

2019年10月2日

# 時刻表示ウェブページ用 簡易なサーバーのサンプルコード

国立研究開発法人 情報通信研究機構

鳥山 裕史

# Perl 版

```
#!/usr/bin/perl -wT

#####
#                                     #
#  JSON 時刻応答 cgi Perl版（うるう秒：非対応）#
#                                     #
#          Ver. 1.0 02.Oct. 2019  H. Toriyama      #
#                                     #
#####

require 5.002;
use strict;
use Time::HiRes qw( gettimeofday );

my $st = gettimeofday();
my $id = $ENV{HTTP_HOST} ? $ENV{HTTP_HOST} : "unknown";
my $it = $ENV{QUERY_STRING} ? $ENV{QUERY_STRING} : "0";

#it の正当性チェック（簡易）
$it = "0" unless ($it =~ /^¥d{1, 10}$|^¥d{1, 10}¥.¥d{0, 3}$/) ;

# HTTPヘッダ情報を送信
print "Access-Control-Allow-Origin: *¥n";
print "Cache-Control: no-cache, no-store¥n";
print "Content-type: application/json¥n¥n";

printf '{"id": "%s",', $id;
printf ' "it": %s,', $it;
printf ' "st": %.3f}', $st;

exit;
```

# PHP 版

```
<?php
#
#####
#                                     #
#  JSON 時刻応答 cgi PHP版（うるう秒：非対応） #
#                                     #
#      Ver. 1.0 02. Oct. 2019  H. Toriyama  #
#                                     #
#####
#
# HTTPヘッダ情報を送信
header('Access-Control-Allow-Origin: *');
header('Cache-Control: no-cache, no-store');
header('Content-type: application/json');
#
# 引数にtrueを指定すると、マイクロ秒までfloatで取得できる
$st = microtime(true);
$id = (empty($_ENV['HTTP_HOST'])) ? 'unknown' : $_ENV['HTTP_HOST'];
$it = (empty($_ENV['QUERY_STRING'])) ? '0' : $_ENV['QUERY_STRING'];

#it の正当性チェック（簡易）
if( !preg_match('/^\d{1,10}$|^d{1,10}¥.¥d{0,3}$/', $it) ) $it = '0';

printf ( '{"id": "%s",', $id);
printf ( "it": %s,', $it);
printf ( "st": %.03f}", $st);
?>
```

# プログラムの仕様

```
# 【プログラムの仕様】
# たとえば、このプログラムを ntp-a1.nict.go.jp のcgi-bin ディレクトリに
# "json.pl" という名前で設置し、以下の URL
# https://ntp-a1.nict.go.jp/cgi-bin/json.pl?1234567890.123
# でアクセスすると次のような文字列 (JSON 形式データ) が返されます。
#
# {"id": "ntp-a1.nict.go.jp", "it": 1234567890.123, "st": 1234567891.234}
#
# それぞれの値の意味は、次の通りです。
# id: "ntp-a1.nict.go.jp" サーバID (通常はホスト名)
# it: 1232963971.248 発信時刻 (クライアントから送信された時刻)
# st: 1232963971.920 サーバ時刻
# 時刻表記は UNIX Time (POSIX仕様) で、少数以下 3 桁 (ミリ秒まで) です。
# クライアントからの発信時刻を使わない場合は、URLの "?" 以下は省略できます。
#
# 【プログラムの解説】
# Perl の通常の時刻関数では、秒以下の時刻情報が扱えませんが、
# Time::HiRes モジュールにある gettimeofday 関数により、秒以下の
# 時刻情報を獲得しています。
#
# $it で始まる行で、この時刻サーバに送られる引数が適切な形式の数値である
# ことをチェックしています。
#
# このサーバの返すヘッダには、# Access-Control-Allow-Origin: *
# が含まれ、これは「他のドメインからの AJAX 実行を許可する」
# ことを示します。
# その必要がない場合、またはWebサーバの設定で同様のヘッダを出す場合には、
# このprint行は不要です。
```

# おまけ: 簡易版NTP(SNTP)サーバ (Perl 版)

```
#!/usr/bin/perl -w

use strict;
use Socket;
use Time::HiRes qw( gettimeofday );

my $proto = getprotobyname('udp');
socket(NTP, PF_INET, SOCK_DGRAM, $proto) or die "socket: $!";
my $paddr = sockaddr_in( 123, INADDR_ANY );
bind(NTP, $paddr) or die "bind: $!";

my ( $rb, $sb, $vn, $client, $seconds, $microsec );
$sb = "\0" x 48;

while ("You love NTP") {
    ($client = recv(NTP, $rb, 256, 0)) or next;
    ( length($rb) >= 48 ) or next;
    ($seconds, $microsec) = gettimeofday;
    $vn = unpack("N", substr($rb, 0, 4)) & 0x38000000;
    substr($sb, 0, 4) = pack("N", 0x040106F0 | $vn );          # flag
    substr($sb, 4, 4) = pack( "N", 0 );                       # delay
    substr($sb, 8, 4) = pack("N", 0x00000010);                # dispersion
    substr($sb, 12, 4) = "LOCL";                               # Ref ID
    substr($sb, 16, 4) = pack("N", $seconds + 0x83AA7E80);
    substr($sb, 20, 4) = pack("N", 0 );
    substr($sb, 24, 8) = substr($rb, 40, 8);
    substr($sb, 32, 4) = pack("N", $seconds + 0x83AA7E80);
    substr($sb, 36, 4) = pack("N", ($microsec/500000) * 0x80000000 );
    ($seconds, $microsec) = gettimeofday;
    substr($sb, 40, 4) = pack("N", $seconds + 0x83AA7E80);
    substr($sb, 44, 4) = pack("N", ($microsec/500000) * 0x80000000 );

    send(NTP, $sb, 0, $client);
}
```

- 仕様変更が容易なので、LI (leap indicator) 等を変更してクライアントのうるう秒前後の挙動テスト等に
- 故意にオフセットを加え、ずれた時刻が供給された場合のクライアントの挙動テスト等に
- NTP の仕様を厳密に満たすわけではないが、大抵の場合、支障なく動作する
- 負荷が大きい場合には、そこそこの精度
- 特権ユーザで起動する必要あり (特権ポート使用)