

Struts Next

•

What should be improved / changed

Development plan of Struts 3

- M1 (aka Struts 2.5)
- M2 (aka Struts 2.6)
- M3 (beta 1 Struts 3)
- M4 (beta 2 Struts 3)

Cleanup website

Undiscussed ideas

What should be improved / changed

Issues connected with the new version

Build of the latest version

- Use Git instead Svn ✓
- Switch to Java 7 ✓
- Update "internal" injection to use current Guice version
 - Consider usage of <http://onami.apache.org/>
- Refactorings for further extensibility based on questions on the list, on CodeRanch, and Stack Overflow
 - JR plugin refactoring. Allow compiled reports as classpath resources.
- Refactor conversion of primitives (and wrappers) in XWork
- Switch to [OGNL 4.0](#).
- Consider renaming packages to org.apache.struts3
- Remove deprecated Plugins ✓
 - Codebehind
 - Dojo
 - Struts1 ?
- Normalize property name camel-casing (foo.bar.BazPlugh to foo.bar.bazPlugh)
- Check possible usage of Tiles Auto Tag (<http://tiles.apache.org/tiles-autotag/>) instead of Struts Annotations
- Convention plugin should be the "normal" way of using Struts
- Plugin to map websocket requests to Struts actions
- API to customize property finding via arbitrary request parameter names ("a" -> getFoo())
 - Happily breaks apart the 1:1 relationship between property paths and HTTP parameter names
 - Will be useful for obfuscation (doesn't expose java model)
 - Can be used to shorten request payload (smaller names)
 - Can be taken advantage by HDIV
- Themes cleanup
 - Support HTML 5 features and Single Page Application development paradigm
 - Must output valid markup
 - Should be compatible with bootstrap, bootstrap validator, and other major UI platforms

Development plan of Struts 3

M1 (aka Struts 2.5)

- ✓ Request Git repo from INFRA
- ✓ Import project
- ✓ Remove deprecated plugins - moved to struts-archive.git repo
 - ✓ we need place for deprecated stuff (to move the source code to) - new project "struts-archive"
 - ✓ drop support for Struts 1 (remove plugin)
 - ✓ dojo plugin
 - ✓ codebehind plugin
 - ✓ jsf plugin
- ✓ Remove deprecated APIs
 - 1. if it cannot be removed, mark to be removed during work on 3.0
- ✓ Switch to Java 7
- ✓ Merge XWork code into Struts Core (without renaming packages)
- ✓ Prepare the first beta release

—

- ✓ Mark current logging layer as `@deprecated` and use Log4j2 as default one
- ✓ Add plugin to support bean validation
 - ✓ Implement plugin
 - ✓ Add showcase example
 - ✓ Add documentation to wiki
- ✓ Upgrade spring plugins spring version to 4.x
- ✓ Move struts2 blank, jboss-blank and mailreader app to struts2-examples
- ✓ Switch to strict DMI by default - allow only defined methods to be accessible via DMI - user can always change that list
- ✓ prepare second beta
- ☐ drop support for "top" object - this can be painful, users are using so migration path is needed
- ✓ Cleanup website
 - ✓ new user guide totally developed out of Confluence - people will be able to contribute to it via GitHub
 - ✓ Confluence (wiki) used only to explain more detailed given aspect of the framework (ie.: how to implement your own converter in details)
 - ✓ keep just one version of the doc (no more release/2.3.x, release/2.2.x and so on - Google indexes everything and people are confused)
- ✓ prepare the last beta

M2 (aka Struts 2.6)

1. Servlets 2.5, JSP 2.1, Java 7
2. `struts-api` module
3. Drop support for internal logging layer and use Log4j2 only

M3 (beta 1 Struts 3)

1. Servlets 3.0, JSP 2.2, Java 8
2. Finish merging XWork code base (rename packages)
3. Rename Struts 2 packages to `org.apache.struts3`
4. Switch to OGNL 4
 - a. we must release it first
 - b. also we need better API interfaces to allow integrate Struts with other EL engines, ie.: MVEL - right now the whole conversion layer is hardly tied with OGNL
5. Switch to Servlet Specification 3.0
 - a. drop embedded JSP plugin

M4 (beta 2 Struts 3)

1. move plugins to separated project (need to be discussed)
 - a. disconnect plugins release cycle from core release

Cleanup website

There should be just one project website - Apache Struts. Right now there is the main website - Apache Struts project, subsite Apache Struts 1 and another subsite Apache Struts 2. After announcing EOL of Struts 1, we basically supporting just one version or rather branch - Apache Struts 2 which should be at some point converted just into Apache Struts (version 2.5, version 3.0, version 3.1, etc.).

With that in mind we should have just one website to manage the project.

Undiscussed ideas

- JSR 371 support (<https://jcp.org/en/jsr/detail?id=371>)
- Better support for modern webframeworks like AngularJS (@grobmeier)
- Make use of slf4j and probably log4j2 (@grobmeier) - support for slf4j is already there, thanks to Johannes, check [Logging](#)
- Provide alternatives to java property file translations (@grobmeier)
- Replace css and xhtml theme with an new html5 theme. (@jogep)
- Extract XML configuration to dedicated plugin - XWork will provide only configuration via code, then user can decide which configuration to use - code, XML, convention, DB?
- Should plugins move out of the Struts release cycle so they can independently released? (@pbenedict)
- Turn all request and response processing into interceptors (perhaps introduce a "system stack" to complement a "user stack" to simplify configuration) (@pbenedict)