

Ajax on Struts:

Coding an Ajax Application with Struts 2




Struts University Series

For the latest version of this presentation,
visit www.StrutsUniversity.org






Abstract

- Struts is Java's most popular web framework. Ajax is the web's hottest user interface. What happens when we put Struts on Ajax?
 - During the session, we will cover
 - Integrating an Ajax UI with Struts 2
 - Using Yahoo User Interface (YUI) Library
 - Using Struts to provide services to Ajax UI
- 




What is Ajax?

- Originally, acronym for "Asynchronous JavaScript And XML."
 - Technique, rather than a framework (such as Struts).
 - Allows web pages to work like GUI apps
 - Already used by Microsoft and Google, among many, many others.
- 





What was life like before Ajax?

- "web page as a flat document"
 - To mimic GUI desktop apps
 - 1 send all data with (a lot of) JavaScript
 - 2 constant form submits back to the server
 - Ajax gives you the best of both worlds:
 - dynamic web pages on the client
 - server-side Java application
- 




How does Ajax work?

- Ajax adds a "background" call to the server
 - 1 XMLHttpRequest requests content asynchronously.
 - 2 Content can be anything; JavaScript updates page.
 - 3 Triggered by JavaScript events: onclick, onchange, onblur, ...
- 




How do we process server-side requests?

- XHR request looks like conventional request
 - Can be handled by any server-side technology
 - PHP, JSP or Java servlets, ASP or .NET handlers
 - Or by remote procedure protocol like JSON-RPC
 - Key point is that result returns the expected payload
 - XML, text, HTML, JSON
- 




What is Apache Struts?

- Free open-source framework for creating Java web applications
 - Provides three major components
 - Request handler
 - Response handler
 - Tag libraries for JSP, Freemarker, and Velocity
- 




Can we use Ajax with a Struts application?

- XHR is just another request/response
 - Struts can stream data as a response
 - Use Ajax JSP tag libraries
 - Use plain-vanilla Ajax libraries
- 





Why use Apache Struts?

- Mature, well-supported, well-understood
 - Provides input validation and data conversion
 - Interacts well with Spring, Hibernate, et al
 - Defacto standard for Java web applications
- 




Where do we start?

- Pure Ajax prototype
 - Test data encapsulated as JSON
 - Later, replace test data with JSON via XHR
 - Separates concerns
- 



How do we select an Ajax library?

- Submit before you commit
 - Code your own sample application
 - Pick the one that works for you and yours
 - We tried Dojo and YUI
- 

Navigation

- ▷ [About](#)
- [Demos](#)
- ▷ [Developer](#)
- ▽ [Documentation](#)
 - ▷ [The Dojo Book, 0.9](#)
 - ▷ [The Dojo Book, 0.4](#)
 - [API](#)
 - [Dojo Porting Guide: 0.4.x to 0.9](#)
 - [Dojo Porting Guide: 0.9.x to 1.0](#)
 - [ShrinkSafe](#)
- [Downloads](#)



0.9: Dojo, Dijit and DojoX

[Getting started](#)
[0.4.x to 0.9 Porting Guide](#)

Plaxo Online Upgrades To 0.9

Sat, 10/06/2007 - 05:32 — alex

As a user of [Plaxo](#), I've been delighted with their [Pulse](#) service for a while now. [Bloglines](#) is a really great start at letting me keep up with things, but Pulse fills an important gap where Bloglines leaves off. Why do I trust my personal data to a service like this?

The Yahoo! User Interface Library (YUI)

Yahoo! UI Library

Home

- YUI Blog
- YUI Discussion Forum
- YUI on Sourceforge
- API Documentation
- YUI Examples Gallery
- Powered by YUI
- YUI Theater

YUI License

YUI Articles

- YUI FAQ
- Graded Browser Support
- Skinning YUI
- Bug Reports/Feature Requests
- Serving YUI Files from Yahoo!
- Security Best Practices

YUI Components

- Animation
- AutoComplete
- Browser History Manager BETA
- Button BETA
- Calendar
- Color Picker BETA

The Yahoo! User Interface Library (YUI)

The Yahoo! User Interface (YUI) Library is a set of utilities and controls, written in JavaScript, for building richly interactive web applications using techniques such as DOM scripting, DHTML and AJAX. The YUI Library also includes several core CSS resources. All components in the YUI Library have been released as open source under a [BSD license](#) and are free for all uses.

Download YUI

Download YUI version 2.3.1, including [full API documentation](#) and [more than 180 functional examples](#) from Sourceforge.

The library's developers blog [at the YUI Blog](#) and the YUI Library community exchanges ideas at [YDN-JavaScript on Yahoo! Groups](#).

Using YUI:

- [FAQ](#)
- [Getting Started](#)
- [Licensing](#)
- [Articles](#)
- [Cheat Sheets](#)
- [Support & Community](#)

YUI Core:

- [The YAHOO Global Object](#) (base requirement for all YUI components)
- [DOM Collection](#) (convenience

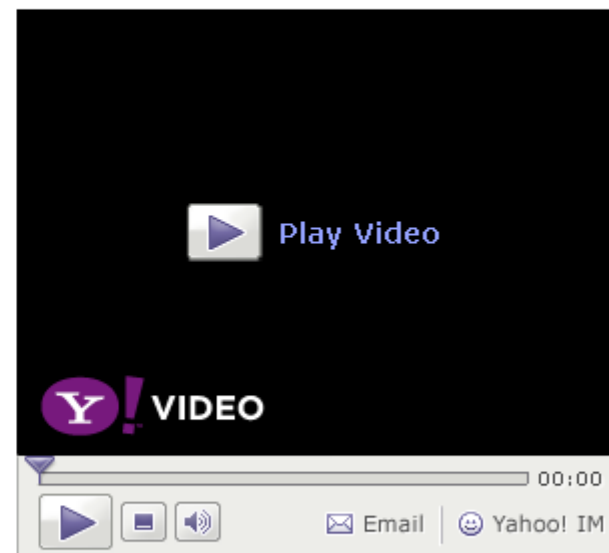
YUI Library Controls/Widgets:

- [AutoComplete](#)
- [Button Control](#) BETA
- [Calendar](#)
- [Color Picker](#) BETA
- [Container](#)


YUI Theater RSS

[\[more videos\]](#)


Steve Souders — "High Performance Web Sites: 14 Rules for Faster Pages"




Steve Souders is Yahoo's chief [performance guru](#) and the author of the popular [YSlow](#) plugin for Firebug; in this talk, he outlines the 14 things you can do to make your web site snappy and responsive. [Available on [Yahoo! Video](#) and as an [MP4 download](#).]




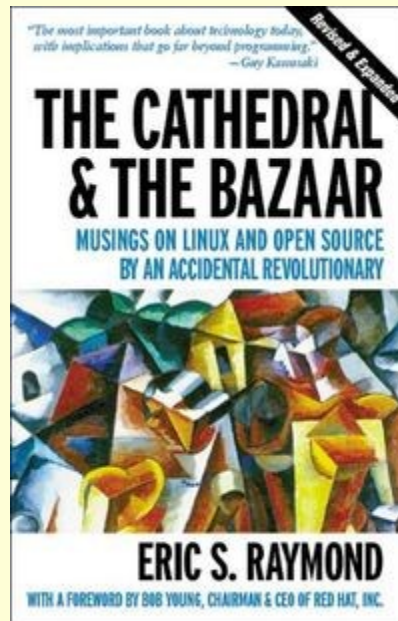
Why Yahoo User Interface (YUI) Library?

- Well documented and supported
 - Lots of working examples
 - BSD license
 - Easy to read code
 - Easy to hack code
- 




Why not Yahoo User Interface (YUI) Library?

- Development team Yahoo! employees
 - No formal mechanism for contributions
 - A cathedral, rather than a bazaar
- 





What's up with Planet Yazaar?

- Yazaar - (Yahoo + Bazaar = Yazaar)
 - Accepts and maintains contributor extensions and documentation
 - Working toward "Soup to nuts" project documentation
 - Public repository with version to version change logs
- 

Planet Yazaar - Additions to the Yahoo! User Interface (YUI) Library

[About](#)[Examples](#)[Yazaar Widgets](#)[Extras](#)[Resources](#)

Planet Yazaar creates and maintains community-supported additions to the [Yahoo User Interface \(YUI\) Library](#) (Yahoo + Bazaar = Yazaar).

Yazaar offers three key services to the YUI community:

- A vehicle for accepting and maintaining contributor extensions in a common location, either in the form of code or documentation
- "Soup to nuts" project documentation, including coding style guidelines and build systems
- A public repository with version to version change logs

Our wish is for the Yazaar community to enjoy the best of both worlds: [the cathedral and the bazaar!](#)

NOTE: The DataForm widget is under active development. At this time, some examples may render correctly only under FireFox. The problem is being addressed.

Planet Yazaar is powered by [YUI 2.2.2!](#)

Yazaar is an independent, volunteer project. Yazaar is not affiliated with Yahoo! or the Yahoo! User Interface (YUI) Library.



What's up with Planet Yazaar?

- Just as an aside ...
 - Yahoo GeoCities for public web site
 - Uses GoogleCode for repository
 - Google Group for mailing list and change logs
 - Yahoo 360 for Blog
- 



yazaar

Additions to the Yahoo User Interface (YUI) Library

ted.husted@gmail.com | [Settings](#) | [What's new?](#) | [Help](#) | [My Account](#) | [Sign out](#)

Search Projects

Search the Web

[Project Home](#)

[Downloads](#)

[Wiki](#)

[Issues](#)

[Source](#)

[Administer](#)

Planet Yazaar creates and maintains community-supported additions to the [Yahoo! User Interface \(YUI\) Library](#) (Yahoo + Bazaar = Yazaar).

Yazaar offers three key services to the YUI community:

- A vehicle for accepting and maintaining contributor extensions in a common location, either in the form of code or documentation
- 'Soup to nuts' project documentation, including as coding style guidelines and build systems
- A public repository with version to version change logs

Our wish is for the Yazaar community to enjoy the best of both worlds: [the cathedral and the bazaar!](#)

Planet Yazaar is an independant, volunteer project. Yazaar is not affiliated with Yahoo! or the Yahoo! User Interface (YUI) Library.

License: [New BSD License](#)

Labels: [Ajax](#), [YUI](#), [JavaScript](#), [widgets](#)

Featured Wiki Pages:

[Show all](#)

[Charter](#)

[CommonProblems](#)

[Conventions](#)

[HappyTrails](#)

[Resources](#)

Links: [Our Website](#)

[YUI Site](#)

[Anvil Business Logic Framework](#)

Blogs: [Our Blog](#)

[YUI Blog](#)

[Ted Husted](#)

Groups: [Development Group](#)



Yazaar Development

Search this group

Search Groups

Home

[\[edit welcome message\]](#)Discussions 10 of 425 messages [view all »](#)[+ new post](#)[r351 - in trunk/examples: contacts.struts-pim](#)

By codesite-nore...@google.com - Oct 10 - 1 author - 0 replies

[r350 - in trunk/examples/contacts: java/actions/webapp/WEB-INF/webapp/WEB-INF/lib/webapp/WEB-INF/...](#)

By codesite-nore...@google.com - Oct 10 - 1 author - 0 replies

[r349 - in trunk/examples/contacts: .java/java/actions/webapp/webapp/WEB-INF/webapp/WEB-INF/lib w...](#)

By codesite-nore...@google.com - Oct 10 - 1 author - 0 replies

[r348 - trunk/examples/contacts](#)

By codesite-nore...@google.com - Oct 10 - 1 author - 0 replies

[r347 - in trunk/examples/contacts/webapps: .WEB-INF/WEB-INF/lib/WEB-INF/results](#)

By codesite-nore...@google.com - Oct 10 - 1 author - 0 replies

[r346 - in trunk/examples/contacts: .java/java/actions](#)

By codesite-nore...@google.com - Oct 10 - 1 author - 0 replies

[Issue 27 in yazaar: Add Contacts example using Struts 2 on the backend](#)

By codesite-nore...@google.com - Oct 10 - 1 author - 0 replies

[r345 - in trunk: .examples/dataform](#)

By codesite-nore...@google.com - Oct 3 - 1 author - 0 replies

Home


[Discussions](#)[Members](#)[Pages](#)[Files](#)[About this group](#)[Edit my membership](#)[Group settings](#)[Management tasks](#)[Invite members](#)

Group info

Members: 7**Activity:** Low activity**Language:** English**Group categories:** *Not categorized*[add a category](#)[More group info »](#)



Planet Yazaar



Planet Yazaar creates and maintains community-supported additions to the Yahoo User Interface (YUI) Library.

View: [Text & Photos](#) | [Photos only](#) | [Text only](#)

Entries: 1 - 1 of 1 [First](#) | [< Prev](#) | [Next >](#) | [Last](#)

Welcome to the Yazaar!


[1 Apr 2007] Planet Yazaar creates and maintains community-supported additions to the [Yahoo! User Interface \(YUI\) Library](#) (Yahoo + Bazaar = Yazaar). Yazaar offers three key services to the YUI community:

- A vehicle for accepting and maintaining contributor extensions in a common location, either in the form of code or documentation
- 'Soup to nuts' project documentation, including as coding style guidelines and build systems
- A public repository with version to version change logs

Our wish is for the Yazaar community to enjoy the best of both worlds: [the cathedral and](#)




What are we going to code?

- 1 Provide a list of employees
 - 2 Record employee's first name, last name, extension, username, hire date, and hours worked per week.
 - 3 All record to be updated by authorized users
 - 4 Allow list to be filtered by first name, last name, or username.
 - 5 Allow full or filtered list to be sorted by any field
- 



What are we going to code?

- Use YUI DataTable to display list
 - Share DataTable record with data-entry form
 - Use TabView to separate list and edit
- 

[Welcome](#)[Contacts](#)[Notes](#)[Log](#)[List Entries](#)[Edit Entry](#)<< < 1 **2** 3 > >> 2

Contact List

First Name	Last Name	Extension	Username	Hire Date	Hours
Zaphod	Beeblebrox	555-123-4565	zaphie	04/01/1978	-1
Vranx	Yooden	555-123-4566	conker	04/01/1978	37.5

<< < 1 **2** 3 > >> 2Filter Entries by:

[Welcome](#)[Contacts](#)[Notes](#)[Log](#)[List Entries](#)[Edit Entry](#)**First Name** **Last Name** **Extension** **Username** **Hire Date** **Hours**



What are we going to code?

- Prototype with static data
- Request data from Struts 2 Action
 - Use Struts 2 Plugin to return JSON






How does the Struts 2 JSON plugin work?

- Exposes action properties as fields in a JSON record.

```
{  
  stringValue : "A string value",  
  intArray: [10, 20],  
  map: {  
    Zaphod : "Just this guy, you know",  
    Arthur : "Monkey-boy"  
  },  
  record: "start a named record"  
}
```



```
public class ExampleOutputAction {

    private String stringValue = "A string value";
    public String getStringValue() {
        return stringValue;
    }
    public void setStringValue(String value) {
        stringValue = value;
    }

    private int[] intArray = {10, 20};
    public int[] getIntArray() {
        return intArray;
    }
    public void setIntArray(int[] value) {
        intArray = value;
    }

    private Map map = new HashMap();
    public Map getMap() {
        return map;
    }
    public void setMap(Map value) {
        map = value;
    }
}
```

```
private String nextRecord = "start a named record";
@JSON(name="record")
public String getNextRecord() {
    return nextRecord;
}

//'transient' fields are not serialized
@SuppressWarnings("unused")
private transient String stayHome;

//fields without getter method are not serialized
@SuppressWarnings("unused")
private String noGetterForMe;

public String execute() {
    map.put("Zaphod", "Just this guy, you know");
    map.put("Arthur", "Monkey-Boy");
    return "success";
}
}
```

[Welcome](#)

[Contacts](#)

[Notes](#)

[Log](#)

Welcome to the PIM example application.

- [About the Application Architecture ...](#)

File Download

Do you want to save this file?



Name: example-output.do

Type: Unknown File Type, 148 bytes


From: 192.168.42.103

Save

Cancel




While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not save this file. [What's the risk?](#)



```
{
  "intArray": [10, 20],
  "map": { "Arthur": "Monkey-Boy",
           "Zaphod": "Just this guy, you know" },
  "record": "start a named record",
  "stringValue": "A string value"
}
```

```
{
  stringValue : "A string value",
  intArray: [10, 20],
  map: {
    Zaphod : "Just this guy, you know",
    Arthur : "Monkey-boy"
  },
  record: "start a named record"
}
```





```
package actions;
```

```
import java.util.Map;
```

```
import java.util.HashMap;
```

```
import com.googlecode.jsonplugin.annotations.JSON;
```

```
import org.texturemedia.smarturls.ParentPackage;
```

```
import org.texturemedia.smarturls.Result;
```

```
@ParentPackage("json-default")
```

```
@Result(name="success", type="json", location="")
```


```
@SuppressWarnings("unchecked")
```

```
public class ExampleOutputAction {
```


```
    private String stringValue = "A string value";
```

```
    // ...
```





How can we code a UI using sample data?

- Entry list
 - Entry form
 - List shares data with form
- 

[Welcome](#)[Contacts](#)[Notes](#)[Log](#)[List Entries](#)[Edit Entry](#)<< < 1 **2** 3 > >> 2

Contact List

First Name	Last Name	Extension	Username	Hire Date	Hours
Zaphod	Beeblebrox	555-123-4565	zaphie	04/01/1978	-1
Vranx	Yooden	555-123-4566	conker	04/01/1978	37.5

<< < 1 **2** 3 > >> 2Filter Entries by:

[Welcome](#)[Contacts](#)[Notes](#)[Log](#)[List Entries](#)[Edit Entry](#)**First Name** **Last Name** **Extension** **Username** **Hire Date** **Hours**

Is there an Ajax architecture?

```
<script src="my.js" type="text/javascript">
</script>


if (typeof parent.MY != "undefined") {
    var MY = parent.MY; // Prototype namespace
    var my = parent.my; // Variable namespace
} else var
    var MY = {};
    var my = {};
    var MY.Event = {
        // ...
    }
    var my.oEvent = {
        // ... instantiate rest of my and MY
```



Is there an Ajax architecture?

```
<script src="contacts.js" type="text/javascript">  
</script>
```

```
MY.Contacts = function() {  
    MY.Contacts.superclass.constructor.call(this);  
};  
YAHOO.lang.extend(MY.Contacts, YAHOO.yazaar.flev-base);
```






What's a FLEV widget?

- Common business workflow is
 - Find / List / Edit / View
 - Or "FLEV"
- The FLEV widget defines one columnset and datasource to use with all four presentations







```
MY.Contact.prototype.oColumnHeaders = [
  {key:"first_name", text:"First Name",
  sortable:true, resizable:true,
  editor:"textbox", formClassName: "required",
  formTitle: "Enter employee's first name"},

  // more entries


];
```

- key and text are shared attributes
 - sortable, resizable are List attribute
 - formClassName, formtitle are Edit attributes (validation)
- 



```
MY.Contact.prototype.oResponseSchema = {
  fields: ["id", "last_name", "first_name",
    "extension", "username", "hired", "hours",
    "editor"]
};
```


```
MY.Contact.prototype.LOCAL_DATA = {
  result : [{id: 'c5b6bbb1-66d6-49cb-9db6-
743af6627828', last_name: 'Beeblebrox ',
first_name: 'Zaphod ', extension: '555-123-
4565', username: 'zaphie ', hired:
'04/01/1978', hours: -1, editor: '1'},
  // ...
];
```





```
my.oEvent.createEvent("contactLoad")
```

```
my.oEvent.onContactLoadReturn = function(oData)
{
  my.info("Firing contactLoad Event");
  my.oEvent.fireEvent("contactLoad", oData);
};
```



```
<script type="text/javascript">
var fnReady = function() {
  YAHOO.widget.Logger = my.Logger;
  var oView = new MY.View("elBody", "contact");
  _Self = new MY.Contact();
  oView.oContent = _Self;
  my.oEvent.subscribe("contactLoad", _Self.load, _Self);
  my.oEvent.onContactLoadReturn(_Self.LOCAL_DATA);
};
YAHOO.util.Event.onContentReady("elBody", fnReady);
</script>
```


[Welcome](#)[Contacts](#)[Notes](#)[Log](#)[List Entries](#)[Edit Entry](#)<< < 1 **2** 3 > >> 2


Contact List

First Name	Last Name	Extension	Username	Hire Date	Hours
Zaphod	Beeblebrox	555-123-4565	zaphie	04/01/1978	-1
Vranx	Yooden	555-123-4566	conker	04/01/1978	37.5

<< < 1 **2** 3 > >> 2Filter Entries by:



Can we switch to server-side?

- Create an example Action that returns example data
 - Create a database Action that returns persistent data
- 

```
public class ContactLoadAction {
private List contactLoad = new ArrayList();
@JSON(name="result")
public List getContactLoad() {
    contactLoad.add(new Contact(
        "c5b6bbb1-66d6-49cb-9db6-743af6627828",
        "Beeblebrox",
        "Zaphod",
        "555-123-4565",
        "zaphie",
        "04/01/1978",
        "-1",
        "1"
    ));
    // ...
    return contactLoad;
}
```

```
public class Contact {

    public Contact() {};

    public Contact(String id, String last_name, String first_name,
String extension, String username, String hired, String hours,
String editor) {
        setId(id);
        setLast_name(last_name);
        setFirst_name(first_name);
        setExtension(extension);
        setUsername(username);
        setHired(new Date(hired));
        setHours(new Double(hours));
        setEditor(editor);
    }

    private String id;
    public void setId(String value) {
        id = value;
    }
    public String getId() {
        return id;
    }

    // more properties ...
```


[Welcome](#)

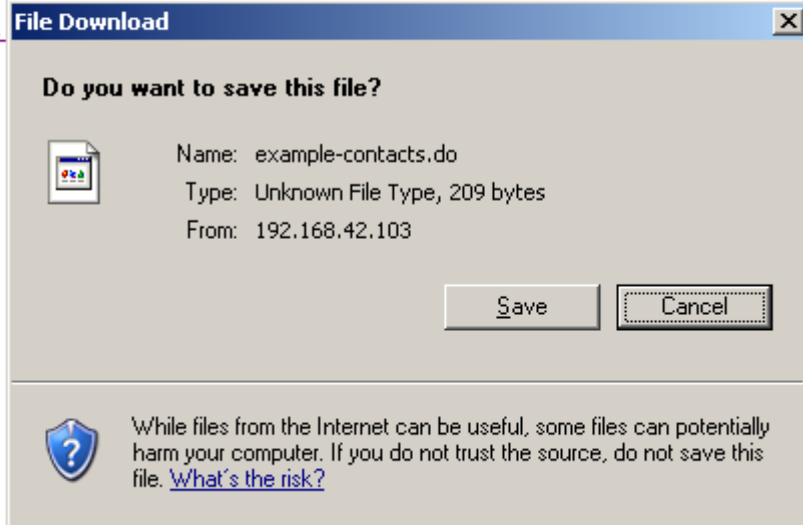
[Contacts](#)


[Notes](#)

[Log](#)

Welcome to the PIM example application.

- [About the Application Architecture ...](#)






How can we switch over to server-side data?

Action data:

```
{"result": [{"editor": "1", "extension": "555-123-4565", "first_name": "Zaphod ", ...
}]}
```

JavaScript data:

```
MY.Contact.prototype.LOCAL_DATA =
{result: [{id: 'c5b6bbb1-66d6-49cb-9db6-743af6627828', last_name: 'Beeblebrox ', first_name: 'Zaphod ' ...
}]}
```



Yahoo! UI Library: Connection Manager

Yahoo! UI Library

- Home
- YUIBlog
- YUI Discussion Forum
- YUI on Sourceforge
- API Documentation
- YUI Examples Gallery
- Powered by YUI
- YUI Theater
- YUI License

YUI Articles

- YUI FAQ
- Graded Browser Support
- Skinning YUI
- Bug Reports/Feature Requests
- Serving YUI Files from Yahoo!
- Security Best Practices

YUI Components

- Animation
- AutoComplete
- Browser History Manager BETA
- Button BETA
- Calendar
- Color Picker BETA
- Connection Manager**

Connection Manager

The Connection Manager is a utility that enables you to make in-page HTTP requests through a simplified interface to the XMLHttpRequest object. The Connection Manager handles cross-browser instantiation of XMLHttpRequest, negotiates the server response and uses a callback pattern to process the response data.

On This Page:

- [Getting Started](#)
- [Using Connection Manager](#)
- [Support & Community](#)
- [Filing Bugs and Feature Requests](#)

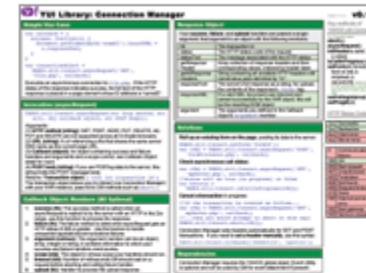
Quick Links:

- [Examples](#): Explore examples of the Connection Manager in action.
- [API Documentation](#): View the full API documentation for the Connection Manager.
- [Release Notes](#): Detailed change log for the Connection Manager.
- [License](#): The YUI Library is issued under a BSD license.
- [Download](#): Download the Connection Manager as part of the full YUI Library on SourceForge.

Getting Started

To use Connection Manager, include the source file and its dependency in your web page with the script tag:

Connection Manager Cheat Sheet:



[Download full set of cheat sheets.](#)

Connection Manager Examples:

- [Connection Manager GET Transaction](#)
- [Connection Manager POST Transaction](#)
- [Connection Manager Transaction Timeout](#)
- [Retrieving a Yahoo! Weather RSS Feed](#)
- [Subscribing to Connection Manager Global Events](#)
- [Subscribing to Connection Manager's Custom Events via the Callback Object](#)

Other YUI Examples That Make Use of the Connection Manager:

<http://developer.yahoo.com/yui/connection/>

```
<script>
var transaction = YAHOO.util.Connect.asyncRequest(
    'POST', sUrl, callback, null);
</script>
```

```
var callback =
{
    success: function(o) { /*success handler code*/ },
    failure: function(o) { /*failure handler code*/ },
    argument: [argument1, argument2, argument3]
}
```

```
var responseSuccess = function(o) {
/* Please see the Success Case section for more
* details on the response object's properties.
* o.tId
* o.status
* o.statusText
* o.getResponseHeader[ ]
* o.getAllResponseHeaders
* o.responseText
* o.responseXML
* o.argument
*/
};
```

```
// my.oEvents.onContactLoadReturn(_Self.LOCAL_DATA);
YAHOO.util.Connect.asyncRequest('POST', "contact-
load.do", callback, null);

var callback = {
    success : function(o) {
        my.oEvent.onContactLoadReturn(o.responseText);
    }
};
```


[Welcome](#)[Contacts](#)[Notes](#)[Log](#)[List Entries](#)[Edit Entry](#)

Contact List

First Name	Last Name	Extension	Username	Hire Date	Hours
----------------------------	---------------------------	---------------------------	--------------------------	---------------------------	-----------------------


Loading data...


Filter Entries by: ▼



```
{ "result": [  
  { "editor": "1", "extension": "555-123-  
4565", "first_name": "Zaphod ", ...  
}] }
```

```
MY.Contact.prototype.LOCAL_DATA =  
{result: [
```






```
// my.oEvents.onContactLoadReturn(_Self.LOCAL_DATA);
YAHOO.util.Connect.asyncRequest('GET', "contact-load.do",
callback, null);
```

```
var callback = {
    success : function(o) {
        var payload = eval("(" + o.responseText + ")");
        my.oEvent.onContactLoadReturn(payload);
    }
};
```

```
- var payload = eval("(" + o.responseText + ")") ;
+ var payload = JSON.eval(o.responseText);
```






What about JavaScript Hijacking?

```
/* {"result":[{"editor":"1","extension":"555-123-4565",  
...  
}] } */
```

```
var data = o.responseText;  
var payload =  
eval (" (" + data.substring (data.indexOf ("\/\*") + 2 ,  
    data.lastIndexOf ("*\//") + ") " ) );  
my.oEvent.onContactLoadReturn (payload) ;
```



What about JavaScript Hijacking?

```
<package name="my-default" extends="json-default">
  <result-types>
    <result-type name="json"
      class="com.googlecode.jsonplugin.JSONResult"
      default="true">
      <param name="wrapWithComments">true</param>
    </result-type>
  </result-types>

  <action name="contact-load"
    class="actions.ContactLoadAction">
    <result />
  </action>
</package>
```

[Welcome](#)[Contacts](#)[Notes](#)[Log](#)[List Entries](#)[Edit Entry](#)<< < 1 **2** 3 > >> 2

Contact List

First Name	Last Name	Extension	Username	Hire Date	Hours
Zaphod	Beeblebrox	555-123-4565	zaphie	04/01/1978	-1
Vranx	Yooden	555-123-4566	conker	04/01/1978	37.5

<< < 1 **2** 3 > >> 2Filter Entries by:



What about error handling?

```
var callback = {  
  success : function(o) {  
    var payload = eval("(" + o.responseText + ")");  
    my.oEvent.onContactLoadReturn(payload);  
  }  
};
```

```
YAHOO.util.Connect.asyncRequest('POST', "contact-  
load.do", callback, null);
```





What about error handling?


```
var callback =  
{  
  success: function(o) { /*success handler code*/ },  
  failure: function(o) { /*failure handler code*/ },  
  argument: [argument1, argument2, argument3]  
}
```





How does JSON-RPC handle errors?

```
{  
  "version" : "1.1",  
  "error" : {  
    "name" : "JSONRPCError",  
    "code" : 123,  
    "message" : "An error occurred parsing the request object.",  
    "error" : {  
      "name" : "JSONError",  
      "message" : "Bad array",  
      "at" : 42,  
      "text" : "{\"id\":1,\"method\":\"sum\",\"params\":[1,2,3,4,5]}"  
    }  
  }  
}
```






How does Struts handle exceptions?

```
<global-results>  
  <result name="error" />  
</global-results>
```

```
<global-exception-mappings>  
  <exception-mapping  
    exception="java.lang.Exception" result="error" />  
</global-exception-mappings>
```



How does Struts handle exceptions?

```
<h2>An unexpected error has occurred</h2>
```

```
<p>Please report this error to your system administrator or appropriate technical support personnel. Thank you for your cooperation.</p>
```

```
<hr/>
```

```
<h3>Error Message</h3>
```

```
<p>
```

```
<s:property value="{exception.message}"/>
```

```
</p>
```

```
<hr/>
```

```
<h3>Technical Details</h3>
```

```
<p>
```

```
<s:property value="{exceptionStack}"/>
```

```
</p>
```


How does Struts handle exceptions?

```
public String getExceptionMessage() {
    ActionContext context = ActionContext.getContext();
    Object value = context.getValueStack().
        findValue("exception.message");
    if (value==null) return null;
    return value.toString();
}
```

```
public String getExceptionStack() {
    ActionContext context = ActionContext.getContext();
    Object value = context.getValueStack().
        findValue("exceptionStack");
    if (value==null) return null;
    return value.toString();
}
```



How does Struts handle exceptions?

```
public String execute() throws Exception {  
    throw new Exception("Whoops!");  
    // return "success";  
}
```

- `getMessage` : "Whoops!"
 - `getExceptionStack`: "java.lang.exception
Whoops! at actions.ContactLoadAction
..."
-



How do widgets handle errors?

```
<!-- ... -->
```

```
</div>
```

```
<div id="elError" class="error"></div>
```

```
<div
```

```
<!-- ... -->
```



How do widgets handle errors?


```
my.asyncRequestException = function (o) {
  alert("Error Communicating with Server! See
message area for details.");
  var sTemplate = "<table>
  <tr><th>Message: &nbsp;</th>
  <td>{exceptionMessage}</td></tr>
  <tr><th>Location:</th>
  <td>{exceptionStack}</td></tr>
</table>";
  var oPayload = eval("(" + o + ")");
  document.getElementById("elError").innerHTML =
  sTemplate.supplant(oPayload);
```



How do widgets handle errors?

Message: Whoops!

Location: java.lang.Exception: Whoops! at actions.ContactLoadAction.execute(ContactLoadAction.java:62) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at





What about error handling?

```
var fnReady = function() {
  YAHOO.widget.Logger = my.oLogger;
  var oView = new MY.View("elBody", "Contacts");
  _Self = new MY.Contact();
  oView.oContent = _Self;
  my.oEvent.subscribe("contactLoad", _Self.load, _Self);
  // YAHOO.util.Connect.asyncRequest('POST', "contact-
    load.do", callback, null);
  my.asyncRequest("contact-load.do",
    my.oEvent.onContactLoadReturn);
};
```



```
my.asyncRequest = function (sAction, fnCallback) {
  return YAHOO.util.Connect.asyncRequest("POST", sAction,
    {
      success : function(o) {
        var oPayload = eval("(" + o.responseText + ")") ;
        if (oPayload.error) {
          my.errorMessage(oPayload.error.message,
            oPayload.error.stackTrack) ;
          return;
        }
        fnCallback(oPayload) ;
      },
      failure : function(o) {
        my.error(o.responseText) ;
      }
    }
  );
};
```

```
my.asyncRequest = function (sAction, fnCallback) {
  return YAHOO.util.Connect.asyncRequest("POST", sAction,
    {
      success : function(o) {
        var oPayload = eval("(" + o.responseText + ")") ;
        if (oPayload.error) {
          my.errorMessage(oPayload.error.message,
            oPayload.error.stackTrack) ;
          return;
        }
        if (oPayload.exceptionMessage) {
          my.asyncRequestException(o.responseText) ;
        }
        fnCallback(oPayload) ;
      },
      failure : function(o) {
        my.asyncRequestError(o.responseText) ;
      }
    }
  );
};
```



```
my.errorMessage = function (sMessage, sStackTrace) {
    alert("Error Communicating with Server! // ...
    var sTemplate = "<table><tr> // ...
    var oContext = {message: sMessage,
        stackTrace: sStackTrace};
    document.getElementById("elError").innerHTML =
        sTemplate.supplant(oContext);
    my.isMessage(true);
    my.error(oContext);
};

my.asyncRequestError = function (o) {
    var oPayload = eval("(" + o + ")");
    my.errorMessage(oPayload.error.message,
        oPayload.error.stackTrace);
};

my.asyncRequestException = function (o) {
    var oPayload = eval("(" + o + ")");
    my.errorMessage(oPayload.exceptionMessage,
        oPayload.exceptionStack);
};
```

[Welcome](#)

[Contacts](#)

[Notes](#)

[Log](#)

[List Entries](#)

[Edit Entry](#)

Filter Entries by:

Microsoft Internet Explorer



Error Communicating with Server! See message area for details.

OK

[Welcome](#)[Contacts](#)[Notes](#)[Log](#)[List Entries](#)[Edit Entry](#)

<< < 1 > >> 2

Contact List

First Name	Last Name	Extension	Username	Hire Date	Hours
Zaphod	Beeblebrox	555-123-4565	zaphie	1978-04-01T00:00:00	-1

<< < 1 > >> 2

Filter Entries by: **Message:** Whoops!**Location:** java.lang.Exception: Whoops! at actions.ContactLoadAction.execute(ContactLoadAction.java:62) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at



Is that all there is?

- During this session, we covered
 - Integrating an Ajax UI with Struts 2
 - Using Yahoo User Interface (YUI) Library
 - Using Struts to provide services to Ajax U





Struts University Series