

ActionMapping

Navigation Trail: [HomePage](#) / [StrutsProjectPages](#) / [StrutsDeveloperPages](#) / [JamesMitchell](#) / [MailDocumentation](#)

Cliff Rowley wrote:

```
> Greetings, very quick question. From the struts config DTD, I
> understand that action classes, like plugins, can be configured via
> bean style properties using the <set-property> in an <action> element:
>
> <quote>
> The "set-property" element specifies the method name and initial value
> of an additional JavaBean configuration property. When the object
> representing the surrounding element is instantiated, the accessor for
> the indicated property is called and passed the indicated value. The
> "set-property" element is especially useful when a custom subclass is
> used with <data-source>, <forward>, <action>, or <plug-in> elements.
> The subclass can be passed whatever other properties may be required
> to configure the object without changing how the struts-config is
> parsed.
>
> property          Name of the JavaBeans property whose setter method
>                   will be called.
>
> value             String representation of the value to which this
>                   property will be set, after suitable type conversion
> </quote>
>
> I've just had a stab at this but my property is not being set. The
> action configuration is as follows:
>
> <action path="/test" type="net.onsea.huts.actions.ListAction">
>   <set-property property="entityType" value="foo"/>
>   <forward name="success" path="/test.jsp"/>
> </action>
>
> and my action (which is a subclass of a base action, as in Action ->
> BaseAction -> ListAction) has a public getter/setter pair for its
> entityType property.
>
> Just wondered if this is a known issue/bug/user_error?
>
> Its late (early?) and I'm not sure my blurry eyes will be much use
> reading through the source code :|
>
> Cheers
>
```

Ted Husted:

I know this is confusing, but the <action> element does not instantiate an Action object. It creates an ActionMapping object (which acts like a decorator for the Action). This is the object that you need to extend. Then, at runtime, the controller will pass your Action object the mapping, and you can get the custom property from there. (Of course, you will need to typecase the mapping first.)

The <set-property> feature is provided by the Digester, so you won't find it in the Struts codebase. It's implemented in the Digester codebase.

-Ted.

--

Ted Husted,
Husted dot Com,

Fairport NY US
co-author, Java Web Development with Struts
Order it today:
<http://husted.com/struts/book.html>

Robert Taylor

Cliff, I use this functionality currently. All I did was extend `ActionMapping`, add my properties, and in the `struts-config.xml`, assigned the `type` attribute of the `action-mappings` element my extended `ActionMapping` class name. Struts will then use introspection to identify and assign the new properties their corresponding values.

```
<action-mappings type="com.companyname.BaseActionMapping">
```

It's an easy way to extend `ActionMapping` class.

robert