



[ホーム](#) [会社概要](#) [ソリューション](#) [開発実績](#) [お問い合わせ](#) [社員ブログ](#)

[お知らせ](#) [採用情報](#)

PHPで簡易チャット

Posted on **2016年6月1日**

勉強としてPHPでチャットを作れるかというところ、Comet(コメット)と言う技術があるらしいのでそちらを利用して作成してみます。

Comet(コメット)は、リクエストを受けたら直ぐにレスポンスを返さずに待機して、なんらかのイベントが発生したらレスポンスを返すというような事ができるそうです。

今回の場合だと、ページを開いたときにサーバー側にリクエストを送り、リクエストを受けたサーバーでチャットログが更新されるまで待機して、更新されたら最新のログを返すことができます。これによりサーバーへのアクセス数を少なくできます。

ではいきなりですが、まずはじめにPHPの処理を書きます。

```
1 <?php
2 $request_method = filter_input(INPUT_SERVER, "REQUEST_METHOD");
3
4 if($request_method === "POST")
5 {
6     //送信されたメッセージを取得
7     $data = file_get_contents("php://input");
8     $message = htmlspecialchars($data)."\r\n";
9
10    //ログファイルに追記する
11    if(file_put_contents("./chat.log",$message,FILE_APPEND) === false){
12        echo '';
13    } else {
14
15        //追記できた場合は追記後の内容を返す
16        $file = file_get_contents("./chat.log");
17
18        if ($file !== false) {
19            echo nl2br($file);
20        } else {
21            echo '';
22        }
23    }
24 }
```

```
1 <?php
2 //ファイルが存在する場合はその内容を返す
3 if (file_exists("./chat.log")) {
4     $file = file_get_contents("./chat.log");
5
6     if ($file !== false) {
7         echo nl2br($file);
8     } else {
9         echo "";
10    }
11 } else {
12     echo "";
13 }
```

```
1  <?php
2  $is_not_exists = ''; //ファイルが存在しなかったかどうか
3  $before_file = ''; //変更前のファイルの内容
4
5  //10秒に一回ループする
6  while (true) {
7      if (file_exists("./chat.log")) {
8
9          //リクエストのはじめにログフの内容を取得する
10         if (empty($before_file)) {
11
12             //ログを取得
13             $before_file = file_get_contents("./chat.log");
14
15             //ループの途中でログファイルが作成された場合はその内容を返す
16             if ($is_not_exists) {
17                 echo nl2br($before_file);
18                 break;
19             }
20
21             continue;
22         }
23
24         //ログを取得
25         $file = file_get_contents("./chat.log");
26
27         if ($file !== false) {
28
29             //はじめに取得した内容から変更がされている場合はその内容を返す
30             if ($before_file !== $file) {
31                 echo nl2br($file);
32                 break;
33             }
34         }
35     } else {
36         //ファイルが存在しない
37         $is_not_exists = true;
38     }
39
40     //10秒間処理をとめる
41     sleep(10);
42 }
```

以下ざっくりとした説明です。

sendMessage.phpは、送信されたメッセージをchat.logに書き込みます。

getLog.phpは、ページ読み込み完了時に実行されて最新のチャットログを返します。

updateLog.phpは、chat.logが更新されるまでループで待機し、更新されたらファイルの内容を返します。

次にHTMLを用意します。

html

```
1  <html>
2    <head>
3      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4      <title>チャット</title>
5    </head>
6    <body>
7      <div id="chat_log">
8      </div>
9      <input type="text" size="50" id="message" />
10     <button type="button" id="send">送信</button>
11   </body>
12 </html>
```

そしてjavascriptです、

```
<script type="text/javascript">
    window.onload = function(){
        getLog();

        //送信ボタン押下時
        document.getElementById("send").addEventListener("click",function(){
            var text = document.getElementById("message");

            if(text.value !== null && text.value !== ""){
                sendMessage(text.value); //メッセージ送信
                text.value = ''; //テキストボックスをクリアする
            }
        });

    };

    //メッセージの送信処理
    function sendMessage(message) {
        var xhr = new XMLHttpRequest();
        xhr.open("POST","/chat/sendMessage.php");
        xhr.send(message); //メッセージを送信する

        xhr.onreadystatechange = function () {
            //受信時の処理
            if(xhr.status === 200 && xhr.readyState === 4){
                //返された最新のチャットログを画面に表示
                document.getElementById("chat_log").innerHTML = xhr.responseText;
            }
        };
    }

    //初期表示時のログ取得
    function getLog() {
        var xhrGet = new XMLHttpRequest();
        xhrGet.open("GET","/chat/getLog.php");
        xhrGet.send();

        xhrGet.onreadystatechange = function () {
            //受信時の処理
            if(xhrGet.status === 200 && xhrGet.readyState === 4){
                //返された最新のチャットログを画面に表示
                document.getElementById("chat_log").innerHTML = xhrGet.responseText;
                //再度リクエストを送る
                updateLog();
            }
        };
    }
}
```

//ログを随時更新する

```
function updateLog() {  
    var xhrUpdate = new XMLHttpRequest();  
    xhrUpdate.open("GET", "/chat/updateLog.php");  
    xhrUpdate.send();  
  
    xhrUpdate.onreadystatechange = function () {  
        //受信時の処理  
        if(xhrUpdate.status === 200 && xhrUpdate.readyState === 4){  
            //返された最新のチャットログを画面に表示  
            document.getElementById("chat_log").innerHTML = xhrUpdate.responseText;  
            //再度リクエストを送る  
            updateLog();  
        }  
    };  
}  
</script>
```

sendMessage()は、送信ボタンが押されたら、サーバー側にメッセージを送信し、返された最新のログを画面に表示する。

getLog()は、ページ読み込み完了時に実行され、最新のログを取得した後updateLog()を実行する。

updateLog()が前述のCometを利用した処理で、下記の処理を繰り返し行っています。

1. javascriptでサーバー側にリクエストを送る
2. 受け取ったサーバー側のupdateLog.phpが実行され、ログが更新されるまで待機
3. ログが更新されたら、ログの内容をクライアント側へ返す
4. javascriptで返されたログの内容を画面表示
5. 再度updateLog()を実行し、(1)から順に処理を行う

以上がComet(コメット)を利用したチャットの作り方でした。

利用できる場面は少なそうですが、良い勉強になったかと思います。

参考URL：http://rewish.jp/blog/tech/php_with_jquery_comet

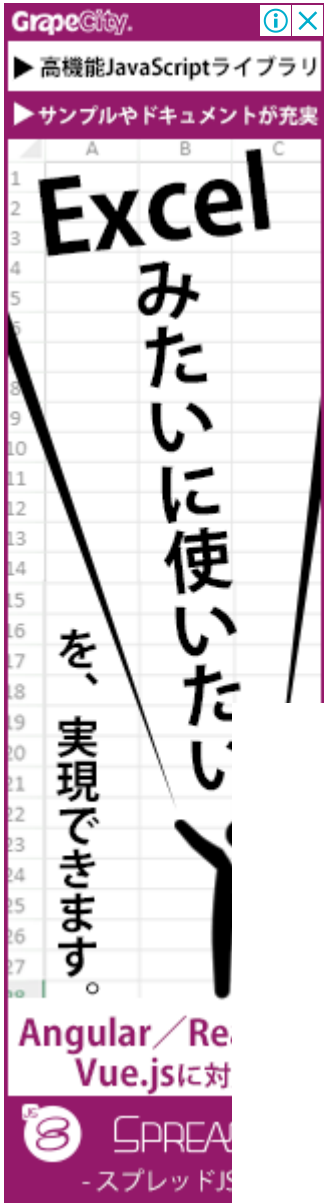


共有:



Posted in [PHP](#)

作成者 : [いまよし](#)



ブログ内検索

検索 ...

検索

最近の投稿

ソッコーでGoogle Apps Script（GAS）をソッコーにする方法

craftopiaやってました

洗濯機

京都観光してきました(嵐山編)

スキレットでローストビーフ

カレンダー

月	火	水	木	金	土	日
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

2016年6月

« 5月 7月 »

アーカイブ

- 2020年10月
- 2020年9月
- 2020年8月
- 2020年7月
- 2020年6月
- 2020年5月
- 2020年4月
- 2020年3月
- 2020年2月
- 2020年1月
- 2019年12月
- 2019年11月
- 2019年10月
- 2019年9月
- 2019年8月
- 2019年7月
- 2019年6月
- 2019年5月
- 2019年4月
- 2019年3月
- 2019年2月
- 2019年1月
- 2018年12月
- 2018年11月
- 2018年10月
- 2018年9月
- 2018年8月
- 2018年7月
- 2018年6月
- 2018年5月
- 2018年4月
- 2018年3月
- 2018年2月
- 2018年1月
- 2017年12月
- 2017年11月
- 2017年10月
- 2017年9月
- 2017年8月
- 2017年7月

2017年6月	2017年5月
2017年4月	2017年3月
2017年2月	2017年1月
2016年12月	2016年11月
2016年10月	2016年9月
2016年8月	2016年7月
2016年6月	2016年5月

カテゴリー

.net	Active Directory
Android	Angular
ASP.net	Atom
AWS	C#
CakePHP2	CakePHP3
EC-CUBE2	EC-CUBE3
Git	Google Apps Script
HTML/CSS	Illustrator
iOS	Java
javascript	jquery
jquery-ui	Kotlin
Laravel5	LINE
Linux	MVC
Node.js	Objective-C
PHP	PHP-CS-Fixer
Python	Selenium3
Spring Framework	SQL
swift4	TypeScript
Vagrant	VS Code
Xamarin	アニメ
イベント	インフラ
キャリアアップ	ゲーム
スタンプ	タイピング
デザイン	ニュース

プログラム全般	会社
勉強会	数学
旅行	日常生活
映画	暗号
未分類	機械学習
正規表現	病院
趣味	部活動
開発Tips	飲食店

▶ [ホーム](#) ▶ [会社概要](#) ▶ [ソリューション](#) ▶ [開発実績](#) ▶ [お問い合わせ](#) ▶ [プライバシーポリシー](#)
▶ [セキュリティポリシー](#) ▶ [社員ブログ](#) ▶ [お知らせ](#) ▶ [採用情報](#)

📍 〒556-0011 大阪市浪速区難波中1-3-18 DAIICHIビル5F

☎ 06-4394-8075

📅 06-4394-8076



Copyright © 2016 REGREX Co.,Ltd.