

IoTサーバで簡易データロガーの作成

2017/10/31

石山隼行

ishiyama@hnl.t.u-tokyo.ac.jp

Web上からPiサーバにデータを送信することでロギング（保存）し、またWebに表示する簡単な仕組みをPerlで作成してみました。

<http://iotmlab.ddns.net/~IoT00/cgi/getpost.pl>

↑動作するPerlスクリプトはココにあります。

実際にブラウザからアクセスすると

入力データ:

ファイル内容:

a = 12345 / b = 22222 /

a = 55555 / b = 22222 /

このような表示になるはずです（ファイル内容はサンプルデータ）

入力データがあればその内容をファイルに保存し、なければただこのように表示するだけの仕掛けです。

データの inputs は、このURLに **?a=12345&b=33333...** のようなテキストを追加してWebアクセスするだけでOK。（getメソッド（後述））

つまり、

<http://iotmlab.ddns.net/~IoT00/cgi/getpost.pl?a=55555&b=22222>

にWebアクセスすると、



このように入力され、ファイルに追加されていくわけです。

※もちろん、a= や b= だけでなく、どんなパラメータでもOKです。ただし、WebのURL部分で通信するので、あまり長い（255文字程度）文字列では最後のほうが切れます。そのようなデータは後述するpostメソッドを使うといいでしょう。

Get/Postメソッドとは

WEBにおいて基本的にサーバとのやりとりはHTTPというプロトコルを用います。ブラウザからサーバへの要求することをリクエストといいます。サーバからブラウザへ応答することをレスポンスといいます。HTTPプロトコルの中で、簡単なデータをサーバにリクエストする方法（メソッド）として、GetとPostというメソッドが用意されています（ほかにもいろいろあります）サーバはこうしたリクエストに応じて返事（レスポンス）をするわけです。

Getメソッド

GETはURL（http://iotmlab.ddns.net/・・・）に付加してリクエストします

GET: /foo/bar?p=.....&u=.....

↑このような書式になります。URLとデータ部の間は？（半角）、データ内のパラメータとパラメータの間は&（半角）で区切る必要があります。データは文字列ですから、数字以外もOK（ただし日本語は変換しないといけけないので面倒）
？のあと、パラメータ名をいれずにただテキストでデータを送ることもできます。このばいはkeywordとして処理されます。

Postメソッド

POSTはBodyに含めてリクエストします。

POST: /foo/bar

param:p:.....,u.....

よくあるパターンでは、HTML内のFormタグを使ってリクエストをします

```
1 <form action="http://foo.com" method="post">
2   <input name="say" value="Hi">
3   <input name="to" value="Mom">
4   <button>Send my greetings</button>
5 </form>
```

サーバから見るとこのリクエストは

```
1 | POST / HTTP/1.1
2 | Host: foo.com
3 | Content-Type: application/x-www-form-urlencoded
4 | Content-Length: 13
5 |
6 | say=Hi&to=Mom
```

このような形になります。

※今回作成したCGIでもこのPOSTメソッドを処理できます。必要な場合は各自挑戦してみてください。

作成したCGI詳細

getpost.pl

```
#!/usr/bin/perl

use strict;
use warnings;
use CGI;

my $datafile='./data.txt';

#データファイル指定。chmod a+w しておくこと
open(FH,">>$datafile");

# POST / GET パラメータを取得
my $q = new CGI;

#HTMLヘッダ出力
print $q->header(-charset=>"utf-8");
print $q->start_html(-title=>"CGI Test");

# パラメータ名を指定して取得する場合はこうします
#my $param1 = $q->param('name1');
#my $param2 = $q->param('name2');

# 全てのパラメータを取得する場合
print $q->strong("入力データ:"),$q->br;
for my $param_name ($q->param) {
    my $inP = $q->param($param_name);
```

```

my $ooo = " $param_name = $inP ";
print $ooo. '<br>';
print FH $ooo. ' / ' ;
}
print FH "\n";
close(FH);

print $q->br,
      $q->strong("ファイル内容:"),$q->br;

open(FH,"$datafile");
while (my $line =<FH>){
    chomp($line);
    print "$line <br>";
}
close (FH);

print $q->end_html;
exit;

```

IoT00 ユーザのディレクトリ /public_html に cgiというディレクトリを作成し、その中でPerl プログラムを作成しています。

```

$ mkdir /public_html/cgi
$ cd /public_html/cgi

```



```

IoT00@pi-server:~/public_html/cgi $ ls
data.txt  getpost.pl

```

なお、データファイルとしてdata.txtというファイルが必要です。

```
$ touch data.txt
```

としてあらかじめ作成し、

```
$ chmod a+w data.txt
```

でdata.txtファイルに書き込み権限を与えておきます。

その後、上記のPerlスクリプトを作成しました。

※各自打ち込んでくてもよいですが、IoT00ユーザのディレクトリ内からコピーしてもっていくのも手です。

perlスクリプトには実行権限をあたえておいてください。

ちなみにdata.txtには、

```
IoT00@piserver:~/public_html/cgi $ cat data.txt
a = 12345 / b = 22222 /
a = 55555 / b = 22222 /
```

このようにデータがテキストとして追加保存されていきます。
このデータをエクセルやGoogleスプレッドシートで集計するなどしても良いでしょう。

なお、スクリプトはWeb上で動作させた際、どんなエラーが発生してもinternal Server Errorとなってしまう、デバッグが難しくなりますが、ターミナル上から実行すればエラーメッセージを見ることができるのでおすすめです。

Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator at webmaster@localhost to inform them of the time this error occurred, and the actions you performed just before this error.

More information about this error may be available in the server error log.

Apache/2.4.25 (Raspbian) Server at iotmlab.ddns.net Port 80

ブラウザでこんなエラーになっても、

```
IoT00@piserver:~/public_html/cgi $ perl getpost.pl
syntax error at getpost.pl line 45, near "->end_html
exit"
Execution of getpost.pl aborted due to compilation errors.
IoT00@piserver:~/public_html/cgi $
```

ターミナルから実行すればエラーメッセージを見ることができます。

```
43
44 print $q->end_html
45 exit;
```

この例では44行目の最後にセミコロンが抜けていてシンタックスエラーが発生していました。

大変簡単な例ですが、こうした方法でWebサーバとクライアントとの通信とロギングを行うことができます。

ブラウザの画面を凝ったり、PerlではなくPHPやRubyなどのより高級な言語をつかって難しい処理をしてもよいかとおもいます。

Piサーバはスピードは遅いですが、機能的には大抵のことができるようにしてありますので、各自創意工夫して面白いものを作ってみてください。

ー以上ー