

Perl

List domains to expire next month

Needs:

Class::Date and Date::Parse from CPAN!

```
#!/usr/bin/perl
```

```
use LWP::UserAgent;
use Data::Dumper;
use Class::Date qw(:errors date localdate gmdate now -DateParse);

my $dmapiURL = "https://dmapi.joker.com/request";

my $ua = LWP::UserAgent-> new;
my $req = HTTP::Request->
new(GET => $dmapiURL. '/login?username='.shift(). '&password='.shift());
my $res = $ua-> request($req);

unless ($res-> is_success) {
print "Failed: ", $res-> status_line, "n";
exit -1;
}

my $auth=$res-> as_string;
my $output;

if ($auth =~ /Auth-Sid:s*(w+)/ms) {
$req=HTTP::Request-> new(GET => $dmapiURL. '/query-domain-list?auth-sid='.$1);
$res = $ua-> request($req);

if ($res-> is_success) {
$output=$res-> content;
}
else {
print "Failed: ", $res-> status_line, "n";
exit -2;
}
}
else {
print "Failed: Got no auth-id from DMAPi:n";
print $res-> content;
exit -3;
}

my @lines=split ("n",$output);
```

Perl

```
my $month=now + ("2M");
my %list;

my $p=0;
for my $line(@lines) {
    #Skip first line
    next unless $p or $line =~ /^s*$/;
    unless ($p) { $p=1; next};
    my ($fqdn,$exp)=split ("[ t]+",$line);
    next unless localdate($exp) < $month;
    $list{$fqdn}=$exp;
}

for ( map {$_-> [0]}
      sort {
        $a-> [1] < => $b-> [1]
        ||
        $a-> [2] cmp $b-> [2]
      }
      map {[$_, localdate $list{$_},$_]}
      keys %list ) {
    print "$_:".list{$_}."n";
}
```

Unique solution ID: #1043

Author: Joker.com

Last update: 2012-09-27 15:45