

# Nginx+PHP-fpmでPHPをNginxで実行できるようにするまで

どうも、シローです。

今回は前回(<https://shiro-secret-base.com/?p=458>)に続いてPHPを使ってサイトのアクセスカウンターを実装しようと思います。

## 作業環境

OS	Ubuntu 18.04.3 LTS
nginx -v	1.14.0 (Ubuntu)
phpfpm -v	PHP 7.3.9 (fpm-fcgi) (built: Sep 17 2019 09:26:12)

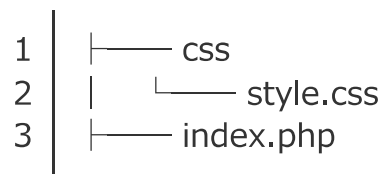
## 今回作るもの

今回作るページはこちらです。



ページを開いた回数をブラウザで表示するようにします。

## ディレクトリ構成



## ソース一覧

index.php

```
1  <?php
2  session_start();
3  if (!isset($_SESSION['count'])) {
4      $_SESSION['count'] = 0;
5  } else {
6      $_SESSION['count']++;
7  }
8  $access_count = $_SESSION['count'];
9  ?>
10 <!doctype html>
11 <html>
12   <head>
13     <meta charset="utf-8" />
14     <link href="css/style.css" rel="stylesheet" />
15   </head>
16   <body>
17     <div class="wrapper">
18       <h1>猫さん</h1>
19       <div class="cat">
20         
21         <div class="reply">
22           <p><?php echo $access_count; ?>回目のアクセスにゃ</p>
23         </div>
24       </div>
25     </div>
26   </body>
```

27 | </html>

---

style.css

```
1  .wrapper {
2    position: absolute;
3    background: #dcdcdc;
4    width: 40%;
5    top: 100px;
6    left: 30%;
7    text-align: center;
8  }
9  .wrapper img {
10   display: block;
11   width: 100%;
12   height: auto;
13   padding: 30px;
14   box-sizing: border-box;
15 }
16 .cat {
17   position: relative;
18 }
19 .reply {
20   position: absolute;
21   top: 10%;
22   right: 10%;
23   width: 30%;
24   min-height: 86px;
25   font-size: 1em;
26   padding: 20px;
27   background: white;
28   border-radius: 50px;
29   box-sizing: border-box;
```

```
30 | }
```

---

## PHP-fpmをインストールする

**PHP-fpm(FastCGI Process Manager)**はPHPをCGIとして実行するためのツールです。

NginxでPHPを実行するためにはPHP-fpmをインストールして、NginxからPHP-fpmへの通信を行うように設定しなければ行けないのです。

### インストール

```
1 | $ sudo apt-get install php-fpm
```

### PHP-fpmの設定ファイルの確認

```
1 | $ sudo find / -name *php-fpm*
2 | /etc/php/7.2/fpm/php-fpm.conf <= これ
3 | .
4 | .
5 | .
```

PHP-fpmをインストールされていることを確認したら。今度はNginx側の設定を編集します。

# NginxにPHPファイルにアクセスされた場合、PHP-fpmにつなぐ設定を追加する

前回と同じく/etc/nginx/sites-available/sample-site.confを編集します。

```
1  server {
2      listen 8888;
3      server_name sample-site.com;
4      root /var/www/html/sample-site;
5      access_log /var/log/nginx/sample-site/access_log;
6      error_log /var/log/nginx/sample-site/error_log;
7      location / {
8          try_files $uri $uri/index.html = 404;
9      }
10
11     location /image/ {
12         root /var/www;
13     }
14
15     location ~* \.php(/|$) {
16         include fastcgi_params; # <= phpに渡すパラメータに関する設定ファイルを読み込んでいる
17         if (!-f $document_root$fastcgi_script_name) {
18             return 404; # <= 実行するphpのファイルパスが見つからない場合は404を返す
19         }
20         fastcgi_pass unix:/var/run/php/php7.2-fpm.sock; # <= php-fpmに接続する
21         fastcgi_split_path_info ^(.+\.php)(/.*)$; # <= ./test.php/article/1 のようなURIを
22             #実行するファイル名($fastcgi_script_name)と
23             #phpに渡すパス情報パラメータ($fastcgi_path_info)に分割する
24         fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name; # <= 実行するスクリプトのパスを ル
25    ートディレクトリ/実行するファイル名 にする
26         fastcgi_param PATH_INFO $fastcgi_path_info; # <= パス情報パラメータ
27     }
```

15~26行目がPHP-fpmに関する記述です。

15行目のlocationディレクティブで拡張子がphpで始まるパスに対してphp-fpmに接続するように指定してます。

"~\*"は大文字小文字を区別しない正規表現を使うという意味です。"\*"を外した場合("~")、大文字小文字を区別する正規表現になります。

16行目ではphpを実行するのに必要なパラメータに関する設定ファイル(/etc/nginx/fastcgi\_params)を読み込んでいます。

後ほどfastcgi\_paramディレクティブで一部のパラメータを設定しているので、順番としてfastcgi\_paramディレクティブより前に記述した方が安全です。

17~19行目では実行するphpファイルのパスが見つからなかった場合に404エラーを返すようになっています。

**try\_filesディレクティブを使わないのはパスが./test.php/article/1ようにphp拡張子のうしろにパス情報が繋がっている場合などにもphpファイルを実行できるようにするため**です。

20行目のfastcgi\_passディレクティブではphp-fpmを起動しているサーバに接続しています。（今回は同じサーバなのでunixによるソケット通信）

外部のphpサーバにつなぎたい場合は、そのサーバのIPアドレスとのphp-fpmように公開しているポート番号をつなげたドメインを指定します。

21~23行目ではURIをphp拡張子の前後で[実行するphpファイル名(\$fastcgi\_script\_name) | phpに渡すパス情報パラメータ(\$fastcgi\_path\_info)]に分割します。

24行目のfastcgi\_paramディレクティブでは、実行するファイルのパス変数(SCRIPRT\_NAME)に

"ルーティングディレクトリ(\$document\_root)"/"実行するphpファイル名(\$fastcgi\_script\_name)" を設定しています。

25行目のfastcgi\_paramディレクティブでは、phpに渡すパス情報パラメータの変数(PATH\_INFO)に"\$fastcgi\_path\_info"に渡してます。

## Nginxを再起動

編集がおわったらNginxを再起動します。

```
1 | $ sudo service nginx restart
```

## PHPが実行できるか試してみる

index.phpと同じ階層にtest.phpファイルを作成します。

```
1 | <?php
2 | echo('<pre>');
3 | var_export($_SERVER);
4 | echo('</pre>');
```

sample-site.com:8888/test.phpでアクセスします。



← → ↺ ⓘ 保護されていない通信 | sample-site.com:8888/test.php

```
array (
  'USER' => 'www-data',
  'HOME' => '/var/www',
  'HTTP_COOKIE' => 'PHPSESSID=kqp2fg986saoqjd7m98bm62u27',
  'HTTP_ACCEPT_LANGUAGE' => 'ja,en-US;q=0.9,en;q=0.8',
  'HTTP_ACCEPT_ENCODING' => 'gzip, deflate',
  'HTTP_ACCEPT' => 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3',
  'HTTP_USER_AGENT' => 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.75 Safari/537.36',
  'HTTP_UPGRADE_INSECURE_REQUESTS' => '1',
  'HTTP_CONNECTION' => 'keep-alive',
  'HTTP_HOST' => 'sample-site.com:8888',
  'PATH_INFO' => '',
  'SCRIPT_FILENAME' => '/var/www/html/sample-site/test.php',
  'REDIRECT_STATUS' => '200',
  'SERVER_NAME' => 'sample-site.com',
  'SERVER_PORT' => '8888',
  'SERVER_ADDR' => '127.0.0.1',
  'REMOTE_PORT' => '60442',
  'REMOTE_ADDR' => '127.0.0.1',
  'SERVER_SOFTWARE' => 'nginx/1.14.0',
  'GATEWAY_INTERFACE' => 'CGI/1.1',
  'REQUEST_SCHEME' => 'http',
  'SERVER_PROTOCOL' => 'HTTP/1.1',
  'DOCUMENT_ROOT' => '/var/www/html/sample-site',
  'DOCUMENT_URI' => '/test.php',
  'REQUEST_URI' => '/test.php',
  'SCRIPT_NAME' => '/test.php',
  'CONTENT_LENGTH' => '',
  'CONTENT_TYPE' => '',
  'REQUEST_METHOD' => 'GET',
  'QUERY_STRING' => '',
  'FCGI_ROLE' => 'RESPONDER',
  'PHP_SELF' => '/test.php',
  'REQUEST_TIME_FLOAT' => 1585927638.681475,
  'REQUEST_TIME' => 1585927638,
)
```

← バス情報

← 実行するファイルパス

← プロジェクトのルートディレクトリ

← 実行するファイル名

PATH\_INFO,SCRIPT\_FILENAME,DOCUMENT\_ROOT,SCRIPT\_NAMEに注目するとfastcgi\_paramで渡された情報が入っていることが確認できます。

次にsample-site.com:8888/test.php/article/1でアクセスします。

← → ↺ ⓘ 保護されていない通信 | sample-site.com:8888/test.php/article/1

```

array (
  'USER' => 'www-data',
  'HOME' => '/var/www',
  'HTTP_COOKIE' => 'PHPSESSID=kqp2fg986saoqjd7m98bm62u27',
  'HTTP_ACCEPT_LANGUAGE' => 'ja,en-US;q=0.9,en;q=0.8',
  'HTTP_ACCEPT_ENCODING' => 'gzip, deflate',
  'HTTP_PURPOSE' => 'prefetch',
  'HTTP_ACCEPT' => 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3',
  'HTTP_USER_AGENT' => 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.75 Safari/537.36',
  'HTTP_UPGRADE_INSECURE_REQUESTS' => '1',
  'HTTP_CONNECTION' => 'keep-alive',
  'HTTP_HOST' => 'sample-site.com:8888',
  'PATH_INFO' => '/article/1',
  'SCRIPT_FILENAME' => '/var/www/html/sample-site/test.php',
  'REDIRECT_STATUS' => '200',
  'SERVER_NAME' => 'sample-site.com',
  'SERVER_PORT' => '8888',
  'SERVER_ADDR' => '127.0.0.1',
  'REMOTE_PORT' => '60644',
  'REMOTE_ADDR' => '127.0.0.1',
  'SERVER_SOFTWARE' => 'nginx/1.14.0',
  'GATEWAY_INTERFACE' => 'CGI/1.1',
  'REQUEST_SCHEME' => 'http',
  'SERVER_PROTOCOL' => 'HTTP/1.1',
  'DOCUMENT_ROOT' => '/var/www/html/sample-site',
  'DOCUMENT_URI' => '/test.php/article/1',
  'REQUEST_URI' => '/test.php/article/1',
  'SCRIPT_NAME' => '/test.php',
  'CONTENT_LENGTH' => '',
  'CONTENT_TYPE' => '',
  'REQUEST_METHOD' => 'GET',
  'QUERY_STRING' => '',
  'FCGI_ROLE' => 'RESPONDER',
  'PHP_SELF' => '/test.php/article/1',
  'REQUEST_TIME_FLOAT' => 1585928761.337105,
  'REQUEST_TIME' => 1585928761,
)

```

← バス情報

← 実行するファイルパス

← プロジェクトのルートディレクトリ

← 実行するファイル名

すると、同じくtest.phpのページが表示されます。

SCRIPT\_FILENAMEを見ると"/**var/www/html/sample-site/test.php**"になっている

"/test.php/article/1"がfastcgi\_split\_path\_infoで分割された"/test.php"(\$fastcgi\_script\_name)が/var/www/html(\$document\_root)に連結して出力されています。

また、PATH\_INFOを見ると"/test.php/article/1"の "/article/1"が出力されています。

## sample-site.com:8888/index.phpにアクセス

では、いよいよsample-site.com:8888/index.phpにアクセスしてみます。

すると、



と言うように、ページを開いた回数をサーバで保存しているので、アクセスするたびにアクセス回数がカウントアップされたのを確認できれば完了です。

## まとめ

- NginxでPHPを実行するにはPHP-fpmに接続する必要がある
- PHPを実行するときのパラメータを渡すためにfastcgi\_paramsを読み込む必要がある
- fastcgi\_paramディレクティブで追加でパラメータを設定できる
- SCRIPT\_FILENAMEでは実行するファイルパスを指定する必要がある

- /test.php/article/1のようにphpファイルの後ろにパスが続いてもphpファイルを実行するためにはfastcgi\_split\_path\_infoディレクティブでファイルとパスのパラメータを分割する必要がある



## nginx実践入門

nginxを現場で活用するための知識を、実践的なノウハウを交えて解説した書籍です。

nginxのインストール方法や基本的な設定方法からはじめ、nginxを利用した「静的コンテンツ配信サーバ」「HTTPSサーバ」「Webアプリケーションサーバ」「大規模コンテンツ配信システム」の構築方法をそれぞれ詳しく紹介しています。後半ではnginxサーバのモニタリングやログの収集、そして軽量スクリプト言語Luaでnginxを拡張する方法について解説しているので、nginxをこれから使う方はもちろん、さらに活用したい方にもお勧めです。

[Amazon!](#)

