

# Interceptor Configuration

Interceptors allow you to define code to be executed before and/or after the execution of an Action method. (The "Filter" pattern.) Interceptors can be a powerful tool when developing applications. There are many, many use cases for Interceptors, including validation, property population, security, logging, and profiling.

|                            |                                                             |
|----------------------------|-------------------------------------------------------------|
| <b>Validation</b>          | Examine input for correctness                               |
| <b>Property Population</b> | Transfer and convert input to object properties             |
| <b>Logging</b>             | Journal details regarding each action                       |
| <b>Profiling</b>           | Time action throughput, looking for performance bottlenecks |

Interceptors can be chained together to create an Interceptor "Stack". If an action needs to check the client's credentials, log the action, and time the action, all of these routines, and more, could be made part of the same Interceptor Stack.

Interceptors are implemented as Java classes, so each Interceptor has a class name. To make it easier to reference Interceptors, each class can be registered with the framework and given a unique, simpler name.

## Registering Interceptors

```
<interceptors>
  <interceptor name="security" class="com.company.security.SecurityInterceptor"/>
  <interceptor-stack name="secureStack">
    <interceptor-ref name="security"/>
    <interceptor-ref name="defaultStack"/>
  </interceptor-stack>
</interceptors>
```

- ✔ Individual Interceptors and Interceptors Stacks can be "mixed and matched" in any order when defining an Interceptor Stack.
- ✔ The framework will invoke each Interceptor on the stack **in the order it is defined**.

Most applications will define a default Interceptor Stack, such as

```
<default-interceptor-ref name="secureStack"/>
```

but any action may also define its own local stack.

## A local Interceptor Stack

```
<action name="VelocityCounter"
class="org.apache.struts2.example.counter.SimpleCounter">
  <result name="success">...</result>
  <interceptor-ref name="defaultComponentStack"/>
</action>
```

The default configuration (`struts-default.xml`) sets up a default Interceptor Stack that will work well for most applications.

💡 For more, see [Interceptors](#).