PerlのCGI.pmとJSON.pmでJSONデータを受信・送信

2017/03/28
受信サイズ制限は入れようぜとの事なので追加。

2017/03/24
自分用メモ。正確性の保証なし。

サーバサイド

Webサーバ

ドメインAに配置したコンテンツをロードしたブラウザからJavaScriptを使って別のドメインBにあるコンテンツへのアクセスはできない。
CORS(Cross-Origin Resource Sharing)という制約に引っかかる。
ドメインAにプロキシーになるプログラムを配置するなどしてこれを回避する方法もあるがCookieなどを必要とする場合うまく動作しない。

ドメインBからのレスポンスヘッダに「うちのコンテンツ使用をドメインAに許可するよ」というお許しの印を追加して貰う必要がある。ブラウザのJavaScriptはそれを確認して初めて動作する。

https://enable-cors.org/server.html

以下はApacheの.htaccessに記述した例。

```
Header set Access-Control-Allow-Origin "*"
Header set Access-Control-Allow-Headers "Content-Type"
```

ブラウザのJavaScriptからの利用を考えないならこれはやらなくても構わないけど、やれた方がいろいろ応用できるだろうから、余裕があれば適用しておく。

レスポンスヘッダ Access-Control-Allow-Origin でリクエストを出した側にコンテンツの使用を許可する。
“*”はすべてのドメインからのリクエストに許可する、の意。
こちらで試した限りではこのヘッダだけでは駄目で、レスポンスヘッダ Access-Control-Allow-Headers で “Content-Type” の利用許可を出す必要があった。

処理コード説明

Content-Type: application/json, charset=utf-8 なコンテンツを受送信する例。
Perlコードはeuc-jpなエンコードで記述している。

若様に「eval使うの怖いすなぁ」と言われたけどtry ... catch の標準的なPerl表現は eval なのです。evalで囲む範囲を狭めた方がよいのは確かな話。このサンプルぐらいでしょう許されそうなのは。

sitteru.pl

```
use CGI;
use JSON;
use Encode;

$CGI::POST_MAX = 1024 * 5; # 5KB

my %reply = ( decode('euc-jp',"リンゴ")       =>"apple",
              decode('euc-jp',"ミカン")       =>"orange",
              decode('euc-jp',"バナナ")       =>"banana",
              decode('euc-jp',"パイナップル") =>"pineapple",
              decode('euc-jp',"スイカ")       =>"watermelon" );
my $fmt1  = decode('euc-jp', "%s ? 知らない名前ですね");
my $fmt2  = decode('euc-jp', "それは %s の事ですね！");
my $cgi   = CGI->new();
my $json  = JSON->new->utf8(0);
my $ans;

##
## "POSTDATA"はPOSTメソッド使用時のBODY全体を指しています。
## "PUTDATA"はPUTメソッド使用時のBODY全体です。
my $postdata = $cgi->param("POSTDATA");

  eval {
    ## JSONデータをパースしてPerlオブジェクトを取り出します。
    my $data = $json->decode($postdata);

    ## $dataのプロパティ q に使用する値が入ってます。
    my $q    = decode('utf-8', $data->{q});

    ## $qの値から送り返すデータをPerlオブジェクトで作ります
    my $message = sprintf($fmt1, $q);
    $message = sprintf($fmt2, $reply{$q})  if exists($reply{$q});
    $ans     = {
                 message => $message,
                 status  => 200
               };
  };
  if ($@)
  {
    $ans = {
             message => 'send data broken.',
             submessage => $@ ,
             status  => 500
           };
  }

  ## JSON形式のレスポンスを返すためのレスポンスヘッダ
  print $cgi->header({type=>'application/json', charset=>'utf-8'});

  ## PerlオブジェクトをJSON形式にエンコードしています
```

```
print $json->encode($ans);
```

その他

$jsonを以下、

```
my $json  = JSON->new->utf8(1);
```

の定義とした場合、プロパティ q は以下で取り出す。

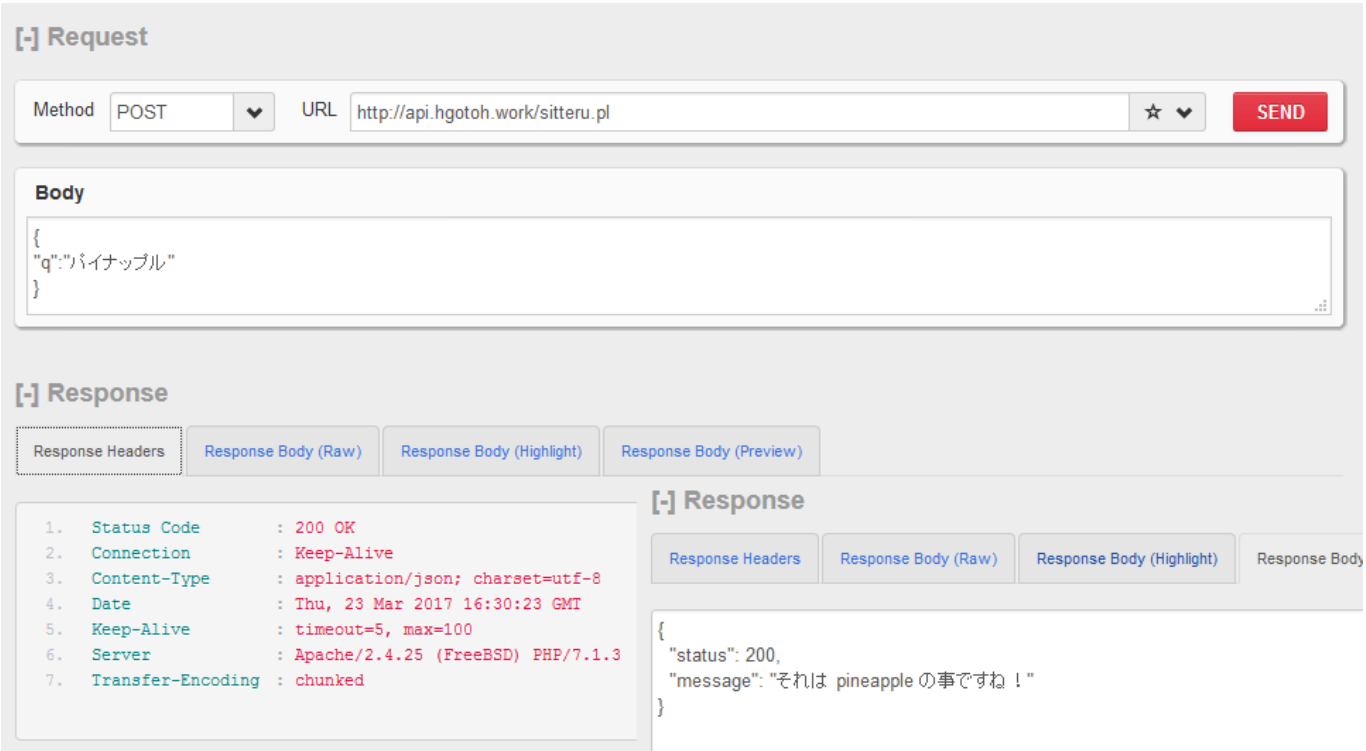
```
my $q    = $data->{q};
```

utf8(1)の時はパース時にUTF-8でdecode()が行われPerl内部形式になっているから。

クライアントサイド

RESTクライアントで確認

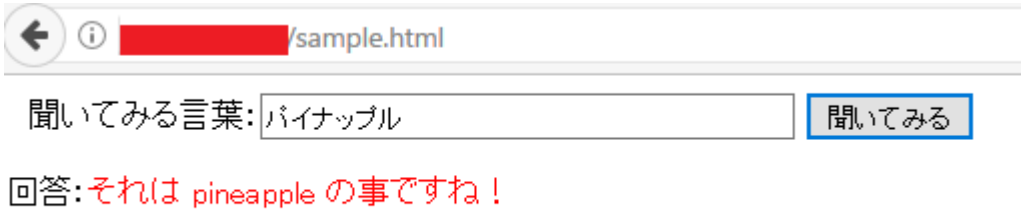
以下のJSONドキュメントのプロパティqに“ パイナップル ”を指定してPOSTメソッドで送り出した。ハッシュ %reply に存在するキーだったため、対応する値が取り出され“pineapple”のレスポンスがJSONドキュメントで返されてきた。



HTML+JavaScriptで確認

JQueryの$ajaxオブジェクトでURLを呼び出し。なおWebサーバに配置しないとAjax関係のコードは機

能しない。



HTMLコード。

sample.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>API http://api.hgotoh.work/sitteru.pl の使用例</title>
    <meta charset="utf-8"/>
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js">
</script>
    <script type="text/javascript" src="sample.js"></script>
  </head>
  <body>
    <form>
      <div>
    聞いてみる言葉：<input type="text" id="q" placeholder="カタカナで果物の名
前を入れてね" size=40/>
        <button type="button" id="submit">聞いてみる</button>
      </div>
      <input type="text" id="dummy" style="display:none;"/>
      <br/>
    </form>
    <div>回答：<span id="answer" style="color: red"></span></div>
  </body>
</html>
```

JavaScriptコード。

sample.js

```
      $(function(){
        $('#submit').click(function (){
          var data = { q: $('#q').val() };
          $.ajax({
            url:         'http://api.hgotoh.work/sitteru.pl',
            type:        'POST',
            contentType: 'application/json',
            data:        JSON.stringify(data)
```

```
          }).done(function(data){
               $('#answer').text(data.message);
          }).fail(function(data){
               $('#answer').text('通信障害かも…');
               alert(data);
          });
        });
      });
```

技術資料, Perl, CGI, JSON, CORS, Ajax, JQuery, 関数