

Re: [PHP] Concurrency in a multi-user environment

Source: <http://coding.derkeiler.com/Archive/PHP/php.general/2007-11/msg01011.html>

- *From:* stuttle@xxxxxxxxxx (Stut)
 - *Date:* Wed, 21 Nov 2007 15:56:52 +0000
-

Don Proshetsky wrote:

I'm been googling to see if I can find information on how programmers handle concurrency in a multi-user PHP/MySQL environment. Here's a snippet of my app.

I have a form where users can maintain client information such as Name, Address, telephone, email, etc.... I've been wondering how to handle a situation where two users are editing the same record.

One idea is to lock the record when User A enters but this may make it inaccessible for a long period if User A decided to get up and leave for a three hour lunch. Also, does locking a record prevent file locks (when new records need to be added)? Finally, locking a record means I have to convert from MyISAM to InnoDB but this is probably necessary for concurrency.

Another idea is to use a semaphore where User A enters and retrieves a numeric field in the table. Upon leaving, the user locks and checks if the field is the same value. If yes, save, increment the numeric field and leave. If not, it indicates another user entered has edited, saved and left. I'm not crazy about this as it would mean User A could spend 15 minutes updating a record only to get a message that User B already edited it.

Does anyone have any user friendly suggestions or can give me some URLs where I can read up on this?

Personally I would approach this from a non-locking perspective. Store a last updated timestamp with each row. Keep a copy of the data you presented to the user when they started editing. When a modification request comes in you...

- 1) Lock the row
- 2) Get the row
- 3) Compare it to the incoming data and your stored copy
 - You can work out which fields the user has changed by comparing the incoming data with the stored copy
 - You can work out which fields have been changed since the user started editing by comparing the stored copy with the latest you got in step 2
 - You can use the intersection of these two to see if there are any conflicts
- 4) If there are conflicts, show the user and let them make a decision
- 5) Update and unlock the row

Re: [PHP] Concurrency in a multi-user environment

Note that step 4 actually contains a copy of all the steps again, but before you start you update the stored copy with the latest data you fetched in step 2.

Locking is bad, m'kay!

Hope that makes sense.

–Stut

—

<http://stut.net/>

.