

Accessing application, session, request objects

The framework provides several access helpers to access Session, Application, Request scopes.

Accessing from Java

All the JEE scope attribute maps can be accessed via `ActionContext`.

Accessing servlet scopes

```
Map attr = (Map) ActionContext.getContext().get("attr");
attr.put("myId", myProp);

Map application = (Map) ActionContext.getContext().get("application");
application.put("myId", myProp);

Map session = (Map) ActionContext.getContext().get("session");
session.put("myId", myProp);

Map request = (Map) ActionContext.getContext().get("request");
request.put("myId", myProp);
```



Do not use `ActionContext.getContext()` in the constructor of your Action class. The values may not be set up, and the call may return null for `getSession()`.

We can also access the `HttpServletRequest` and `HttpServletResponse` objects themselves through `ServletActionContext`. In general this isn't recommended as it will tie our action to the servlet specification.

Setting session attribute through session object

```
ServletActionContext.getRequest().getSession().put("myId", myProp);
```

Implementing `ServletRequestAware` or `ServletResponseAware`, combined with the `"servletConfig"` [interceptor](#), is an alternative way to access the request and response objects, with the same caveat.

Accessing from the view (JSP, FreeMarker, etc.)

Request and session attributes are accessed via OGNL using the `#session` and `#request` stack values.

The `#attr` stack value will search the `javax.servlet.jsp.PageContext` for the specified key. If the `PageContext` doesn't exist, it will search the request, session, and application scopes, in that order.

Accessing the Session or Request from a JSP

```
<s:property value="#session.myId" />

<s:property value="#request.myId" />

<s:property value="#attr.myId" />
```