

Getting Started with the Stripes Framework

Emerging Technologies for the Enterprise 2009

Michael P. Redlich

March 27, 2009



▲ Degree

- B.S. in Computer Science
- Rutgers University (go **Scarlet Knights!**)

▲ “Petrochemical Research Organization”

- Senior Research Technician (1988-1998, 2004-present)
- Systems Analyst (1998-2002)

▲ Ai-Logix, Inc. (Now AudioCodes)

- Technical Support Engineer (2003-2004)

▲ Amateur Computer Group of New Jersey (ACGNJ)

- Java Users Group Leader (2001-present)
- President (2007-present)
- Secretary (2006)





▲ Publications

- Java Boutique (<http://www.javaboutique.com/>)
 - ❖ Co-authored with Barry Burd
 - ❖ Design Patterns
- <http://publications.redlich.net/>

▲ Presentations

- Emerging Technologies for the Enterprise 2008
- Trenton Computer Festival (TCF) since 1998
- TCF IT Professional Conference since 2006
- Princeton Java Users Group
- Capital District Java Developers Network
- New York Software Industry Association (NYSIA) Java Users Group



Trenton Computer Festival (TCF) 2009

- April 24-26, 2009
- IT Professional Conference (April 24, 2009)
- General Show (April 25-26, 2009)
- The College of New Jersey
- Ewing, New Jersey
- <http://www.tcf-nj.org/>

///Stripes

Objectives



▲ What is Stripes?

- “What! Yet another web framework?!?”

▲ Stripes vs. Struts

▲ How to get started

▲ Overview of some basic components

▲ Stripes Lifecycle

▲ Live demo

- Source code, source code, source code (*woo-hoo!*)
- Build a small web application from scratch...



What is Stripes?



▲ Open source, action-based Java web framework

▲ Tim Fennell

- “Father” of Stripes

- Lead Developer

- Anti-Struts!

▲ Designed around the principles that web development should be:

- Simple

- Productive



- ▲ Struts
- ▲ Tapestry
- ▲ Cocoon
- ▲ MyFaces
- ▲ Wicket
- ▲ WebWork
- ▲ Spring
- ▲ Google Web Toolkit
- ▲ Turbine
- ▲ Makumba
- ▲ Helma
- ▲ Restlet
- ▲ Echo
- ▲ JPublish
- ▲ Trimpath Junction
- ▲ Calyxo
- ▲ RIFE
- ▲ DWR
- ▲ JOSSO (Java Open Single Sign-On)
- ▲ OpenXava
- ▲ Click
- ▲ ZK
- ▲ RSF



▲ Strecks

▲ flexive

▲ Maverick

▲ Anvil

▲ Jaffa

▲ Vraptor

▲ Millstone

▲ wingS

▲ Aranea

▲ Hamlets

▲ ThinWire

▲ Pustefix

▲ jWic

▲ Mentawai

▲ SpringWeb

▲ Macaw

▲ Ztemplates

▲ Jucas

▲ Chrysalis

▲ Caramba

▲ jZeno

▲ Aurora

▲ SOFIA

▲ Verge

*Goals of Stripes**



- ▲ Make developing web applications in Java easy
- ▲ Provide simple yet powerful solutions to common problems
- ▲ Make the Stripes ramp up time for a new developer less than 30 minutes
- ▲ Make it really easy to extend Stripes, without making you configure every last thing

* Source: Stripes web site, <http://www.stripesframework.org/>.



- ▲ Zero external configuration per page/action
 - Action Beans are auto-discovered and configured using annotations
- ▲ Powerful binding engine that will build complex object webs out of the request parameters
- ▲ Easy to use (and localized) validation and type conversion system
- ▲ Localization system that works even when you use direct JSP->JSP links
- ▲ Ability to re-use Action Beans as view helpers

* Source: Stripes web site, <http://www.stripesframework.org/>.

Key Features (2)*



- ▲ Ridiculously easy to use indexed property support
- ▲ Built-in support for multiple events per form
- ▲ Transparent file upload capabilities
- ▲ Support for incremental development
 - e.g., you can build and test your JSP before even thinking about your Action Bean
- ▲ And a lot of built in flexibility that you only have to be aware of when you need to use it

* Source: Stripes web site, <http://www.stripesframework.org/>.

*Stripes Aims to...**



- ▲ ...provide an experience similar to owning:
 - Apple hardware
 - Sony TVs
 - Luxury German cars (without the price premium)
- ▲ Things just work...
- ▲ Things just feel right...
- ▲ And you sometimes get that “Oh, it does that too? Awesome!” feeling...

* Source: Stripes web site, <http://www.stripesframework.org/>.



▲ ActionBean

▲ JSP

▲ web.xml

▲ Specific Annotations

@UrlBinding

@Validate

▲ Form and action information in one location

▲ Action

▲ Form

▲ JSP

▲ web.xml

▲ struts-config.xml

form-bean stanza

action stanza

Forward stanzas

///Stripes

Struts

Stripes vs. Struts2 (1)



	Stripes	Struts2
Version	1.5	2.0.12
Configuration	web.xml	web.xml, struts.xml, optionally struts.properties and others
Main workhorse	Classes that implement <code>ActionBean</code>	Classes that have an <code>execute()</code> method, optionally implement <code>Action</code> , or extend <code>ActionSupport</code>
Response mechanism	Instance of <code>Resolution</code>	String identifier that maps to a result in <code>struts.xml</code> or in an annotation
View technology	JSP or FreeMarker	JSP, FreeMarker, or Velocity
Layout mechanism	Built-in, with three layout tags. For people who like Tiles or SiteMesh, they can be used as well.	Tiles or SiteMesh
Binding mechanism	Built-in	OGNL
Validation	@Validate and @ValidateNestedProperties	Configure in an XML file, or use annotations

Stripes vs. Struts2 (2)



	Stripes	Struts2
Validation short-circuiting	Built-in, configurable with <code>when=ValidationState.ALWAYS</code> and <code>Validation.InvokeValidateWhenErrorsExist</code>	Set <code>short-circuit="true"</code> on <code><field-validator></code>
Custom validation	Annotate your method with <code>@ValidationMethod</code>	Extend either <code>ValidatorSupport</code> or <code>FieldValidatorSupport</code> , and configure in <code>validators.xml</code>
Model-to-view data transfer	<code>#{actionBean}</code> attribute	<code>ValueStack</code>
Type conversion	Implementations of <code>TypeConverter<T></code> (generified)	Implementations of <code>ognl.TypeConverter</code> , typically extensions of <code>StrutsTypeConverter</code> (not generified)
Formatting	Implementations of <code>Formatter<T></code> (generified)	Implementations of <code>ognl.TypeConverter</code> , typically extensions of <code>StrutsTypeConverter</code> (not generified)
Custom module configuration	Automatically loaded with <code>Extension.Packages</code> init-parameter	Configuration in <code>struts.xml</code>
Interceptors	Implementations of <code>Interceptor</code> , or methods annotated with <code>@Before/@After</code>	Implementations of <code>Interceptor</code> , with configuration in <code>struts.xml</code>
Localization	Resource bundle(s) for errors and field names, and JSTL	Resource bundle search mechanism



What Do I Need to Get Started?



▲ Stripes 1.5.1

□ <http://www.stripesframework.org/>

▲ Java classes (as needed)

▲ JSP pages (as needed)

▲ web.xml file

▲ StripesResources.properties file

▲ Servlet container



- ▲ ActionBean
- ▲ ActionBeanContext
- ▲ Resolution
- ▲ StripesFilter
- ▲ DispatcherServlet
- ▲ @Validate



- ⚠ Interface for all classes that respond to user events through an `ActionBeanContext` object
- ⚠ Allows the Stripes Dispatcher to inject an `ActionBeanContext` object into the action being serviced
- ⚠ All Stripes actions need to implement this interface



▲ Encapsulates information about the current action

- State information
- Informational messages
- Error messages

▲ Provides access to the Servlet API

- `HttpServletRequest`
- `HttpServletResponse`
- `ServletContext`
- `ValidationErrors`



- ⚠ An interface designed to be returned by handler methods in Action Beans
- ⚠ Responsible for executing the next step after an Action Bean has handled the user's request
- ⚠ Called by the Stripes Dispatcher to invoke a Resolution
- ⚠ Should use the request and response provided to direct the user to an appropriate view

```
void execute(HttpServletRequest request, HttpServletResponse response);
```



▲ ForwardResolution

- Uses a server-side forward directive to forward the user to another path in the same application

▲ RedirectResoution

- Uses a client-side redirect the user to another path in the same web application or another web application on the web

▲ StreamingResoution

- Used to stream content back to the user

▲ JavaScriptResolution

- Used to convert a Java object into JavaScript objects and arrays and stream them back to the client



- ▲ Configured in `web.xml` file
 - No additional external configuration files necessary!
- ▲ All requests are filtered through `StripesFilter`
- ▲ Ensures that all requests coming to a Stripes application are handled consistently
- ▲ All actions are auto-discovered at deployment time
- ▲ Performs initialization of `<init-param>` elements
- ▲ `ActionResolver.Packages` is the only *required* `<init-param>` element
 - Specify one or more package roots
 - Subpackages automatically included



```
<filter>
  <display-name>Stripes Filter</display-name>
  <filter-name>StripesFilter</filter-name>
  <filter-class>net...StripesFilter</filter-class>
  <init-param>
    <param-name>ActionResolver.Packages</param-name>
    <param-value>org.emergingtech.hello</param-value>
  </init-param>
</filter>
```



- ▲ Configured in `web.xml` file
 - No additional external configuration files necessary!
- ▲ Controls how requests to the Stripes framework are processed
- ▲ Resolves a URL to a Stripes `ActionBean` class

```
org/emergingtech/calculator/Calculator.action  
org.emergingtech.calculator.CalculatorActionBean
```




```
<servlet>
  <servlet-name>StripesDispatcher</servlet-name>
  <servlet-class>net...DispatcherServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>StripesDispatcher</servlet-name>
  <url-pattern>*.action</url-pattern>
</servlet-mapping>
```

///Stripes

Validation



- ▲ Specifies validation for form fields
- ▲ Custom validations

```
@ValidateNestedProperties ({
    @Validate (field = "firstName", required = true, on =
{"hello"}),
    @Validate (field = "age", required = true, minvalue = 13)
})
private Person person;

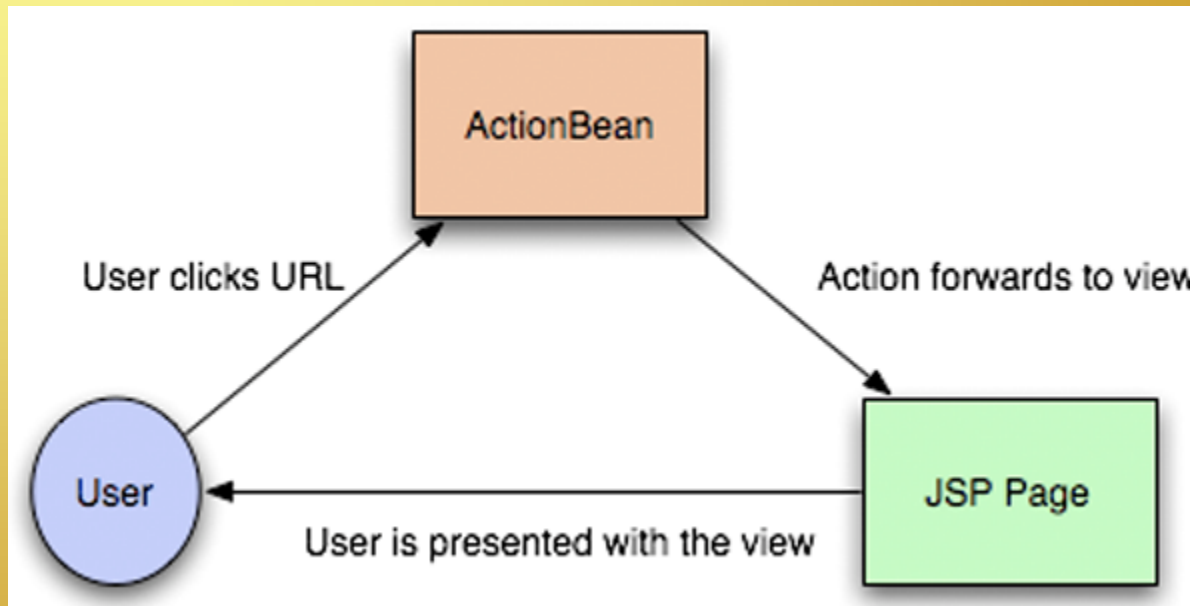
@Validate (required=true)
Private double numberOne;
```





- ▲ Represented by a method in a concrete `ActionBean` class
- ▲ Must have the following method signature:

```
public Resolution eventName();
```





- ▲ Resolve an `ActionBean` based on the URL of the request and set the `ActionBeanContext` on it
- ▲ Resolve the handler method that will handle the event received in the request
- ▲ Bind properties from the `HttpServletRequest` into the `ActionBean`, running validation as necessary
- ▲ Invoke any custom validation methods
- ▲ Invoke the appropriate handler method on the `ActionBean`
- ▲ Execute a concrete `Resolution` returned by the `ActionBean`



- ▲ “Hides” the current configuration through a concrete `Configuration` object
 - The configuration can be retrieved by calling `StripesFilter.getConfiguration()`
- ▲ Resolves a `Locale` object that should be used for the current request
- ▲ Wraps the `HttpServletRequest` with a `StripesRequestWrapper`
 - Detects when the request is a multipart/form-data request
 - Correctly parses such requests to provide access to the request parameters and uploaded files



▲ Flow of control to either:

- Directly to a JSP page
- Stripes Dispatcher
 - ❖ If the request is for an `ActionBean` event



🐼 The Stripes Dispatcher:

- ❑ Manufactures an `ActionBeanContext`
 - ❖ From the `ActionBeanContextFactory`
- ❑ Resolves the appropriate `ActionBean` instance
 - ❖ From the configured `ActionResolver`
- ❑ Matches the URL path of the request to the URL binding of an `ActionBean` class
- ❑ Creates an instance of the `ActionBean`
- ❑ Inserts the bean into the relevant scope (request or session)
- ❑ Returns the `ActionBean`



▲ The Stripes Dispatcher:

- ❑ Uses `ActionResolver` to determine the name of the event submitted
- ❑ If there was no identifiable event name, `ActionResolver` uses the method annotated with `@DefaultHandler`

▲ The event name is then set on the `ActionBeanContext`



- ▲ Performs required field validation on all required fields
- ▲ Perform pre-conversion validations
 - Such as minimum and maximum length attributes
- ▲ For each field supplied in the request that had a non-empty-string value
 - Convert the field using the type conversion system
 - Bind the converted values on to the `ActionBean`
- ▲ Run post conversion validations including min/max numeric value and expression checks
- ▲ Return a `ValidationErrors` containing any errors that arose during validation and binding



- ▲ The Stripes Dispatcher invokes the handler method
- ▲ If the `ActionBean` throws an Exception:
 - It is propagated by Stripes Dispatcher, either:
 - ❖ Directly if it is a `Servlet` or `Runtime` exception, or
 - ❖ By wrapping it in a `StripesServletException`
- ▲ The `ActionBean` may execute arbitrary code, including handling the response directly
- ▲ Handler methods may return any `Object`
 - However, the return is ignored unless it is an instance of `Resolution`
 - If the `ActionBean` returns a non-null `Resolution`, the Stripes Dispatcher will call its `execute ()` method to complete the request



▲ Custom Configurations

▲ Extensions

▲ Can use Stripes with:

Spring

AJAX



🔥 ...the live demo!

Are You Ready?

///Stripes

Summary



- ▲ Stripes was designed on the principles that web development should be easy and productive
- ▲ You can get started with Stripes in a minimal amount of time
- ▲ You can download Stripes from:
 - <http://www.stripesframework.org/>
- ▲ You can get this presentation and source code from:
 - <http://www.chariotsolutions.com/>
 - <http://presentations.redlich.net/>
- ▲ If you happen to be in the Scotch Plains, NJ area, please visit us at an ACGNJ Java User Group meeting!

▲ Stripes Framework

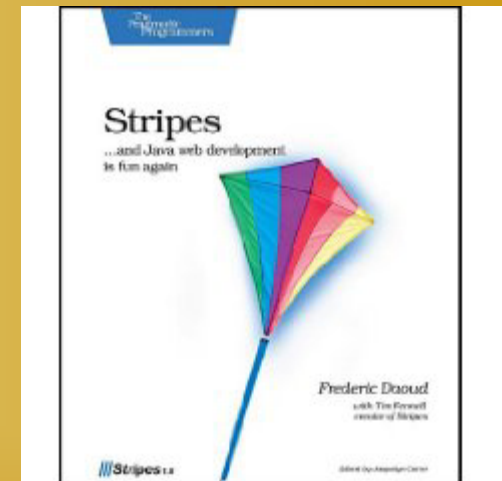
- <http://www.stripesframework.org/>

▲ Stripes ...and Java web development is fun again

- Book (Pragmatic Programmers)
- Frederic Daoud (with Tim Fennell)
- <http://www.amazon.com/>

▲ Java Web Development with Stripes

- Article (ONJava)
- Mark Eagle
- <http://www.onjava.com/lpt/a/6901/>





▲ Using Stripes as a Web MVC Framework without that thing called XML

- Blog Article (Gridshore)
- Jetro Coenradie
- <http://www.gridshore.nl/2008/12/13/using-stripes-as-a-webmvc-framework-without-that-thing-called-xml/>



▲ ACGNJ Java Users Group

facilitated by Mike Redlich

<http://www.javasig.org/>

▲ Princeton Java Users Group

facilitated by Yakov Fain

<http://www.myflex.org/princetonjug/>

▲ NYJavaSIG

facilitated by Frank Greco

<http://www.javasig.com/>

▲ Capital District Java Developers Network

facilitated by Anthony DeBonis

<http://www.cdjdn.com/>





▲ Chariot Solutions

☐ <http://www.chariotsolutions.com/>

