

Re: Issues of state

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2007-08/msg01053.html>

- *From:* Steve Holden <steve@xxxxxxxxxxxxxxx>
 - *Date:* Thu, 09 Aug 2007 10:52:14 -0400
-

Cameron Laird wrote:

In article <7xfy2t815n.fsf@xxxxxxxxxxxxxxxxxxxxxxxx>, Paul Rubin <http://phr.cx@xxxxxxxxxxxxxxxxxxx> wrote:

claird@xxxxxxxx (Cameron Laird) writes:

Others have answered this at other levels. In elementary terms, there truly is a difference, Paul, and one that's widely reified: a "desktop client-server" application typically listens through one socket, which therefore constitutes an index of the connection or client, while a Web application communicates through a sequence of independent HTTP transactions. The latter can manage state only to the extent it passes session information around.

Is this significant? In the case of a single user http app running on the same computer as the browser, the server should only listen on 127.0.0.1. Every http hit then almost certainly comes from the same session. If there's doubt, the app can always set a random cookie at the initial screen and check that the cookie never changes.

If there's only a small amount of session state (say up to a few hundred bytes) you can put it entirely into browser cookies and send it on every single http hit.

I'm not sure what we're discussing. Yes, I agree those are mechanisms by which Web applications manage state. Apparently we agree that, in a general Web application, state management, or related persistence, requires **some** mechanism or assumption.

As far as I'm concerned the major issue with trying to have "desktop web apps" compete with true windowed applications is the difficulty of maintaining sensible interactions with the interface. AJAX designs have

Re: Issues of state

increased the interaction level at the expense of greater complexity – there is more state to be transferred, and a much higher interaction rate with the server. But the browser is a terrible front–end for AJAX designs because it doesn't record the state changes that take place as a result of requests for updated InnerHTML content, so if the user is unwise enough to press the "back" button she loses all traces of the non–page interactions that have taken place since the page was loaded.

So "desktop web apps" should ensure that they get displayed in browser windows with minimal user interface decoration. But even then there's always that chance that sophisticated users will use keyboard shortcuts like ALT–left–arrow.

That, in summary, is why my 2004 PyCon paper[1] was subtitled "The Back Button is Not Your Friend".

regards
Steve

[1]: http://www.python.org/pycon/dc2004/papers/18/Setting_A_Context.pdf

—
Steve Holden +1 571 484 6266 +1 800 494 3119
Holden Web LLC/Ltd <http://www.holdenweb.com>
Skype: holdenweb <http://del.icio.us/steve.holden>

----- Asciimercial -----
Get on the web: Blog, lens and tag the Internet
Many services currently offer free registration
----- Thank You for Reading -----

.