

How do we get access to the session

You can obtain the session attributes by implementing `SessionAware` or getting them from the `ActionContext`.

Implementing `SessionAware`

★ This is the preferred mechanism: it makes unit testing easier by allowing simple injection of session attributes instead of having to mock the action context or go through an entire request process.

- Ensure that the action's stack includes the `servletConfig` interceptor.
 - ⓘ *The default stack includes `servletConfig`.*
- Ensure the action implements the `SessionAware` interface.
 - ⓘ The `SessionAware` interface defines a `setSession` method that sets the session attributes into the action.
- Changes to the action's session map are reflected in the underlying `HttpSession`. You may query, insert, and remove session attributes as needed.

ⓘ The `servletConfig` interceptor looks for actions implementing `SessionAware` during request processing. When it finds them it passes the map of session attributes using the action's `setSession` method.

✓ To unit test a `SessionAware` Action, create your own `Map` with the pertinent session attributes and call `setSession` as part of the test's setup method.

Ask the `ActionContext`

Session attributes are available from the `ActionContext` instance, a `ThreadLocal`.

```
Map attributes = ActionContext.getContext().getSession();
```

@see [struts-default.xml](#)

@see [org.apache.struts.action2.interceptor.SessionAware](#)

@see [org.apache.struts.action2.interceptor.ServletConfigInterceptor](#)