Struts Dialogs

1. Version 2.0 is available

Version 2.0 of Struts Dialogs retains the same functionality, simpler configuration and less classes. In fact, version 2.0 does not contain any of the version 1.x classes, and depends solely on one EventDispatcher class.

Version 1.x documentation can be found here.

2. Overview

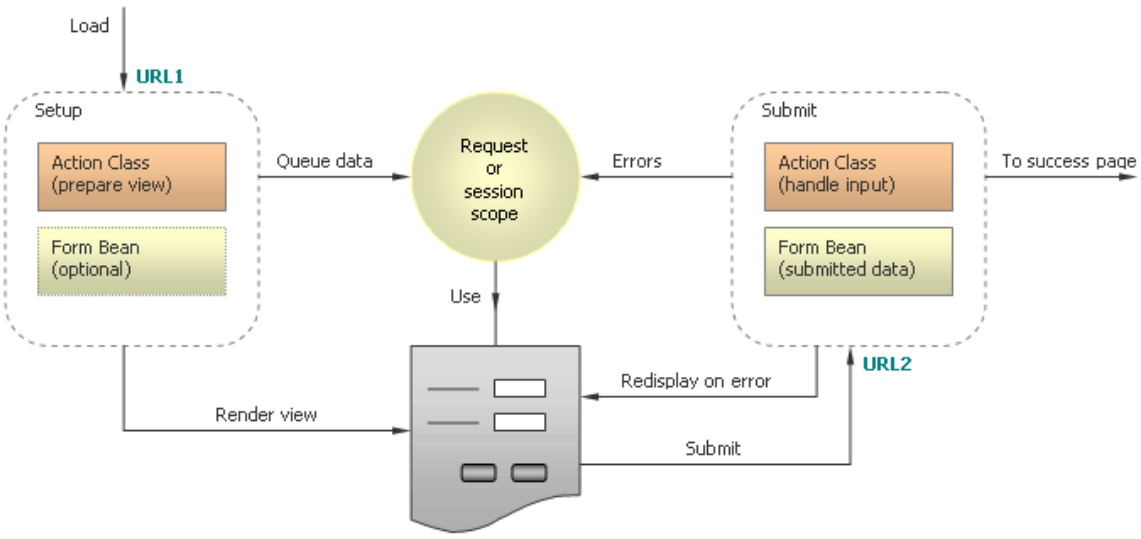
Struts Dialogs is both a library and a development approach for Struts 1.2.x framework that improves development process and makes applications more robust and user-friendly. Struts Dialogs demonstrates how event handling, basic state management, and provides simplified control flow.

- **Simplified control flow** - cleaner separation of concerns between actions, action forms and JSP pages.
- **Event handling** - uniform processing of command links and form submission events.
- **State management** - using session-scoped form bean as stateful input/output buffer.
- **Easier configuration** - improved request/response cycle is controlled with less XML markup and fewer Java classes.
- **Web Wizards** - controlled flow of web pages for a given web resource, similar to traditional desktop wizard dialogs.

3. Struts: traditional request/response cycle

Class `ActionForm` was initially designed as convenience object for input data only. Struts guidelines recommend using ActionForm in request scope. This precludes from storing state information in ActionForm. It is up to developer to decide where to queue output data to, and where to store information between requests.

Struts users came up with idea of *setup action* (output action, pre-action) and *submit action* (input action, post-action) dispatchers. This pattern is page-centric, actions and JSP pages are interlaced, business data is located outside of an action or a form bean:



Pre- and Post- actions

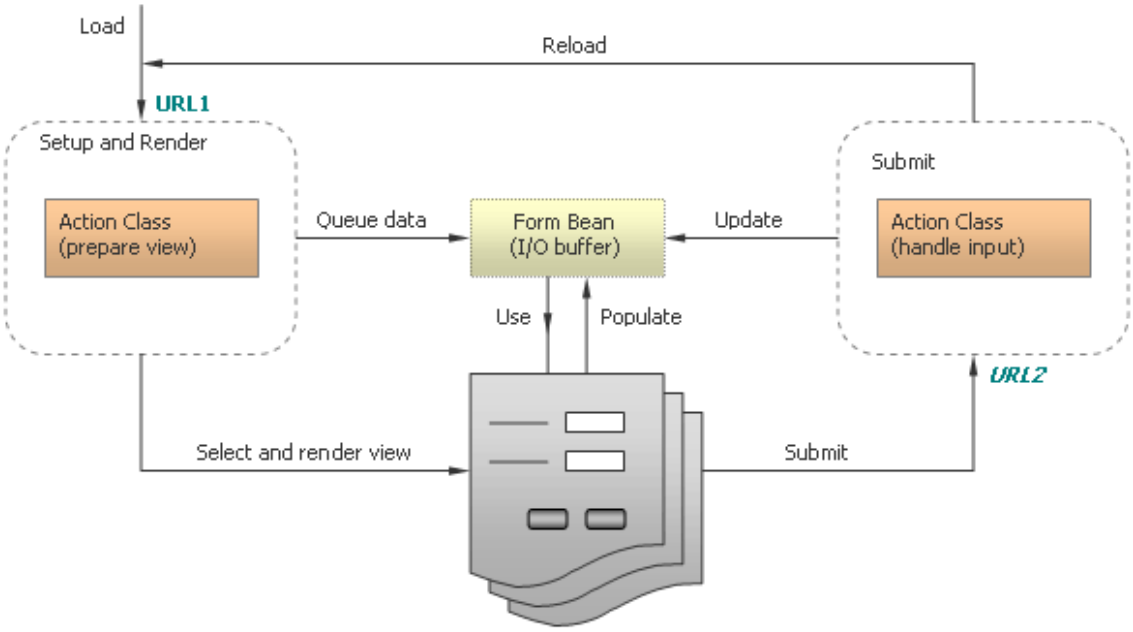
This approach is not perfect:

- Focused on a JSP page, not on a web resource in general.
- Every page of a web resource is likely to have its own pair of setup and submit actions. The picture above represents one JSP page and two actions associated with it. More pages, more actions, and things can quickly get out of control.
- One web resource is defined with several action mappings in the `struts-config.xml` file as well as with several Java classes.
- Output data is scattered in an uncontrolled manner throughout request and session scope.
- In case of error a page is redisplayed by a submit action, not by setup action; that opens a whole can of worms:
  - If input data is invalid and autovalidation is turned on, a submit action class is never get called and cannot affect the workflow.
  - One page is represented with two different URLs in the browser.
  - An attempt to refresh a page after it has been redisplayed causes double submit.
- Success page often corresponds to a logically different web resource, this leads to a spaghetti code both in Java code as well as in struts-config.xml file.

4. Struts Dialogs: clean two-phase approach

Struts Dialogs makes development simpler by using only two, or even just one action class per web resource. A web resource can be served with one setup/render action and one submit action. Submit action dispatches client event to a respective handler method, while render

action selects a JSP page appropriate to current resource state, and fill out a form bean with output data.



Dialog action

See Data Entry Form wiki page for more discussion on Struts MVC.

5. State management

Struts Dialogs does not introduce new classes to manage application state. Instead, it uses existing ActionForm class as a stateful input/output object. In JSF terms, ActionForm acts as a backing bean for JSP page(s) of a web resource. There is nothing groundbreaking in using session scope for ActionForm, or in storing output data in it. Online poll shows that about 60% of respondents use ActionForm for queueing output data.

With changing ActionForm scope to session, it is possible to initialize form bean only once, and to reuse data between requests. This is convenient for form resubmissions, for page reloading or for navigating back to previous resource. Having all resource data in a single ActionForm simplifies JSP page. Struts automatically populates ActionForm with submitted data on input phase, the same data can be used for presentation on render phase without additional efforts.

Session scope justifies the usage of nested properties within ActionForm. It is easy and

convenient to use business objects or DTOs as nested properties instead of copying their data to ActionForm and from ActionForm.

6. Using action dispatcher

Struts Dialogs 2.0 does not use custom versions of DispatchAction class anymore. Instead, it employs ParameterListActionDispatcher class renamed to EventDispatcher. This ensures that when ParameterListActionDispatcher is included in Struts 1.3.x core, you won't experience problems with same class names. The dispatcher class allows to use any Action class to handle user events.

7. Samples

7.1. Simple dispatch

`DispatcherSampleAction` in `net.jspcontrols.dialogs.samples.dispatch` package shows how to dispatch user events to an arbitrary action class. An event can be sent with GET or POST, with submit button or with link. Struts Cancel button is supported too.

7.2. Web component with two views

`LoginxxxAction` class in `net.jspcontrols.dialogs.samples.login` package shows how to create a multi-state multi-view web component.

7.3. Robust page flow

`SignupxxxAction` classes in `net.jspcontrols.dialogs.samples.signupwizard` package show how to create a *web wizard* similar to traditional desktop wizard dialog. A wizard has predefined sequence of states, and is rendered with HTML forms, containing Back, Forward, Cancel and Done pushbuttons. This sample shows how certain states can be skipped depending on wizard state.

Notice, that web wizard is about component state and corresponding view, not about merely flipping pages.

7.4. CrUD operations

`EmployeexxxAction` classes in `net.jspcontrols.dialogs.samples.crud` package show how to perform standard create, read, update and delete operations along with

displaying the list of items to choose from.

8. Live Demos

Each action class from Struts Dialogs library is illustrated with sample code and live demos (version 1.x).

9. Download

Download Struts Dialogs.