype Code

```
public/javascripts/prototype.js
Ajax.Updater = Class.create();

Object.extend(Object.extend(Ajax.Updater.prototype, Ajax.Request.prototype), {
  initialize: function(container, url, options) {
    this.containers = {
      success: container.success ? $(container.success) : $(container),
      failure: container.failure ? $(container.failure) :
        (container.success ? null : $(container))
    }
    this.transport = Ajax.getTransport();
    updateContent: function() {
      var receiver = this.responsesSuccess() ?
        this.containers.success : this.containers.failure;
      var response = this.transport.responseText;

      if (!this.options.evalScripts)
        response = response.stripScripts();

      if (receiver) {
        if (this.options.insertion) {
          new this.options.insertion(receiver, response);
        } else {
          Element.update(receiver, response);
        }
      }
    }
  }
});
```

Getting to Actual XMLHttpRequest

public/javascripts/prototype.js

```
var Ajax = {  
  getTransport: function() {  
    return Try.these(  
      function() {return new ActiveXObject('Msxml2.XMLHTTP')},  
      function() {return new ActiveXObject('Microsoft.XMLHTTP')},  
      function() {return new XMLHttpRequest()}  
    ) || false;  
  },  
};
```

Periodic Updates - Asynchronous

```
app/controllers/ajax_controller
```

```
class AjaxController < ApplicationController
  def ps
    render :text => "<pre>" + CGI::escapeHTML(`ps -a`) + "</pre>"
  end
  def remoteLink
  end
end
```

<http://0.0.0.0:3000/ajax/periodic>

```
app/views/ajax/periodic.rhtml
```

```
<h2>Processes</h2>
<div id="process-list">
</div>

<%= periodically_call_remote(
  :update => 'process-list',
  :url    => { :action => :ps},
  :frequency => 10 )%>
```

Processes

PID	TT	STAT	TIME	COMMAND
686	p1	Ss	0:00.02	login -fp whitney
687	p1	S	0:00.12	-tcsh
694	p1	R+	2:49.01	ruby script/server
1215	p1	R+	0:00.01	ps -a

Forms without Page Update

```
app/controllers/ajax_controller
class AjaxController < ApplicationController
  def checkAnswer
    @guess = params[:guess] || ""
    if @guess.strip.match /^rails$/
      render(:text => 'You got it')
    else
      render :partial => 'quizForm'
    end
  end
end

  def quiz
  end
end
```

Views

```
app/views/ajax/quiz.rhtml
```

```
<h2>Guess!</h2>
```

```
<%= render :partial => 'quizForm'%>
```

```
app/views/ajax/_quizForm.rhtml
```

```
<div id="update_div">
```

```
<% if @guess %>
```

```
  <p><%= h @guess %> is wrong</p>
```

```
<% end %>
```

```
<%= form_remote_tag(
  :update => "update_div",
  :url => { :action => :checkAnswer})%>
```

```
<label for="guess">Ruby on ?</label>
```

```
<%= text_field_tag :guess %>
```

```
<%= submit_tag "Submit answer"%>
```

```
<%= end_form_tag %>
```

```
</div>
```

<http://127.0.0.1:3000/ajax/quiz>

Guess!

Ruby on ?

After incorrect guess

Guess!

ring is wrong

Ruby on ?

After correct guess

Guess!

You got it