

Implementasi JWT menggunakan library PHP-JWT (studi kasus: Wisata)

Rancangan endpoint API Restfull untuk JWT pada studi kasus Wisata :

No	URL API endpoint	HTTP Verb/Method	Keterangan
1	http://localhost/json/server/auth/login	POST	Layanan untuk <i>generate</i> token menggunakan username dan password login
	Jika token di tabel “kosong” , maka: <ul style="list-style-type: none">• generate JWT token Jika token di tabel “expired” , maka: <ul style="list-style-type: none">• generate JWT token baru Jika token di tabel “tidak expired” , maka: <ul style="list-style-type: none">• gunakan JWT token yang ada di tabel		
2	http://localhost/json/server/auth/token	POST	Layanan untuk mengambil token sesuai username dan password login
	Jika token di tabel “kosong” , maka: <ul style="list-style-type: none">• return status “empty” Jika token di tabel “expired” , maka: <ul style="list-style-type: none">• return status “expired” Jika token di tabel “tidak expired” , maka: <ul style="list-style-type: none">• return token		

Menggunakan project “Wisata” pada praktikum dan implementasi API Design pada pertemuan sebelumnya, kita akan melakukan beberapa modifikasi dan penambahan file dan folder, silahkan download file pendukung pada elearning dan lakukan langkah berikut:

```

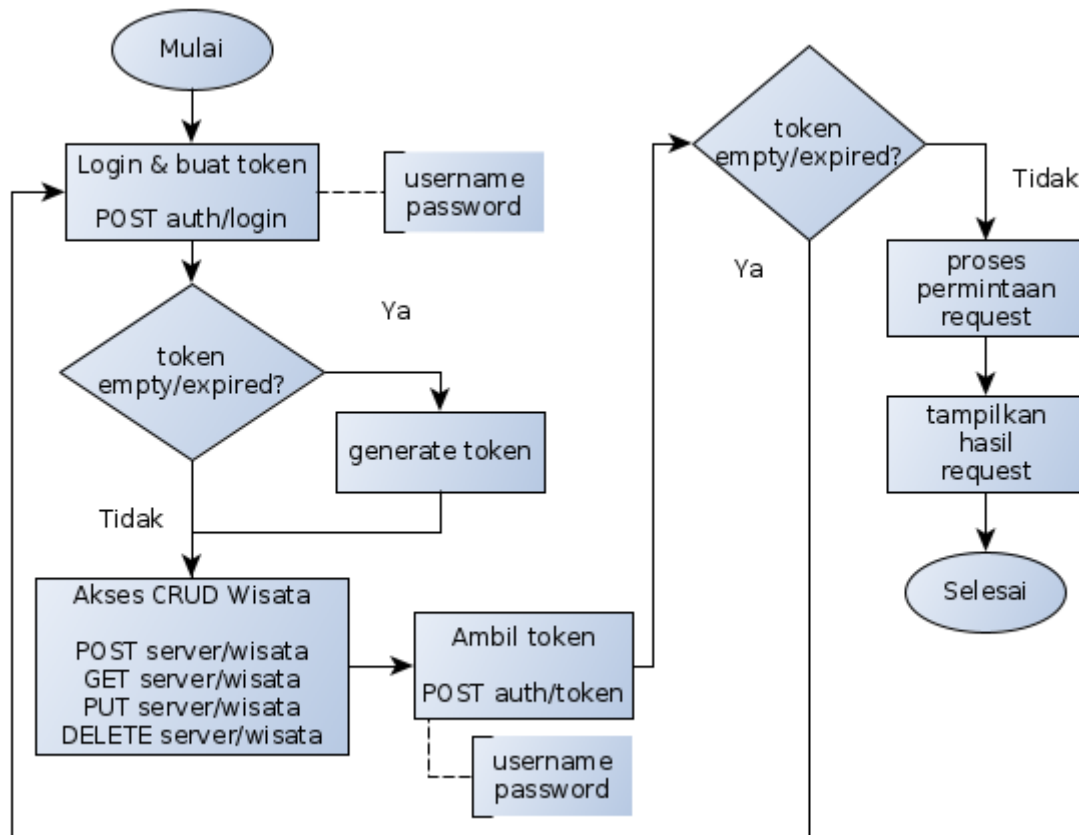
.
├── config
│   └── config.php
├── controllers
│   ├── Auth.php
│   └── Wisata.php
├── core
│   └── MY_Controller.php
├── libraries
│   ├── BeforeValidException.php
│   ├── ExpiredException.php
│   ├── JWT.php
│   └── SignatureInvalidException.php
├── models
│   ├── User_m.php
│   └── Wisata_m.php
└── tabel_user
    └── users.sql

```

6 directories, 11 files

No	Nama File	Langkah	Folder	Keterangan
1	Users.sql	Import ke database json	-	Tabel user untuk login
2	BeforeValidException.php ExpiredException.php SignatureInvalidException.php JWT.php	Copy/pindahkan ke folder "application/libraries/"	Json/server/application/libraries	File libraries JWT Token
3	MY_Controller.php	Copy/pindahkan ke folder "application/core/"	Json/server/application/core	File controller anak RestController
4	Auth.php Wisata.php	Copy/pindahkan ke folder "application/controllers"	Json/server/application/controllers	Auth.php → controller token Wisata.php → update controller wisata
5	User_m.php Wisata_m.php	Copy/pindahkan ke folder "application/models"	Json/server/application/models	User_m.php → model tabel user Wisata_m.php → update model wisata
6	config.php	Copy/pindahkan ke folder "application/config/" tambah config: // kunci untuk JWT \$config['thekey'] pada baris paling akhir	Json/server/application/config	Update file config, tambah baris \$config['thekey'] untuk kunci rahasia enkripsi dan deskripsi JWT Token

Alur Akses CRUD Wisata dengan Autorisasi Token (JWT)



Struktur file Model dan Class untuk menyediakan endpoint TOKEN dan update file model dan controller CRUD Wisata

Nama file Class: User_m.php (Model)

Fungsi : query dan akses ke tabel user

Extends: CI_Model

No	Komponen class	Parameter method	Nilai return	Keterangan
1	class User_m extends CI_Model{ }	-	-	Deklarasi class model
2	function get_user(\$q = NULL) { }	Keyword pencarian	Data user	Ambil data user
3	function get_token(\$q = NULL){ }	Keyword pencarian	Data token	Ambil data token
4	function update_token(\$id, \$data) { }	Id user dan token	Status true/false	Update token

Nama file Class: MY_Controller.php (Core Controller)

Fungsi : core controller untuk API REST Token JWT

Extends: RestController → library class REST

No	Komponen class	HTTP Method	Keterangan
	require_once APPPATH . 'libraries/RestController.php'; require_once APPPATH . 'libraries/Format.php'; require_once APPPATH . 'libraries/JWT.php'; require_once APPPATH . 'libraries/BeforeValidException.php'; require_once APPPATH . 'libraries/ExpiredException.php'; require_once APPPATH . 'libraries/SignatureInvalidException.php';	-	Load file librari REST dan JWT
1	use chriskacerguis\RestServer\RestController; use Firebase\JWT\JWT;	-	Menggunakan namespace library milik class REST dan class JWT
2	class MY_Controller extends RestController {}	-	Deklarasi class MY_Controller untuk JWT Token extends RestController
3	function __construct() {}	-	Construct
4	public function auth() {}	-	Membaca header request “Authorization”, jika valid proses request dilanjutkan, jika invalid proses request ditolak

Nama file Class: Auth.php (Controller)

Fungsi : controller API REST Token JWT

Extends: MY_Controller → library class REST modifikasi ‘application/core’

No	Komponen class	HTTP Method	Keterangan
1	use chriskacerguis\RestServer\RestController; use Firebase\JWT\JWT;	-	Menggunakan namespace library milik class REST dan class JWT
2	class Auth extends MY_Controller {}	-	Deklarasi class Auth untuk API REST Token extends MY_Controller
3	function __construct() {}	-	Construct, load model User_m dan load kunci rahasia token
4	public function login_post() {}	POST	Untuk melayani request login dengan POST
5	public function token_post() {}	POST	Untuk melayani request ambil token dengan POST
6	private function is_exp(\$token) {}	-	Untuk melakukan cek token expired atau tidak
7	private function generate_token(\$data) {}	-	Untuk mengenerate token

Beberapa modifikasi file Controller “Wisata.php”

No	Sebelum	Sesudah
1	Perubahan pada induk class Wisata, dari RestController → MY_Controller	
	<pre> 1 <?php 2 3 if (!defined('BASEPATH')) 4 exit('No direct script access allowed'); 5 6 require APPPATH . 'libraries/RestController.php'; 7 require APPPATH . 'libraries/Format.php'; 8 use chriskacerguis\RestServer\RestController; 9 10 class Wisata extends RestController { 11 </pre>	<pre> 1 <?php 2 3 if (!defined('BASEPATH')) 4 exit('No direct script access allowed'); 5 6 //require_once APPPATH . 'libraries/RestController.php'; 7 //require_once APPPATH . 'libraries/Format.php'; 8 //use chriskacerguis\RestServer\RestController; 9 10 //class Wisata extends RestController { 11 class Wisata extends MY_Controller { 12 </pre>
2	Penambahan method \$this → auth() pada _construct	
	<pre> 12 function __construct() 13 { 14 parent::__construct(); 15 \$this->load->model('Wisata_m'); 16 } </pre>	<pre> 13 function __construct() 14 { 15 parent::__construct(); 16 \$this->load->model('Wisata_m'); 17 \$this->auth(); 18 } </pre>
3	Penambahan skrip tangkap variabel hasil kirim dengan GET	
	<pre> 18 function index_get(\$id = NULL) { 19 // \$id = \$this->get('id'); 20 if (\$id == '') { 21 \$wisata = \$this->Wisata_m->getWisata(); 22 } else { 23 \$wisata = \$this->Wisata_m->getWisata(\$id); 24 } 25 \$this->response(\$wisata, 200); 26 } </pre>	<pre> 20 function index_get(\$id = NULL) { 21 // \$id = \$this->get('id'); 22 // tambahan 23 if (\$id == NULL) \$id = \$this->get('id'); 24 // akhir tambahan 25 if (\$id == '') { 26 \$wisata = \$this->Wisata_m->getWisata(); 27 } else { 28 \$wisata = \$this->Wisata_m->getWisata(\$id); 29 } 30 \$this->response(\$wisata, 200); 31 } </pre>
4	Penambahan skrip tangkap variabel hasil kirim dengan GET dan perbaiki parameter fungsi, set dengan nilai NULL	
	<pre> 66 function read_limit_get(\$limit, \$start, \$q = NULL) { 67 // \$limit = \$this->get('limit'); 68 // \$start = \$this->get('start'); 69 // \$q = \$this->get('q'); 70 \$wisata = \$this->Wisata_m->getWisata_limit(\$limit, \$ 71 start, \$q); 72 \$this->response(\$wisata, 200); 73 } </pre>	<pre> 73 function read_limit_get(\$limit = NULL, \$start = NULL, \$q = 74 NULL) { 75 // \$limit = \$this->get('limit'); 76 // \$start = \$this->get('start'); 77 // \$q = \$this->get('q'); 78 // tambahan 79 if (\$limit == NULL) \$limit = \$this->get('limit'); 80 if (\$start == NULL) \$start = \$this->get('start'); 81 if (\$q == NULL) \$q = \$this->get('q'); 82 // akhir tambahan 83 \$wisata = \$this->Wisata_m- 84 >getWisata_limit(\$limit, \$start, \$q); 85 \$this->response(\$wisata, 200); 86 } </pre>

	<pre> 74 function total_rows_get(\$q = NULL) { 75 //\$q = \$this->get('q'); 76 if (\$q == '') { 77 \$total_rows = \$this->Wisata_m-> getWisata_total_rows(); 78 } else { 79 \$total_rows = \$this->Wisata_m-> getWisata_total_rows(\$q); 80 } 81 \$this->response(array('total_rows' => \$total_rows 82), 200); 83 } </pre>	<pre> 86 function total_rows_get(\$q = NULL) { 87 //\$q = \$this->get('q'); 88 // tambahan 89 if (\$q == NULL) \$q = \$this->get('q'); 90 // akhir tambahan 91 if (\$q == '') { 92 \$total_rows = \$this->Wisata_m-> >getWisata_total_rows(); 93 } else { 94 \$total_rows = \$this->Wisata_m-> >getWisata_total_rows(\$q); 95 } 96 \$this->response(array('total_rows' => \$total_rows), 97 200); 98 } </pre>
5	Penambahan skrip tangkap variabel hasil kirim dengan DELETE dan perbaiki parameter fungsi, set dengan nilai NULL	
	<pre> 55 function index_delete(\$id) { 56 //\$id = \$this->delete('id'); 57 \$delete = \$this->Wisata_m->deleteWisata(\$id); 58 59 if (\$delete) { 60 \$this->response(array('status' => 'success'), 61 201); 62 } else { 63 \$this->response(array('status' => 'fail'), 64 502); 65 } 66 } </pre>	<pre> 60 function index_delete(\$id = NULL) { 61 //\$id = \$this->delete('id'); 62 // tambahan 63 if (\$id == NULL) \$id = \$this->delete('id'); 64 // akhir tambahan 65 \$delete = \$this->Wisata_m->deleteWisata(\$id); 66 if (\$delete) { 67 \$this->response(array('status' => 'success'), 201); 68 } else { 69 \$this->response(array('status' => 'fail'), 502); 70 } 71 } </pre>
6	Penambahan status respon request jika sukses pada layanan INSERT dan UPDATE	
	<pre> 39 if (\$insert) { 40 \$this->response(array('status' => 'success', 'data' => \$data), 200); 41 } else { 42 \$this->response(array('status' => 'fail'), 502); 43 } </pre>	
	<pre> 53 if (\$update) { 54 \$this->response(array('status' => 'success', 'data' => \$data), 200); 55 } else { 56 \$this->response(array('status' => 'fail'), 502); 57 } </pre>	

Beberapa modifikasi file Model “Wisata_m.php”

No	Sebelum	Sesudah
1	Penambahan cek hasil query saat akan di return, jika operasi query sukses return “true”, sebaliknya return “false”	
	- public function insertWisata(\$wstData){ }	
	<pre> 62 \$this->db->query(\$wstQuery); 63 \$resultData = \$this->db->affected_rows(); 64 return \$resultData; </pre>	<pre> 62 \$this->db->query(\$wstQuery); 63 \$resultData = \$this->db->affected_rows(); 64 return \$val = (\$resultData > 0)? true : false; </pre>

- public function updateWisata(\$wstData){}		
97	<code>\$this->db->query(\$wstQuery);</code>	97 <code>\$this->db->query(\$wstQuery);</code>
98	<code>\$resultData = \$this->db->affected_rows();</code>	98 <code>\$resultData = \$this->db->affected_rows();</code>
99	<code>return \$resultData;</code>	99 <code>return \$val = (\$resultData > 0)? true : false;</code>
- public function deleteWisata(\$wstId) {}		
128	<code>\$this->db->query(\$wstQuery);</code>	128 <code>\$this->db->query(\$wstQuery);</code>
129	<code>\$resultData = \$this->db->affected_rows();</code>	129 <code>\$resultData = \$this->db->affected_rows();</code>
130	<code>return \$resultData;</code>	130 <code>return \$val = (\$resultData > 0)? true : false;</code>

Ujicoba API

<div> <div>Wisata</div> <div> <div>Token</div> <div> <div>POST mendapatkan token</div> <div>POST mengambil token</div> </div> </div> <div> <div>CRUD</div> <div> <div>GET Ambil data wisata semua</div> <div>GET Ambil data wisata sesuai id (parameter)</div> <div>GET Ambil data wisata sesuai id (URI)</div> <div>POST Tambah data wisata</div> <div>PUT Koreksi data wisata</div> <div>DEL Hapus data wisata (parameter)</div> <div>DEL Hapus data wisata (URI)</div> <div>GET Ambil total data</div> <div>GET Ambil data wisata limit (URI)</div> <div>GET Ambil data wisata limit (parameter)</div> <div>GET Ambil data wisata dengan limit dan keyword (URI)</div> <div>GET Ambil data wisata dengan limit dan keyword (para...</div> </div> </div> </div>	
---	--

Tidak memerlukan Authorization dengan token pada Header Request

Memerlukan Authorization dengan token pada Header Request

Contoh Header Request TANPA Authorization

Request Headers:

POST /json/server/auth/login HTTP/1.1

User-Agent: PostmanRuntime/7.28.0

Accept: */*

Cache-Control: no-cache

Postman-Token: 8a9903d8-9c59-47b0-8025-28ad6b5ad5e9

Host: localhost

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Content-Type: multipart/form-data; boundary=-----079370277489257955133156

Content-Length: 281

-----079370277489257955133156

Content-Disposition: form-data; name="username"

dodi

-----079370277489257955133156

Content-Disposition: form-data; name="password"

dodi123

-----079370277489257955133156--

Response Headers:

HTTP/1.1 200 OK

Date: Mon, 17 May 2021 09:35:28 GMT

Server: Apache/2.4.18 (Ubuntu)

Content-Length: 200

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Content-Type: application/json; charset=utf-8

Response Body:

```
{"status": "token not expired", "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjIiLCJ1c2VyIjoiaWoiZG9kaSI6ImIhdCI6MTYyMTI0MzA0OCwiZmxwIjojNjI0MjYxMDQ4fQ.pyIudliYZ6Xd1OGkDHb-IgTPxeZxgE-mTiFQXX_ORgU"}
```

Contoh Header Request dengan Authorization

Request Headers:

GET /json/server/wisata HTTP/1.1

Authorization: Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6IjIiLCJ1c2VyIjoiaWoiZG9kaSI6ImIhdCI6MTYyMTI0MzA0OCwiZmxwIjojNjI0MjYxMDQ4fQ.pyIudliYZ6Xd1OGkDHb-IgTPxeZxgE-mTiFQXX_ORgU

User-Agent: PostmanRuntime/7.28.0

Accept: */*

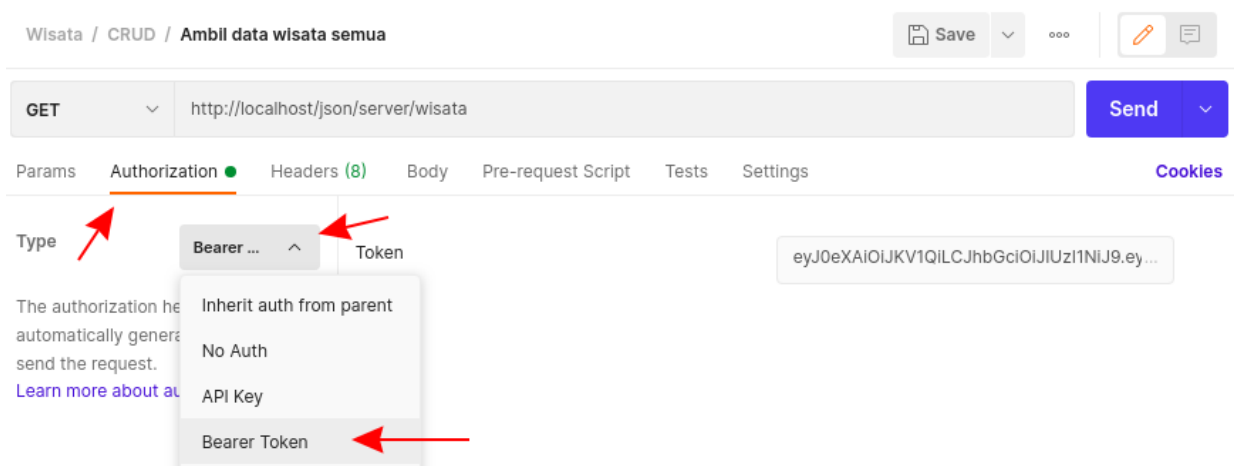
Cache-Control: no-cache

Postman-Token: cbd639e0-6d9f-4848-922d-1e54e4b0b49b

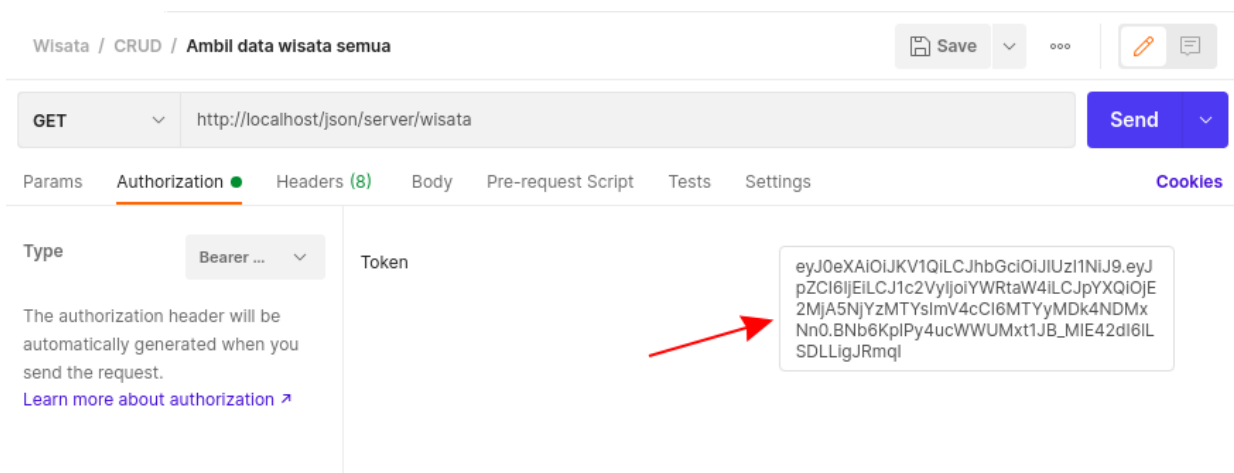
Host: localhost

Accept-Encoding: gzip, deflate, br

Connection: keep-alive



4. Paste/tempel token tersebut pada isian token



5. Kirim request dengan klik Send.

Lakukan ujicoba endpoint API diatas menggunakan aplikasi POSTMAN