Struts Features

- A Back Mechanism -

Manfred Wolff, wolff@manfred-wolff.de

Introduction

There are several situation, that you can't realise with struts:

- You want to know where you come from.
- You want to go back and forward in your application like a "wizard".
- You want to make a language switching and stay where you are.
- ...

This is because struts don't have a protocol of these steps. The action-mapping only extracts what the next step is. This problem is as old as struts is and it has been discussed in all struts user lists. In 2003 I wrote an article and offered a little source code to solve this problem (see `http://www.manfred-wolff.de/struts/articles-old/HowTo-Back.html`). After that I got 70 visits per month on my homepage - only from people which were interested in this feature. So in 2006 I did a refactoring and made a sourceforge project of this solution. Now it is possible for anyone to download the code and an example application and everybody can offer patches for the solution.

The first solution tracks all `*.do` pathes and stores it into a ringbuffer. If the buffer is full, the first entry will be overwritten. So a first struts best practice is forced: Every request must pass the `RequestProcessor`. The second solution can be parameterize with a pattern, so that only a certain pathes will be saved, e.g. only pathes to actions, that are connected to a JSP-site.

Just have a look to figure 1. Every step to a jsp will be directed through a `Pre*Action` class. So a back has only to keep in mind of those actions. A

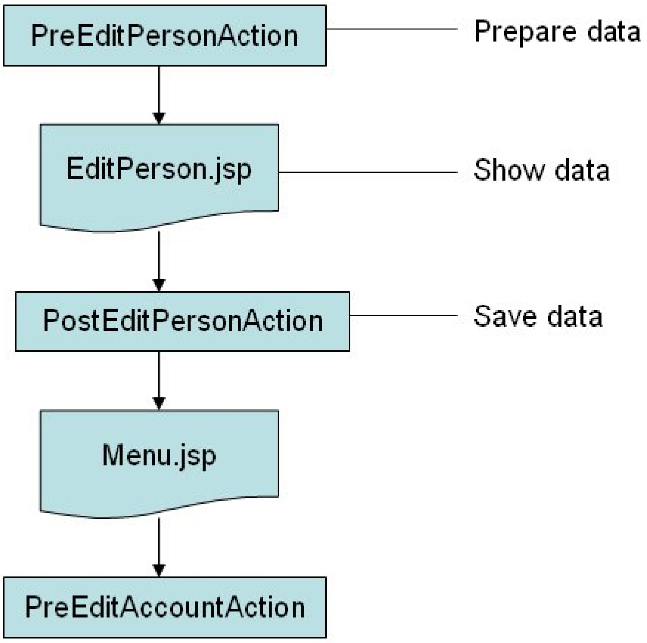

Figure 1: Concept of PreActions

back from the menu is directed to the `PreEditPersonAction`. If all links are created in such a way the back solution has only keep in mind for request-parameter and not for request attributes, because a back isn't really a back but a "forward into the past".

The new solution is really easy to configure. First you have to add two entries in the `struts-config.xml`

```
<action path="/back"
        type="org.mwolff.struts.back.BackRequestAction"/>
<action path="/forward"
        type="org.mwolff.struts.back.ForwardRequestAction"/>
```

The second thing is to add the `BackRequestProcessor` as a controller:

```
<controller
   processorClass="org.mwolff.struts.back.BackRequestProcessor"
  locale="true"
/>
```

After that it is no problem to go back and forward in struts applications:

```
  <html:link action="back.do">Go back to last action</html:link>
```

The BackRequestProcessor

coming soon

Resources

The source code is hosted at sourceforge

Coming up

- A demo application
- A complete documentation