Membuat JWT Token di Code Igniter

Buat file jwt.php di folder application/config/

------

<?php

// application/config/jwt.php

defined('BASEPATH') OR exit('No direct script access allowed');

// Store your secret key here

// Make sure you use better, long, more random key than this

$config['jwt\_key'] = '#iL0v3YoU#';

$config['token\_timeout'] = 1;

-----

Tambahkan baris kode berikut ini di file **application/config/autoload.php**

----

**$autoload['config'] = array('jwt');**

----

Buat file **jwt\_helper.php** pada folder **application/helpers** (isinya copy saja dari pyoyek yg sudah ada)

Buat file **authorization\_helper.php** pada folder **application/helpers** (isinya copy saja dari pyoyek yg sudah ada)

------

<?php

class AUTHORIZATION

{

public static function validateTimestamp($token)

{

$CI =& get\_instance();

$token = self::validateToken($token);

if ($token != false && (now() - $token->timestamp < ($CI->config->item('token\_timeout') \* 60))) {

return $token;

}

return false;

}

public static function validateToken($token)

{

$CI =& get\_instance();

return JWT::decode($token, $CI->config->item('jwt\_key'));

}

public static function generateToken($data)

{

$CI =& get\_instance();

return JWT::encode($data, $CI->config->item('jwt\_key'));

}

}

------

Buat file **REST\_Controller.php** pada folder **application/libraries** (filenya copy saja dari project yg ada)

Buat file **Format.php** pada folder **application/libraries** (filenya copy saja dari project yg ada)

Buat file **rest.php** pada folder **application/config** (filenya copy saja dari project yg ada)

Buat file **Rest\_server.php** pada folder **application/controller**

------

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Rest\_server extends CI\_Controller {

public function index()

{

$this->load->helper('url');

$this->load->view('rest\_server');

}

}

-----

Buat file rest\_controller\_lang.php simpan pada folder **application/language/english/**

**-----**

**<?php**

**/\***

**\* English language**

**\*/**

**$lang['text\_rest\_invalid\_api\_key'] = 'Invalid API key %s'; // %s is the REST API key**

**$lang['text\_rest\_invalid\_credentials'] = 'Invalid credentials';**

**$lang['text\_rest\_ip\_denied'] = 'IP denied';**

**$lang['text\_rest\_ip\_unauthorized'] = 'IP unauthorized';**

**$lang['text\_rest\_unauthorized'] = 'Unauthorized';**

**$lang['text\_rest\_ajax\_only'] = 'Only AJAX requests are allowed';**

**$lang['text\_rest\_api\_key\_unauthorized'] = 'This API key does not have access to the requested controller';**

**$lang['text\_rest\_api\_key\_permissions'] = 'This API key does not have enough permissions';**

**$lang['text\_rest\_api\_key\_time\_limit'] = 'This API key has reached the time limit for this method';**

**$lang['text\_rest\_unknown\_method'] = 'Unknown method';**

**$lang['text\_rest\_unsupported'] = 'Unsupported protocol';**

**-----**

**Uji coba**

Buat file Tes\_token.php di folder application/controller

-----

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . 'libraries/REST\_Controller.php';

require APPPATH . 'libraries/Format.php';

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");

class Tes\_token extends REST\_Controller {

public function \_\_construct() {

parent::\_\_construct();

// Load these helper to create JWT tokens

$this->load->helper(['jwt', 'authorization']);

}

public function hello\_get()

{

$tokenData = 'Hello World!';

// Create a token

$token = AUTHORIZATION::generateToken($tokenData);

// Set HTTP status code

$status = parent::HTTP\_OK;

// Prepare the response

$response = ['status' => $status, 'token' => $token];

// REST\_Controller provide this method to send responses

$this->response($response, $status);

}

}

Cara aksesnya -> <http://192.168.1.5/iapolbat/tes_token/hello>

Sekarang coba tes login statis

Tambahkan method berikut ini dalam file Tes\_token.pphp diatas

---

**public function login\_post()**

**{**

**// Have dummy user details to check user credentials**

**// send via postman**

**$dummy\_user = [**

**'username' => 'fendi',**

**'password' => '123'**

**];**

**// Extract user data from POST request**

**$username = $this->post('username');**

**$password = $this->post('password');**

**// Check if valid user**

**if ($username === $dummy\_user['username'] && $password === $dummy\_user['password']) {**

**// Create a token from the user data and send it as reponse**

**$token = AUTHORIZATION::generateToken(['username' => $dummy\_user['username']]);**

**// Prepare the response**

**$status = parent::HTTP\_OK;**

**$response = ['status' => $status, 'token' => $token];**

**$this->response($response, $status);**

**}**

**else {**

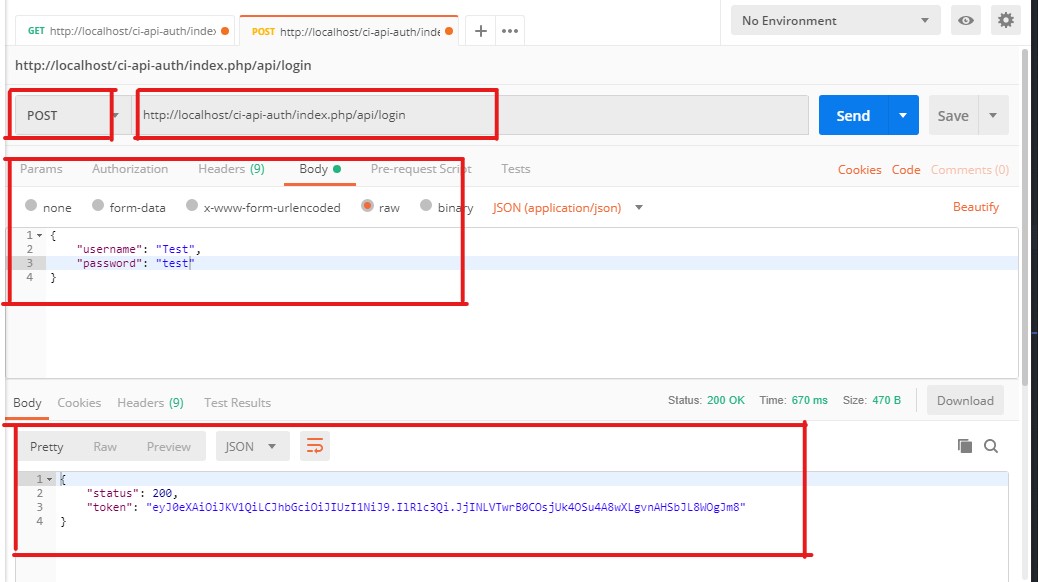
**$this->response(['msg' => 'Invalid username or password!'], parent::HTTP\_NOT\_FOUND);**

**}**

**}**

-----

Uji coba dengan postman



http://192.168.1.5/iapolbat/tes\_token/login

Sekarang uji coba akses data dengan ketentuan memiliki token hasil login

Tambahkan method verifikasi pada file Tes\_token.php

----

private function verify\_request()

{

// Get all the headers

$headers = $this->input->request\_headers();

// Extract the token

$token = $headers['Authorization'];

// Use try-catch

// JWT library throws exception if the token is not valid

try {

// Validate the token

// Successfull validation will return the decoded user data else returns false

$data = AUTHORIZATION::validateToken($token);

if ($data === false) {

$status = parent::HTTP\_UNAUTHORIZED;

$response = ['status' => $status, 'msg' => 'Unauthorized Access!'];

$this->response($response, $status);

exit();

} else {

return $data;

}

} catch (Exception $e) {

// Token is invalid

// Send the unathorized access message

$status = parent::HTTP\_UNAUTHORIZED;

$response = ['status' => $status, 'msg' => 'Unauthorized Access! '];

$this->response($response, $status);

}

}

public function get\_me\_data\_post()

{

// Call the verification method and store the return value in the variable

$data = $this->verify\_request();

// Send the return data as reponse

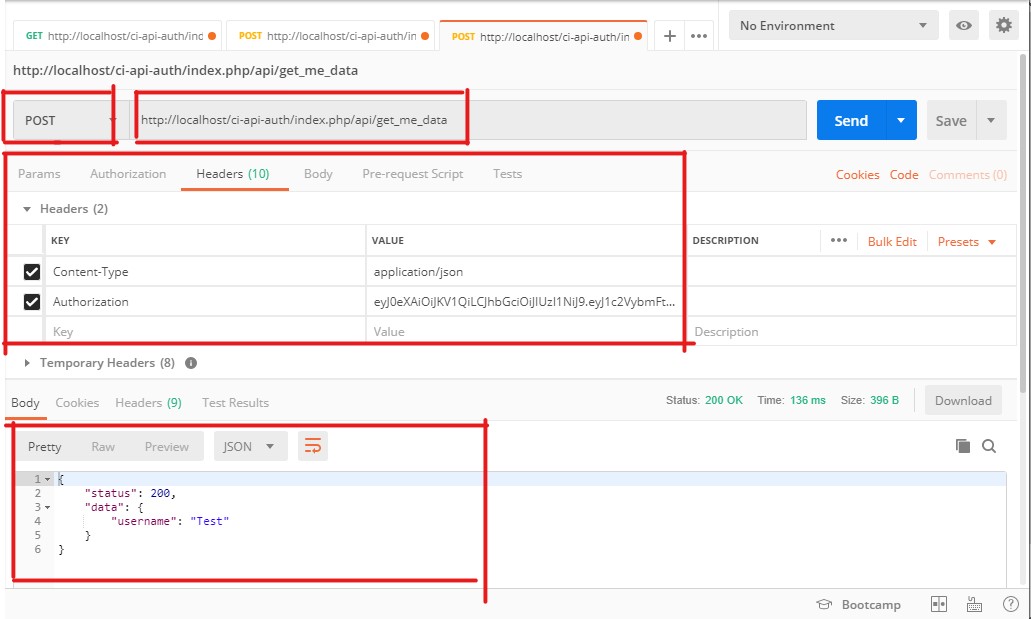
$status = parent::HTTP\_OK;

$response = ['status' => $status, 'data' => $data];

$this->response($response, $status);

}

------



http://192.168.1.5/iapolbat/tes\_token/get\_me\_data