

AJAX

AJAX is an acronym for Asynchronous JavaScript and XML. Essentially, a JavaScript can make a HTTP request and update portions of a page directly, without going through a conventional POST or GET and refreshing the entire page. Better yet, a page can contain several JavaScripts making simultaneous (asynchronous) requests.

The key point is that when a script makes an "Ajax request" (XHR), the server doesn't know it came from a script, and handles it like any other request. One reason Ajax is so successful is that it works just fine with existing server technologies, including Struts.

It's not the Ajax request that is different, but the Ajax response. Instead of returning an entire page for the browser to display (or redisplay), an Ajax response will just return a portion of a page. The response can take the form of XML, or HTML, or plain text, another script, or whatever else the calling script may want.

Both Struts 1 and Struts 2 can return any type of response. We are not limited to forwarding to a server page. In Struts 1, you can just do something like:

```
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("Hello World! This is an AJAX response from a Struts
Action.");
out.flush();
return null;
```

In Struts 2, we can do the same thing with a Stream result.

There are easier ways!

Using a Struts 2 plugin (e.g., [JSON plugin](#), [jQuery plugin](#), etc.) is, in general, preferred to writing the response directly from within an action. See sections following this for further details.

Struts 2 Stream result Action

```
package actions;

import java.io.InputStream;
import java.io.StringBufferInputStream;
import com.opensymphony.xwork2.ActionSupport;

public class TextResult extends ActionSupport {
    private InputStream inputStream;
    public InputStream getInputStream() {
        return inputStream;
    }

    public String execute() throws Exception {
        inputStream = new ByteArrayInputStream("Hello World! This is a
text string response from a Struts 2 Action.".getBytes("UTF-8"));
        return SUCCESS;
    }
}
```

Struts 2 Configuring the TextResult Action

```
<action name="text-result" class="actions.TextResult">
  <result type="stream">
    <param name="contentType">text/html</param>
    <param name="inputName">inputStream</param>
  </result>
</action>
```

Struts 2 can also return a JSON (JavaScript Object Notation) response, using a [plugin](#).

On the client side, there are two basic strategies, which can be mixed and matched.

First, you can use some type of JSP tag. Here, you don't have to know very much at all about Ajax or JavaScript. The taglib does all the work, and you just have to figure out how to use the taglib. The standard Struts 2 taglib includes several [Ajax JSP tags](#), and many third-party libraries are available, including:

- [Ajax Tags](#)
- [AjaxParts Taglib](#)
- [ColdTags Suite](#)
- [Prize Tags](#)
- [JSON-taglib](#)

Alternatively, you can use a plain-old Ajax widget on a plain-old HTML page, using libraries like [Dojo](#), [jQuery](#), or [YUI](#), and the [StreamResult](#) or the [JSON Plugin](#). Here, the sky's the limit, but you actually have to learn something about JavaScript as a language.

Ajax Plugins

While Struts works fine with Ajax out-of-the-box, for added value, several Ajax-centric plugins are available.

Ajax Tag Plugins

- **jQuery** - The [jQuery Plugin](#) provide ajax functionality and UI Widgets an JavaScript Grid based on the jQuery javascript framework.
- **Ajax Parts** - The [AjaxParts Taglib \(APT\)](#) is a component of the Java Web Parts (JWP) project (<http://javawebparts.sourceforge.net>) that allows for 100% declarative (read: no Javascript coding required!) AJAX functionality within a Java-based webapp.
- **Dojo** - The [Ajax Tags Dojo Plugin](#) was represented as a theme for Struts 2.0. For Struts 2.1, the Dojo tags are bundled as a plugin until version 2.3.x. Since version 2.5 this plugin is not part of th Struts2 distribution anymore
- **YUI** - The [Yahoo User Interface \(YUI\) Plugin](#) has only a few tags are available so far, but the YUI tags tend to be easier to use than the Dojo versions.

Other Ajax Plugins

- **Ajax File Upload** - With the [Ajax File Upload Plugin](#) we can upload a file to the server and asynchronously monitor its progress.
- **GWT** - The [Google Web Toolkit Plugin](#) exposes Struts 2 actions to the GWT RPC mechanism.
- **JSON** - The [JSON Plugin](#) serializes Actions properties into JSON, making it easy to respond to JavaScript requests.

See the [Struts Plugin Repository](#) for a complete list of Struts 2 plugins.

Ajax Results with JSP

While server pages are most often used to generate HTML, we can use server pages to create other types of data streams. Here's an example:

book.jsp

```
<%@ page import="java.util.Iterator,
                java.util.List,
                com.esolaria.dojoex.Book,
                com.esolaria.dojoex.BookManager" %>

<%
    String bookIdStr = request.getParameter("bookId");
    int bookId = (bookIdStr == null || "".equals(bookIdStr.trim()))
        ? 0 : Integer.parseInt(bookIdStr);
    Book book = BookManager.getBook(bookId);
    if (book != null) {
        out.println(book.toJSONString());
        System.out.println("itis: " + book.toJSONString());
    }
%>
```

In the code example, we use `System.out.println` to return a JSON data stream as the response. For more about this technique, see the article [Using Dojo and JSON to Build Ajax Applications](#).

Next: Dependency Injection