Turning Pages into Crons

Occasionally you may run into an issue with a complex report that just isn’t going to complete in a reasonable amount of time for a user clicking a link.

There is no “right” way to organize these special crons but for myself I created a httpdocs/cron\_reports folder to separate these reporting crons used to spit out Excel spreadsheets for various employees from the crons that run to process data for storage in the data warehouse.

The start of the cron is the same as any other cron

$opts = **'h:'**;  
  
$options = *getopt*($opts);  
  
$\_HOST = **isset**($options[**'h'**]) ? $options[**'h'**] : **''**;  
  
**if**(!$\_HOST) {  
 **exit**(**'-h<host> required'**);  
}  
  
**require\_once '../index.php'**;

But then we include the code behind of the page we want to turn into a cron

**require\_once '../pages/Marketing/ActiveCustomers/ActiveCustomers.code.php'**;

And finally create a simple class to process the page and export the file

**class** GetActiveCustomers **extends** SafeClass  
{  
 **public function** Run()  
 {  
 Log::*Insert*(**'DoInit'**, **true**);  
 ActiveCustomers::*DoInit*();  
  
 Log::*Insert*(**'DoGet'**, **true**);  
 ActiveCustomers::*$Limit* = 0; *// no limit* ActiveCustomers::*DoGet*();  
  
 Log::*Insert*(**'ExportToXLS'**, **true**);  
 ActiveCustomers::*DoExportToXLS*();  
 }  
}  
  
Log::*Insert*(**'Start'**, **true**);  
  
$cron = **new** GetActiveCustomers();  
$cron->Run();  
  
Log::*Insert*(**'End'**, **false**);

QuickDRY will recognize that “DoExportToXLS” was run from the command line and save the file to disk rather than try to push it as a download.

Whatever name is configured for the filename in DoExportToXLS is what it will be saved as.

Since the properties for ActiveCustomers are public, we can use the command line options to gather settings and then configure the page model accordingly. For this particular page, DoInit and DoGet handle getting the report information and no user input is required to configure it.

Notice that the public property $Limit is set to zero. This is because by default it is set to a relatively low number to allow a sample report to quickly render. We override that default value in this cron to get the full report which takes several minutes to create.

This is one of the key advantages to using the PageModel design: we can easily reuse code as situations arise.