

7 Online Cron Job Tips

from OnlineCronServices.com

As online services continue to expand, so does the need for online script scheduling tools. Online cron services are an excellent way to automate and simplify your website administration. More and more they are required to power the heartbeat of major web applications.

Experienced users of online cron services have developed a few tricks of the trade over the years that can save some time and headache.

1. Protect Your Script

One weakness of online cron jobs is that the command to be executed is a public URL. This means that any one with Internet access could potentially call your job URL and therefore execute your script. For some this is of no real concern but for others this can be very problematic.

Using query string tokens can help you prevent unauthorized calls on your script. Simply add a query string parameter for your token, code or password and provide the value. Make the secret code configurable in your script settings so you can change it as needed.

Basic HTTP authentication can provide some degree of protection. You'll need to setup the page or directory in which your script resides to require a username and password for access. Some online cron services allow you to provide this username and password through a method originally used in the FTP protocol. Here is an example.

```
http://username:password@myownwebsite999.com:80/myscript.php
```

This URL will call the URL and automatically provide the supplied username and password.

Some random calls to your web script can happen unintentionally through robots. Robots usually find your page when they see it listed on another page somewhere on the Web. To prevent this, avoid referencing the URL of your script from public web pages.

2. Problem Escalation Plan

Have you considered what will happen when your script fails? Will the script fail gracefully? Will the administrator be notified?

These can be very important questions when creating your online cron job. Use error trapping and handling techniques if possible. Consider coding your script to send an email when an unhandled exception occurs.

It also good to remember problems that can occur with your service provider. What happens when you leave on extended vacation? If your email address changes, will you stop getting your job results?

Providers also have occasional down time, expected and unexpected. This is a fact.

Take responsibility for your script and plan for handling these down times.

3. create a test interface

A test interface for your script can save you hours of debugging and can avoid public errors when you are trying to debug a problem. Use a switch in the query string of your job URL to switch enable your test interface. As an example, you may want to add a parameter on the end of your URL to switch on the test interface.

`http://myownwebsite999.com/myscript.php?code=sge8nK37&debug=1`

A URL with a debug parameter like the one above may carry out all the functionality but avoid the final writing of results to a database. Or it may generate your newsletter automatically but only send it out to the admin.

Use the debug/test interface to return verbose results from your script. This will allow you to switch this verbose information off by merely editing your URL. If your service charges you for bandwidth this little switch will end up saving you money.

You may also consider scheduling a parallel test call to regularly examine the functionality and health of your script. Consider scheduling a version of your script to run every 6 months with a checkup parameter appended (i.e. &checkup=1). Instead of running your usual script functions, it runs a checkup function that will check resource usage, logs and other indicators and then return the results in the script response or send it all in an email. This works great for checking the remaining disk space available or verifying that a 3rd party username and password are still working.

4. use asynchronous code where possible

Most online cron services limit the length of time your script is allowed to run. If you have a script that requires a longer running time, consider using some asynchronous techniques. Some script languages allow you to tell the script engine to continue to process the script even if the HTTP request is cancelled. It's a good idea to implement this feature if you expect your script to run longer than the timeout period of the online cron service.

5. Monitor your resource usage

Remember that when you schedule your job to run at midnight every night, there may be a couple hundred other people thinking the same thing. If the exact time is not critical, consider scheduling your job for 11:56 PM or 12:03 AM. You'll find there are much fewer jobs scheduled at this time and your job will often run faster and smoother than when you are wrestling for that coveted midnight position. You may want to avoid using minutes that are multiples of five (i.e. 5, 10, 15, 20, 25).

Keep in mind the amount of output your job will generate. The more data in your output, the longer the potential running time of your job will be. If possible keep your response output short.

6. Queue your work

Whatever job your script is performing, consider implementing finite limits to the work. For example, instead of creating a job to email everyone on your mailing list daily, consider emailing the next 100 people on your list every hour. This way your job will continue to run gracefully even when you receive unanticipated spikes in the total addresses on your mailing list. You can also make the limits configurable so you can

easily tweak your job as demand changes. Using the above example, as your mailing list grows you may want to change your job to run every 30 minutes and send email to the next 250 people.

7. Archive your reports and alerts

If your job runs often you may want to consider a plan for archiving the job response. Most services store the results and at least a part of the HTTP response from your job. However, there is often size and time limits for this storage. If the data returned by your job is important, consider a system for backing up this data to another location.

If you would like to provide feedback regarding this article please feel free to contact us by email (feedback@onlinecronservices.com).

For more information about online cron services please visit our website at the URL below.

Online Cron Services
www.onlinecronservices.com