

Chaining Interceptor

Description

An interceptor that copies all the properties of every object in the value stack to the currently executing object, except for any object that implements [Unchainable](#). A collection of optional `includes` and `excludes` may be provided to control how and which parameters are copied. Only `includes` or `excludes` may be specified. Specifying both results in undefined behavior. See the javadocs for [ReflectionProvider#copy\(Object, Object, java.util.Map, java.util.Collection, java.util.Collection\)](#) for more information.



It is important to remember that this interceptor does nothing if there are no objects already on the stack. This means two things:

1. you can safely apply it to all your actions without any worry of adverse affects.
2. it is up to you to ensure an object exists in the stack prior to invoking this action. The most typical way this is done is through the use of the `chain` result type, which combines with this interceptor to make up the action chaining feature.

By default Errors, Field errors and Message aren't copied during chaining, to change the behaviour you can specify the below three constants in struts.properties or struts.xml:

- `struts.xwork.chaining.copyErrors` - set to true to copy Action Errors
- `struts.xwork.chaining.copyFieldErrors` - set to true to copy Field Errors
- `struts.xwork.chaining.copyMessages` - set to true to copy Action Messages

Example:

```
<constant name="struts.xwork.chaining.copyErrors" value="true" />
```

Parameters

- `excludes` (optional) - the list of parameter names to exclude from copying (all others will be included)
- `includes` (optional) - the list of parameter names to include when copying (all others will be excluded)

Extending the Interceptor

There are no known extension points to this interceptor.

Examples

Simple example how to chain two actions

```
<action name="someAction" class="com.examples.SomeAction">
  <interceptor-ref name="basicStack"/>
  <result name="success" type="chain">otherAction</result>
</action>

<action name="otherAction" class="com.examples.OtherAction">
  <interceptor-ref name="chain"/>
  <interceptor-ref name="basicStack"/>
  <result name="success">good_result.ftl</result>
</action>
```

This examples chains two actions but only one property from first action is copied to another

```
<action name="someAction" class="com.examples.SomeAction">
  <interceptor-ref name="basicStack"/>
  <result name="success" type="chain">otherAction</result>
</action>

<action name="otherAction" class="com.examples.OtherAction">
  <interceptor-ref name="chainStack">
    <param name="chain.includes">prop1</param>
  </interceptor-ref>
  <result name="success">good_result.ftl</result>
</action>
```