

Re: Is there an obvious way to do this in python?

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2006-08/msg01036.html>

- *From:* "H J van Rooyen" <[mail@xxxxxxxxxxxxxxxxx](mailto:xxxxxxxxxxxxxxxxx)>
 - *Date:* Fri, 4 Aug 2006 10:10:44 +0200
-

"Dennis Lee Bieber" <wlfraed@xxxxxxxxxxxxxxxxx> wrote:

| On Thu, 3 Aug 2006 14:05:19 +0200, "H J van Rooyen"
| <[mail@xxxxxxxxxxxxxxxxx](mailto:xxxxxxxxxxxxxxxxx)> declaimed the following in comp.lang.python:
|
| > What I mean by this is that the server does stuff that I think belongs on
| the
| > client –
| > like getting involved in the nitty gritty of what the client should
| display –
| > I want the client to be smart enough to do the assembly of the elements of a
| > transaction
| > by itself, going back to the server for data only when its needed – remember
|
| One thing to consider: Where is the separation between the database
| and the client. I believe most textbooks these days tend recommend:
|
| [db-server] <-> [app-server] <-> [client]
|
| (db-server and app-server can be the same hardware; the idea is that
| clients do not have direct access to the database system, and hence the
| back-end can be changed out without affecting any client... also,
| clients don't need to handle database errors, etc.)

Agreed – I would also need this application server to do the server end of my
"terminal management system"

|
| > so I see the client interacting with the server quite a lot, eventually to
| be
| > able do things like auto completion of things like stock codes and
| descriptions,
| > customer details, etc. – but I don't want every keystroke flying over the
| LAN
| > and
| > being handled by the server...
| >

Re: Is there an obvious way to do this in python?

| And where did you see the client obtaining the "completion data" ---
| direct access to some other database tables or did you intend to
| download /all/ possible data.

no just a framework of more or less static stuff like document types, name and address data, and account names and codes that is not subject to continuous change... *grin* I would have to apply *some* skull sweat to the problem...

|
| Typical web-based applications may have a minimal bit of data
| validation running on the client (JavaScript ... things like making sure
| /something/ has been entered into required fields, but not verifying
| that it makes sense), and only when the user clicks on a submit is
| everything sent to the application server, which then generates the
| needed SQL from the data fields for submittal to the database server.
|
| > transaction is completed – like in a banking system, I envisage ways for the
| > client to 'poll' the server to get the state of the last transaction, to
make
| > this possible.
| >
| Bad choice... Upon submittal to the server, there should be a
| mandatory "good/bad" return code...

Here I take umbrage – not bad, good choice – let me elucidate – the return code you are talking about is the normal thing, and it is the dead wrong way to operate if that is all you do – what I am talking about is the failure scenario – the way to make any transactionally based system robust is to have a "GetTranResult" transaction that is routinely used before starting a new transaction, to see if the previous one has completed properly – that way, the client can either – continue with the new transaction, or – resubmit the old one to try to get it to work or fail, or thirdly – advise the user that the previous transaction has failed – it is, when you have thought about the flows and all the possible ways in which they can fail – truly the only way to operate reliably.

The above is the simple minded way to use such a facility –

You could also argue that the time to use this "polling" transaction is after submitting the transaction, but if your return code is to be trusted, its not needed – if the flows complete normally, its not needed – it is just needed for built in recovery, and it works like a charm if both the server and the client try to keep state over power failures and other disastrous events like losing comms halfway through – especially for multi flow transactions that must update different things on the server...

And it does no harm to ask the server the result of your last transaction when you come up after power failure – you can then set the screens and stuff up to the point where the server knows about them and have the user just curse a bit, instead of a lot...

Re: Is there an obvious way to do this in python?

Re: Is there an obvious way to do this in python?

And most importantly, the data is safe (unless the database is corrupted, in which case the likelihood is high that yer screwed in any case) – and on this note – if you have the reversed transaction too – i.e. a kind of "GetPreviousLoggedTransactionResult" from the server to the client, you can make recovery less painful, even in the case of database corruption...

Bad choice indeed! *snorts*

;–)

– Hendrik

.