

Re: Issues of state (was: Tkinter or wxpython?)

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2007-08/msg00978.html>

- *From:* claird@xxxxxxxx (Cameron Laird)
 - *Date:* Wed, 8 Aug 2007 20:23:45 +0000
-

In article <7x8x8mpn37.fsf@xxxxxxxxxxxxxxxxxxxxxx>, Paul Rubin <<http://phr.cx@xxxxxxxxxxxxxxxx>> wrote:

.
. .
.

I'm not sure what you're getting at in this context. You can write a desktop app where the window system communicates with a gui toolkit through a socket (at least that's how X windows works), or you can write a web app where a browser communicates with an http listener through a socket. What's the difference, as far as application state is concerned?

I haven't used wxpython but for tkinter you'd typically have a gui event loop in its own thread, communicating with the application through queues. Similarly you can use BaseHTTPServer to collect browser hits and get the data out of them with the cgi module functions before passing them to the app. If you want to handle multiple concurrent users you get into the usual issues of web servers, but if you're just doing a single user web implementation as an alternative to a desktop gui, some rudimentary locking is probably enough to stop accidental simultaneous connections.

If the application is simple enough, you can just write it as a cgi and keep the state in disk files.

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.

As you correctly point out, there certainly are plenty of Web applications which cheerfully **do** pass session information. As common as it is, though—and I'm **still** writing CGI on occasion, a decade after I was convinced it was a dead technique—it's not automatic.

Re: Issues of state (was: Tkinter or wxpython?)

HTTP is a stateless protocol, and therefore state management must remain an issue.

.