

Session: Struts Best Practices
by James Mitchell



EdgeTech, Inc.
www.edgetechservices.net
678.910.8017

*Consulting, Training, Mentoring, and
Project recovery*

Introducing the speaker: James Mitchell

- Independent contractor with clients in Atlanta, California, and New York.
- Has been writing Web Applications for 11 years in the E-commerce, Government, Banking and Financial Services industries.
- Provide specialized services and training for companies needing team development training, mentoring, and/or project recovery.
- Serves as a member of the Struts and Jakarta PMC (Project Management Committee) and is one of a handful of active committers on the Apache Struts and several of the Jakarta Commons subprojects.

Introducing the speaker: James Mitchell

- Founder and President of Struts-Atlanta, an all volunteer group of Atlanta, GA – based Struts enthusiasts. (Monthly meetings resume in June 2005)
- President of newly created Atlanta JBoss Users Group. Next meeting June 2, 2005 (same location as last one)
- Proud father (4 children ranging from 5 to 15)
- Soccer coach (Dacula Soccer Club - u6, u10)
- Cub Scouts (Den leader, Asst Cub Master, Web Site chair for Pack and District)

Overview

For more than 4 years, Struts has enjoyed enormous success. From one man's vision to a de facto standard, no one would argue that Struts has filled a critical need at a critical time. However, as new frameworks and technologies emerge, is Struts still the right choice for your J2EE web projects?

This session briefly introduces Struts and quickly moves to cover some advanced topics that are considered by many to be “best practices” concerning web application development with the Apache Struts framework.

A quick poll of the audience

- Who is using (or has used) Struts?
- What else are you using?
- How do you like other frameworks compared to Struts?

Introduction to Apache Struts

- Apache Struts is a Java Servlet-based Web Development Framework that implements the V and C portion of the Model-View-Controller (MVC) design pattern*.
- Struts is developed as a top level project at the Apache Software Foundation.

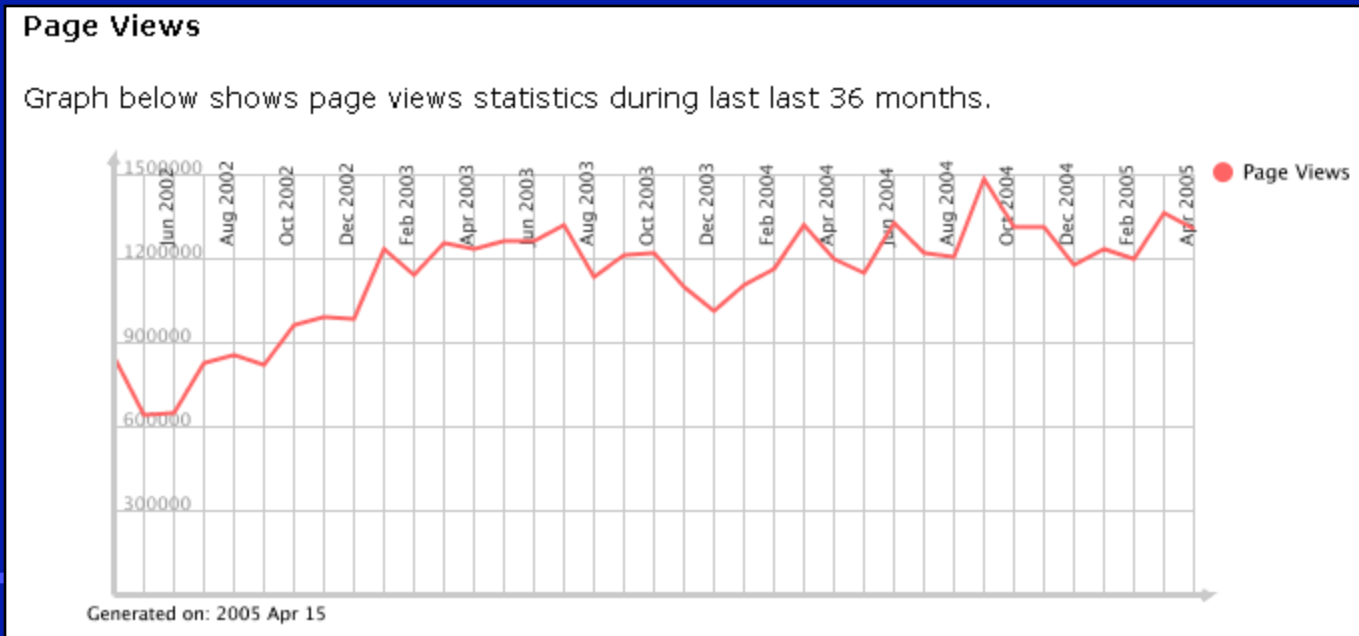
* <http://java.sun.com/blueprints/patterns/MVC.html>

Apache Struts - past

- Struts was originally created by Craig McClanahan and donated to Apache in 2000.
- Struts 1.0 - 15 June, 2001
- Struts 1.0.1 - 12 Jan 2002
- Struts 1.0.2 - 11 Feb 2002
- Struts 1.1 - 30 Jun 2003
- Struts 1.2.2 - 31 Aug 2004
- Struts 1.3.x - ???
- Struts 2.0 ??? Shale ???
- So when's the next release?

Apache Struts - present

- Since Jan 2003, the Struts web site has an average sustained rate of between 1.2 and 1.5 million page views per month. (This includes page views for documentation and API reference)



Apache Struts - present

- The download rate for Struts distributions currently hovers around 30,000 downloads per month, but this estimate does not include stats for Apache mirrors (which are the preferred method of distribution)
- Struts became a Top Level Project (TLP) under Apache.
- The Struts code repository was moved from CVS to Subversion (SVN).
- *Martin Cooper* was appointed as the new PMC chair

Apache Struts - present

- Refactoring the build process.
 - Currently all of Struts (as you knew it) can be built with Maven.
 - Changes for Releases
 - The release policy for Struts has changed over the last year. (Apache Tomcat style)

Apache Struts - present

- Current Release
 - The current stable (GA) build for Apache Struts is 1.2.4
- Nightly builds
 - The Apache Struts nightly builds for both 1.2.x (1.2 branch) and 1.3.x (head) are available at:
<http://svn.apache.org/builds/struts/maven/>

Apache Struts - present

- Struts project was split into multiple subprojects

apps - Sample Struts applications

bsf - Bean Scripting framework

build- Shared build directory

core - Core Struts classes

el - A port of the Standard Tag Libraries to support el expressions as attributes

faces- An extension library to allow Struts and JSF to play nice

flow - An extension for implementing workflow with Struts

Apache Struts - present

- Struts project was split into multiple subprojects
 - plugins** - Subproject for custom Components implemented as Struts plugins
 - sandbox** - A place to experiment with components
 - shale** - Recently adopted as a Struts subproject, Shale builds on the concepts and advantages provided by JSF
 - taglib** - The Struts Tag Libraries
 - tiles** - The Struts Tiles subproject (does anyone **NOT** know what Tiles is?)

Apache Struts - present

- So when is the next release?

Struts 1.2.7 - Right now !!!!

Apache Struts - What's new?

- Pluggable Request Processor

Struts now sports a new pluggable Request Processor based on the Chain of Command pattern, which isn't a pattern so much as it is a combination of patterns (Chain of Responsibility and Command patterns).

A Look at Commons Chain, Part 1

<http://www.onjava.com/pub/a/onjava/2005/03/02/commonchains.html>

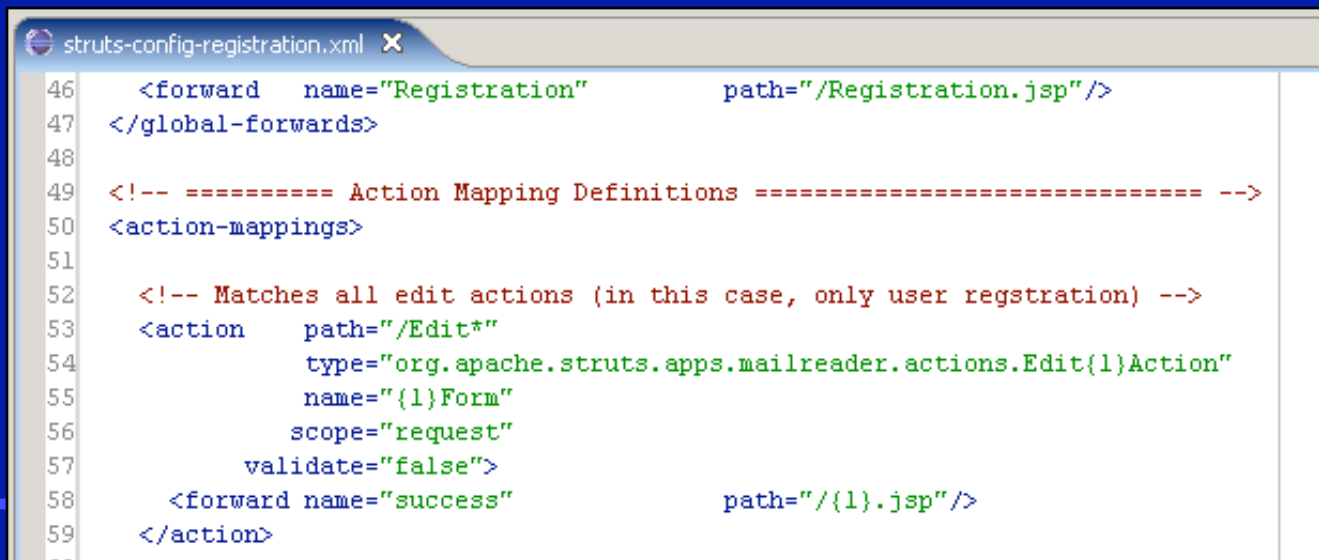
A Look at Commons Chain, Part 2

<http://www.onjava.com/pub/a/onjava/2005/03/02/commonchains.html>

Apache Struts - What's new?

- Wildcard Mappings

Reduce the number of action mappings in your struts-config.xml file by allowing pattern matching for your configuration.



```
struts-config-registration.xml x
46 <forward name="Registration" path="/Registration.jsp"/>
47 </global-forwards>
48
49 <!-- ===== Action Mapping Definitions ===== -->
50 <action-mappings>
51
52 <!-- Matches all edit actions (in this case, only user registration) -->
53 <action path="/Edit*"
54         type="org.apache.struts.apps.mailreader.actions.Edit{1}Action"
55         name="{1}Form"
56         scope="request"
57         validate="false">
58     <forward name="success" path="{1}.jsp"/>
59 </action>
```

Apache Struts - What's new?

- Validator

- ValidWhen

- If you've ever struggled with RequiredIf, you'll love the freedom ValidWhen brings to the table.

- Core

- Saving Messages in the Session

- Re-directing ActionForward

- Download Action

- Dispatch Helper

- *Lazy Validator Form*

- Config Files in jars

Apache Struts - What's new?

- Automatic Highlighting of Errors
- Readonly / Disabled Forms
- HTML Tag Refactoring
 - prepareName()
 - prepareOtherAttributes()
 - prepareValue()
 - prepareAttribute()
- Bundle Attribute (*now added to the following*)
 - ButtonTag, CancelTag, CheckboxTag, FileTag, FrameTag, HiddenTag, LinkTag, MultiboxTag, PasswordTag, RadioTag, ResetTag, SelectTag, SubmitTag, TextTag, TextareaTag
- ErrorTag – (*now has header, footer, prefix, suffix attributes*)
- Resource Bundle Support (*Validator*)

Apache Struts - Future

- What does the future hold for the Apache Struts project?
 - Struts 1.3.x (pending)
 - Enhance all configs to extend one configuration element from another, as is done with Tiles Definitions (in progress)
 - Leverage *Commons Resources* for messages bundles (pending release 1.0)
 - Struts 1.4.x
 - Support for an action context
 - Support for Porlets

Apache Struts - Future

- What does the future hold for the Apache Struts project?
 - Beyond 1.4.x
 - Consider Cocoon Plugin
 - Consider Spring Plugin
 - Consider adopting several popular extensions, including:
 - Struts Layout
 - SSL Ext
 - Stxx (XLST)
 - StrutsTestCase

Apache Struts – Best Practices

- Getting started
 - Build Options
 - Ant
 - Maven
 - Others?

Apache Struts – Best Practices

- IDEs
 - Eclipse
 - NetBeans
 - MyEclipse
 - Nitrox
 - IBM WebSphere Studio
 - M7 NitroX
 - Borland JBuilder
 - Sun Java Studio Creator
 - WebLogic Workshop
 - IntelliJ IDEA
 - JCreator
 - Oracle JDeveloper

Apache Struts – Best Practices

- Configuration
 - Multi-config
 - Break up your configuration into manageable pieces by moving related functionality into their own struts-config-foo.xml and validator-foo.xml.
 - Use Declarative Exception Handling
 - Let the framework help you deal with exceptions.
 - All requests should go through the controller.
- Application Messaging
 - Share your bundle (with JSTL)
 - Consider your options (DBMessageResources)

Apache Struts – Best Practices

- JSPs
 - Avoid JSP scriptlets `<%=“AT ALL COSTS!!!”%>`
 - Validation (Struts Validator)
 - Avoid adding static code in every call. (optional client side validation)
 - Create a simple and common way to deal with validation errors (related to application messaging)
 - Stylesheet declaration
 - Which would you rather use?
 - a)

```
<link rel="stylesheet" type="text/css" href="/myapp/stylesheets/global.css">
```
 - b)

```
<bean:define id="style">
  <html:rewrite page="/stylesheets/global.css" />
</bean:define>
<link rel="stylesheet" type="text/css" href="<%=style%>">
```
 - c)

```
<style type="text/css" media="screen">
  @import url(<html:rewrite page="/stylesheets/global.css" />);
</style>
```

Apache Struts – Best Practices

- JSPs
 - Use descriptive naming for your JSP pages. Pick a convention and **STICK TO IT!!** (*especially if you are using wild card mapping*)
 - /create-customer.jsp
 - /edit-customer.jsp
 - ...or...
 - /customer/create.jsp
 - /customer/edit.jsp

Apache Struts – Best Practices

- JSPs
 - Reuse markup where you can, but do not reuse mockup pages
 - Static includes
 - Dynamic includes
 - JSTL
 - Tiles
 - Protect your JSPs
 - Use a Security Constraint (this is the optimal solution)
 - Place files under WEB-INF (if your container supports it)
 - Custom taglib (not recommended, even though the struts-example app demonstrates how to do it)

Apache Struts – Best Practices

● JSPs

- Always use Struts tags to reference included content. Do not hard code your application mappings.

Example 1: Path mapping (root context)

(web.xml)

```
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>/struts/app/*</url-pattern>
</servlet-mapping>
```

(nav.jsp)

```
<html:link action="/home">Home</html:link>
```

(e.g. /struts/app/HomeAction)

Apache Struts – Best Practices

● JSPs

- Always use Struts tags to reference included content. Do not hard code your application mappings.

Example 2: Extension mapping (root context)

(web.xml)

```
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

(nav.jsp)

```
<html:link action="/home">Home</html:link>
```

(e.g. /HomeAction.do)

Apache Struts – Best Practices

- Actions
 - Don't create Actions that just to return success, use a SuccessAction
 - Provide a single point of entry in your controller classes (AbstractBaseAction.java)
 - Keep business logic out of your view layer (Actions, JSP pages, etc).
 - Use tokens to deal with duplicate requests

Apache Struts – Best Practices

- ActionForms

- Prefer request scope over session
- Use Dyna*Form (various flavors)
- If you can't use Dyna*Form, use FormDef

If you can't (or won't) use either of the above, then the rest of the page is for you

- Use String fields only
- Should be used as a buffer between layers
- Accessors should be dumb

If you can't (or won't) use a String only approach, then the rest of the page is for you

- Use a custom converter to handle type conversion that is not already handled by Struts

Apache Struts – Best Practices

- ActionForwards
 - Use Global forwards as a common way to deal with redundant local forwards
 - Avoid forwarding to Actions

Apache Struts – Best Practices

- Struts Modules
 - Don't use them. Any questions?

Apache Struts – Best Practices

- Session Management
 - Application Filter (All)
 - Custom Request Processor (All)
 - Abstract Base Action (All)
 - Custom Chain (1.2.7 +)
 - Custom Taglib (not recommended)

Apache Struts – Best Practices

- Team Development
 - What's in your tool kit?
 - Get efficient with your tools.
 - Know (and use) an established set of standards.

Apache Struts - Discussion

- Techniques (for dealing with ...)
 - Data Access
 - HTTP Compression (gzip, etc)
 - Asynchronous and Scheduling Jobs
 - File upload
 - Character encoding
 - <http://www.joelonsoftware.com/articles/Unicode.html>
 - Debugging and/or Profiling
 - Progress bar or “Please wait” screen
 - Autocomplete

Apache Struts - Discussion

- Components
 - Calendar
 - Paging
 - Reporting (Charts and Graphs)
 - Menus

Apache Struts - Discussion

- Testing
 - Web Tier
 - Business Objects
 - Data Layer
- Test Coverage
 - JCoverage
 - Clover

Apache Struts - Discussion

- Design Patterns (GoF and beyond)
 - MVC
 - Command Pattern
 - Chain of Responsibility
 - Front Controller
 - View Dispatcher
 - Service To Worker
 - Composite View

Apache Struts - Discussion

- Nice to haves
 - Allow `${bundle.msg}` in config files
 - `struts-config.xml`
 - `tiles-defs.xml`
 - `validator.xml`
 - Allow import and use of static finals (`com.company.Constants.*`) in config files
 - (same as above)

Apache Struts – Q & A

Q & A
Fire away !!!