



STMIK EL RAHMA

Headers

Wahyu Widodo, S.Kom.,M.Kom



Point Pembahasan Pertemuan 6 :

- Informasi Penting Request Headers dan Response Headers
- HTTP Basic Authentication
- Studi Kasus Integrasi Pembayaran Stripe





Request Headers

```
GET / HTTP/1.1
Host: www.lornajane.net
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.19 (KHTML, like
Gecko) Chrome/25.0.1323.1 Safari/537.19
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-GB,en-US;q=0.8,en;q=0.6
Accept-Charset: ISO-8859-1,utf-8;q=0.7,+;q=0.3
```

- **Method request** yang digunakan (misal : GET)
- **Accept** : merupakan jenis batasan data yang boleh dikirim melakukan request.
- **User-Agent** : browser yang anda gunakan.



Response Headers

```
HTTP/1.1 200 OK
Server: Apache/2.2.14 (Ubuntu)
X-Powered-By: PHP/5.3.2-1ubuntu4.11
X-Pingback: http://www.lornajane.net/xmlrpc.php
Last-Modified: Thu, 06 Dec 2012 14:46:05 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Content-Type: text/html; charset=UTF-8
Content-Length: 25279
Date: Thu, 06 Dec 2012 14:46:05 GMT
X-Varnish: 2051611642
Age: 0
Via: 1.1 varnish
Connection: keep-alive
```

- **Status Response** : Nasib berhasil tidaknya permintaan. 200 OK (berhasil)
- **Accept-Type** : batasan data yang diterima Anda setelah berhasil request.
- **Cookies** : jika dibutuhkan untuk create cookies (opsional)
- **Authorization** : jika dibutuhkan otoritas untuk mengakses (opsional).



User Agent itu apa?

User Agent adalah *script* yang dikirimkan oleh *web browser* ke web server yang kita tuju, atau ke setiap situs yang kita kunjungi, jadi setiap situs yang kita kunjungi dapat mengetahui browser dan sistem operasi yang kita gunakan sehingga konten dapat disesuaikan dengan jenis sistem operasi kita.



Info apa aja yang didapat?

Sistem Operasi yang kita gunakan (*dan kadang nama distro jika kita menggunakan Linux*), *engine browser*, *engine browser version*, dan nama browser.

(Lihat gambar request header)



Buat script php berikut untuk menampilkan user agent

```
1 <?php
2     echo "This request made by: ".
3     filter_var($_SERVER['HTTP_USER_AGENT'], FILTER_SANITIZE_STRING);
4 ?>
```

Kemudian jalankan via browser





Membuat batasan data request

Secara default, jika data yang bisa dikirim :

This request made by:

*text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,
/;q=0.8.*

dihasilkan dari perintah :


```
1 <?php
2     echo "This request made by: ".
3     filter_var($_SERVER['HTTP_ACCEPT'], FILTER_SANITIZE_STRING);
4     ?>
```

Tapi kita juga bisa menentukan sendiri, misalnya boleh mengirim ke server hanya dengan format json.

Membuat batasan data request dan response

Dengan menggunakan CURLOPT_HTTPHEADER, dapat memodifikasi Accept dan Accept-Type.

```
$options = [  
    CURLOPT_CUSTOMREQUEST => "GET",  
    CURLOPT_HTTPHEADER => [  
        'Content-Type: application/json',  
        'Accept: application/json'  
    ],  
    CURLOPT_RETURNTRANSFER => 1  
];  
  
$ch = curl_init( $url ); // Inisialisasi Curl  
curl_setopt_array( $ch, $options ); // Set Opsi  
$content = curl_exec( $ch ); // Eksekusi Curl  
curl_close( $ch ); // Stop atau tutup script
```






HTTP Basic Authentication

- Headers menyediakan informasi untuk memungkinkan aplikasi mengidentifikasi si pengguna yang melakukan request.
- Caranya dengan mengirimkan teks Base64 encoded username dan password pada HTTP header Authorization. Username dan password tersebut selalu dikirimkan untuk setiap HTTP request untuk proses authorization.



Contoh Request API yang diproteksi

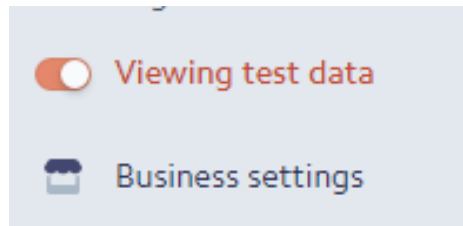
```
$options = [  
  CURLOPT_CUSTOMREQUEST => "GET",  
  CURLOPT_HTTPHEADER => [  
    'Content-Type: application/json',  
    'Accept: application/json',  
    'Authorization: Basic ' . base64_encode($server_key . ':')  
  ],  
  CURLOPT_RETURNTRANSFER => 1  
];  
  
$ch = curl_init( $url ); // Inisialisasi Curl  
curl_setopt_array( $ch, $options ); // Set Opsi  
$content = curl_exec( $ch ); // Eksekusi Curl  
curl_close( $ch ); // Stop atau tutup script
```





Studi Kasus Integrasi Pembayaran Stripe

- Belum punya akun? Silahkan registrasi dulu <https://stripe.com/>
- Jika sudah, silahkan login. Anda akan mendapat tampilan dashboard. <https://dashboard.stripe.com>
- Silahkan switch view test data, pada menu dashboard bawah.





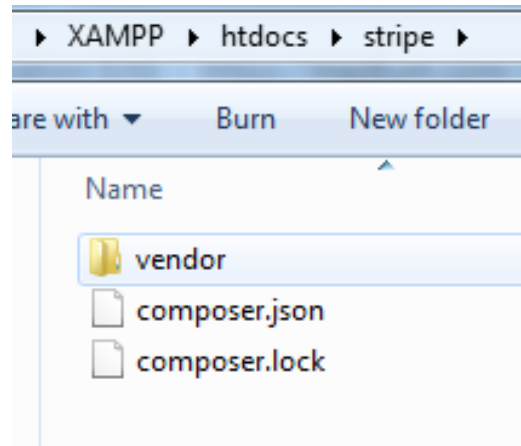
Petunjuk Instalasi

- Jika komputer sudah terinstal composer dan terhubung ke internet, folder baru dengan nama stripe di c:\xampp\htdocs\stripe
- Dengan command line masuk pada direktori stripe ketik kode berikut :
composer require stripe/stripe-php
- Tunggu sampai proses download libraries stripe selesai



Petunjuk Instalasi (1)

- Jika berhasil, akan muncul 1 folder dan 2 file seperti gambar berikut ini :



- Alternatif lain, silahkan gunakan folder vendor yang sudah disediakan



Studi Kasus Akses API Stripe

- Buka dokumentasinya :
 - versi online : <https://stripe.com/docs/api> -
 - Lihat pada bagian *Register Card Status* -- ---
- Coba lihat Method Create a Charges
- <https://stripe.com/docs/api/charges/create>



- Buat file dengan nama Stripegateway.php, simpan sejajar dengan folder vendor, kemudian ketik kode berikut ini :

```
1 <?php
2
3 include("../vendor/autoload.php");
4
5 class Stripegateway {
6
7     public function __construct(){
8         $stripe = array(
9             "secret_key" => "lihat dashboard punyamu",
10            "public_key" => "lihat dashboard punyamu"
11        );
12        \Stripe\Stripe::setApiKey($stripe["secret_key"]);
13    }
14
```




- Lanjutan :

```
15 public function checkout($data){
16     $message = "";
17     try{
18         $mycard = array('number' => $data['number'],
19                         'exp_month' => $data['exp_month'],
20                         'exp_year' => $data['exp_year']);
21         $charge = \Stripe\Charge::create(array('card'=>$mycard,
22                                                'amount'=>$data['amount'],
23                                                'currency'=>'usd'));
24         $message = $charge->status;
25     }catch (Exception $e){
26         $message = $e->getMessage();
27     }
28     return $message;
29 }
30
31 }
```



- Buat file index.php untuk desain form input credit card

```
<form action="pay.php" method="post" class="form-horizontal">
  <div class="form-group">
    <label class="control-label col-md-3">Card Number</label>
    <input type="text" size="20" autocomplete="off" name="cardnumber" value="4242424242424242" />
  </div>
  <div class="form-group">
    <label class="control-label col-md-3">Expiration (MM/YYYY)</label>
    <input type="text" size="2" name="expirymonth" value="8" />
    <span> / </span>
    <input type="text" size="4" name="expiryyear" value="2019" />
  </div>
  <div class="form-group">
    <label class="control-label col-md-3">Amount</label>
    <input type="text" size="4" name="amount" />
  </div>
  <button type="submit" name="btnsubmit" class="btn btn-primary">Submit Payment</button>
</form>
```



- Buat file pay.php untuk submit data dari form

```
1 <?php
2 include('Stripegateway.php');
3 $myStripe = new Stripegateway();
4 if(isset($_POST['btnsubmit'])){
5     $data= array('number' => $_POST['cardnumber'],
6                 'exp_month' => $_POST['expirymonth'],
7                 'exp_year' => $_POST['expiryyear'],
8                 'amount'=>($_POST['amount']*100));
9     $result= $myStripe->checkout($data);
10    echo "<pre>";print_r($result);
11 }
12 ?>
```



STMIK EL RAHMA

