

Development-Component Actions

Overview

The Component Action Request Model is the default means for handling user requests in WebObjects. It is based on the idea of interactive page objects (components).

A page creates some HTML and paints the screen for the user. The component object is stored as part of the user's session, with all of its state. When the user clicks a link or submits a form on the page, the URL has been constructed in such a way that the server can determine which component object was used to create the page originally, and re-uses this object to handle the request.

Thus, you can do quite a good job of forgetting you are on the web, and that the only information the user supplies with each request is the URL, form submission, cookies etc. You can create interactive pages which deal with many requests, simply changing their internal state in response to each one, as though you were writing a non-web application to be used locally.

I'm not familiar myself with JSP, but I'm told by people who are that this makes highly dynamic database editing pages (for example editing several linked tables at once) much easier in WebObjects than JSP. I can confirm this is indeed trivial in WebObjects, although I don't know myself why it is harder in the JSP model.

Note that the Component Action Request Model has some disadvantages. Each URL, to be handled, must find the matching component object in memory. Thus, once the session has timed out, the URL becomes largely incomprehensible and good handling of this situation is difficult. One option is to add additional values as a query string on the URL - this will not break standard request processing but could be used to provide the correct information to recover if the session is gone. Additionally, sessions cannot remember infinite component objects, unless you want to throw memory at the problem, so by default only 30 are stored. This means it is possible for users to backtrack too far and create problems, especially if you are using multiple frames or windows.

For cases where highly interactive pages are not required, you may prefer [Direct Actions](#), which has none of these disadvantages, but provides less help.