

Membuat RESTful API dengan PHP dan MySQLi

Tutorial PHP kali ini mencoba untuk menjelaskan secara mudah cara membangun suatu RESTful API yang mendukung semua operasi CRUD tanpa harus menggunakan suatu framework REST. Adanya framework memudahkan banyak urusan tetapi kurang baik untuk memahami cara kerja konsep dasar dari suatu sistem. Kita dapat mendefinisikan struktur dan mengubahnya sesuai dengan keperluan, termasuk mendefinisikan aturan bagi framework REST API dan kemudahan aksesnya. Adalah tidak wajib untuk menggunakan framework untuk membangun REST API menggunakan PHP. Kita dapat membuat API hanya menggunakan kode PHP inti.

Apa beda tutorial ini dengan sebelumnya yang juga membahas pembangunan RESTful API dengan PHP? Jika sebelumnya komunikasi antara PHP dan MySQL menggunakan PDO, maka kali ini hanya menggunakan mysqli. Perlu diingat kembali bahwa PHP menyediakan dua cara mengakses MySQL, yaitu berorientasi obyek (PDO) dan prosedural (mysqli). Bagaimana dengan mysqli? Sudah tidak boleh digunakan!.

Ada beberapa framework REST yang bagus bagi PHP, misalnya:

- [Phalcon](#)
- [Lumen](#)
- [Slim](#)
- [Laravel](#)
- [Silex](#)

Route	Metode	Jenis	JOSN Hasil	Deskripsi
<code>/employees</code>	GET	JSON	–	Meminta semua data pegawai
<code>/employees/{id}</code>	GET	JSON	–	Meminta data satu pegawai
<code>/employees</code>	POST	JSON	<pre>{"employee_name": "Adam", "employee_age": "34", "employee_salary" : "23421"}</pre>	Menambahkan record pegawai baru ke dalam database
<code>/employees</code>	PUT	JSON	<pre>{"employee_name": "Adam", "employee_age": "34", "employee_salary" : "23421", "id":21}</pre>	Mengupdate record pegawai tertentu
<code>/employees</code>	DELETE	JSON	<pre>{"id" : 59}</pre>	Menghapus data pegawai tertentu

Mari kita membuat contoh REST API. Kita mulai dengan membuat folder 'api/' di dalam root directory dari web server, misalnya /var/www/html/ atau c:\xampp\htdocs\. Folder ini akan menampung semua file yang diperlukan untuk tutorial ini. Kita akan membuat REST API untuk modul pegawai yang akan menangani request HTTP Get, Post, Put dan Delete untuk mengambil, menambahkan, mengupdate, dan menghapus record-record dari database MySQL.

Inilah file-file yang akan digunakan selama percobaan tutorial REST API ini:

1. index.php : Ini adalah file entri, sekaligus mencegah navigasi terhadap file-file di dalam folder.
2. connection.php : File ini digunakan untuk membangun koneksi ke database server MySQL.
3. v1/employees.php : File ini mengandung semua titik ujung (end point) dari REST API dan metode mewakili aksi yang diperlukan.
4. v1/.htaccess : File ini digunakan untuk mengatur redirect dan routing.

Skema Tabel Employee

Buatlah suatu database MySQL bernama 'db_test', atau jika anda sudah mempunyai database dengan nama lain, maka cukup salin kode di bawah ini untuk membuat tabel bernama tb_employee.

```
1 CREATE TABLE IF NOT EXISTS `tb_employee` (  
2   `id` int(11) NOT NULL AUTO_INCREMENT COMMENT 'primary key',  
3   `employee_name` varchar(255) NOT NULL COMMENT 'employee name',  
4   `employee_salary` double NOT NULL COMMENT 'employee salary',  
5   `employee_age` int(11) NOT NULL COMMENT 'employee age',  
6   PRIMARY KEY (`id`)  
7) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='datatable demo table'  
8 AUTO_INCREMENT=64 ;
```

Koneksi ke Database MySQL Menggunakan PHP mysqli

Kita akan membuat suatu file bernama `connection.php` dan menambahkan string koneksi MySQL untuk menghubungi MySQL dengan menggunakan metode `mysqli_connect()`. Berikut ini adalah kode yang perlu dimasukkan dalam file tersebut:

```
1 <?php  
2 Class dbObj {  
3     /* Database connection start */  
4     var $servername = "localhost";  
5     var $username = "root";  
6     var $password = "";  
7     var $dbname = "db_test";  
8     var $conn;  
9  
10    function getConnstring() {  
11        $con = mysqli_connect($this->servername, $this->username,  
12            $this->password, $this->dbname) or  
13            die("Connection failed: " . mysqli_connect_error());  
14  
15        /* check connection */  
16        if (mysqli_connect_errno()) {
```

```

17             printf("Connect failed: %s\n", mysqli_connect_error());
18             exit();
19         } else {
20             $this->conn = $con;
21         }
22         return $this->conn;
23     }
24 }
25 ?>

```

Catatan:

Anda harus mengganti nilai dari variabel string *database name*, *db username*, *db password* dan *hostname* sesuai dengan kondisi di komputer yang anda gunakan.

Pengaturan URL dengan Rewrite .htaccess

Kita akan membuat file `.htaccess` di dalam folder `v1/` dan memasukkan beberapa baris aturan untuk mengatur akses ke REST API dengan URL yang lebih singkat, bagus dan aman. Inilah baris-baris yang harus dimasukkan dalam file `.htaccess` tersebut:

```

1 RewriteEngine On # Turn on the rewriting engine
2 RewriteRule ^employees/?$ employees.php [NC,L]
3 RewriteRule ^employees/([0-9]+)/?$ employees.php?id=$1 [NC,L]

```

REST API untuk Mengambil Data Semua Pegawai

Kita akan membuat suatu request REST HTTP jenis GET untuk mengakses semua record pegawai dari database MySQL. Kita akan menggunakan query MySQL untuk mengambil data dari tabel `tb_employee` dan mengirimkan array data JSON ke client sebagai obyek respon.

Sekarang kita buat file bernama `employees.php` di dalam `v1` dan tambahkan file koneksi MySQL untuk mengakses data di dalam tabel di server database MySQL.

```

1 // Connect to database
2     include("../connection.php");
3     $db = new dbObj();
4     $connection = $db->getConnection();
5
6     $request_method=$_SERVER["REQUEST_METHOD"];

```

Kita menggunakan metode `$_SERVER` untuk mendapatkan informasi mengenai aksi, seperti request REST untuk menambahkan, mengedit atau menghapus.

Perintah `switch..case` digunakan untuk menyeleksi dan menentukan metode akses yang tepat untuk menangani request REST yang masuk.

```

1 switch($request_method) {
2     case 'GET':
3         // Retrive Products
4         if(!empty($_GET["id"])) {
5             $id=intval($_GET["id"]);

```

```

6             get_employees($id);
7         }
8         else {
9             get_employees();
10        }
11        break;
12    default:
13        // Invalid Request Method
14        header("HTTP/1.0 405 Method Not Allowed");
15        break;
16    }

```

Kita memanfaatkan request GET untuk mengambil data semua pegawai dari database, sedangkan untuk pegawai tunggal (tertentu) perlu melewati employee id. Kita mendefinisikan metode `get_employees()` yang berisi kode seperti di bawah ini:

```

1 function get_employees() {
2     global $connection;
3     $query="SELECT * FROM tb_employee";
4     $response=array();
5     $result=mysqli_query($connection, $query);
6
7     while($row=mysqli_fetch_array($result)) {
8         $response[]=$row;
9     }
10
11    header('Content-Type: application/json');
12    echo json_encode($response);
13 }

```

Metode `mysqli_query()` mengambil data dari tabel `tb_employee` di MySQL dan menyimpannya sebagai obyek ke dalam variabel `'result'`. Metode `json_encode()` digunakan untuk mengkonversi array data ke dalam string json.

Silakan akses URL REST API `http://localhost/api/v1/employees` menggunakan web browser dan kita akan mendapatkan informasi tentang semua record pegawai yang ada di dalam tabel `tb_employee`.

REST API untuk Mengambil Data Satu Pegawai

Kita akan menggunakan suatu request REST HTTP jenis GET untuk mengakses record tunggal (tertentu) dari pegawai dari database. Hampir sama dengan sebelumnya, tetapi diperlukan id dari pegawai yang akan diminta datanya. Fungsi `get_employees` di bawah ini merupakan modifikasi dari fungsi sebelumnya yang bernama sama.

```

1 function get_employees($id=0) {
2     global $connection;
3     $query="SELECT * FROM tb_employee";
4
5     if($id != 0) {
6         $query.=" WHERE id=".$id." LIMIT 1";
7     }
8
9     $response=array();

```

```

10     $result=mysqli_query($connection, $query);
11
12     while($row=mysqli_fetch_array($result)) {
13         $response[]=$row;
14     }
15
16     header('Content-Type: application/json');
17     echo json_encode($response);
18 }

```

Sekarang silakan akses URL REST API <http://localhost/api/v1/employees/1> dari web browser dan harusnya diperoleh sebaris record dari pegawai yang terdapat di dalam tabel tb_employee, tepatnya yang mempunyai Id = 1.

REST API untuk Menambahkan Data Baru

Kita akan membuat suatu REST API baru untuk menambahkan (insert) record pegawai baru ke dalam MySQL menggunakan PHP. Kita menggunakan request REST jenis POST karena kita akan menge"pos"kan beberapa data JSON ke server.

Dalam struktur switch sebelumnya, silakan tambahkan case untuk "POST", seperti di bawah ini:

```

1 case 'POST':
2 // Insert Product
3 insert_employee();
4 break;

```

Fungsi yang akan menangani saat ada permintaan penambahan data baru adalah metode `insert_employee()` di dalam file `employees.php`.

```

1 function insert_employee() {
2     global $connection;
3
4     $data = json_decode(file_get_contents('php://input'), true);
5     $employee_name=$data["employee_name"];
6     $employee_salary=$data["employee_salary"];
7     $employee_age=$data["employee_age"];
8
9     echo $query="INSERT INTO tb_employee SET
10     employee_name='".$employee_name."',
11     employee_salary='".$employee_salary."',
12     employee_age='".$employee_age."'";
13
14     if(mysqli_query($connection, $query)) {
15         $response=array(
16             'status' => 1,
17             'status_message' =>'Employee Added Successfully.'
18         );
19     }
20     else {
21         $response=array(
22             'status' => 0,
23             'status_message' =>'Employee Addition Failed.'
24         );

```

```

25     }
26
27     header('Content-Type: application/json');
28     echo json_encode($response);
29 }

```

REST API untuk Mengupdate Pegawai Tertentu

Selanjutnya adalah membuat kode program untuk menangani request REST API jenis HTTP PUT. Request ini ditujukan untuk mengupdate data yang telah ada di dalam database MySQL. Update didasarkan pada Id dari pegawai, karena itu Id pegawai harus dijadikan sebagai parameter dari fungsi untuk menangani pembaruan record pegawai tersebut.

Sebelumnya, kita harus menambahkan case baru “PUT” ke dalam struktur switch...case, seperti di bawah ini:

```

1 case 'PUT':
2 // Update Product
3 $id=intval($_GET["id"]);
4 update_employee($id);
5 break;

```

Now we will create `update_employee()` method into `employees.php` file. We will use `employee_id` which we want to update and second is updated data in json format.

```

1 function update_employee($id) {
2     global $connection;
3     $post_vars = json_decode(file_get_contents("php://input"),true);
4     $employee_name=$post_vars["employee_name"];
5     $employee_salary=$post_vars["employee_salary"];
6     $employee_age=$post_vars["employee_age"];
7
8     $query="UPDATE tb_employee SET employee_name='".$employee_name."',
9         employee_salary='".$employee_salary."',
10        employee_age='".$employee_age."' WHERE id=".$id;
11
12    if(mysqli_query($connection, $query)) {
13        $response=array(
14            'status' => 1,
15            'status_message' =>'Employee Updated Successfully.'
16        );
17    }
18    else {
19        $response=array(
20            'status' => 0,
21            'status_message' =>'Employee Updation Failed.'
22        );
23    }
24
25    header('Content-Type: application/json');
26    echo json_encode($response);
27 }

```

Perhatikan, kita menjadikan `employee_id` sebagai parameter. Jika update berhasil, pesan keberhasilan dikirimkan ke client REST API dalam format JSON.

REST API untuk Menghapus Pegawai Tertentu

Terakhir, kita harus membuat fungsi untuk menangani request REST API jenis DELETE. Request ini meminta server API untuk menghapus record tertentu. Seperti pada CREATE dan UPDATE, DELETE juga memerlukan Id Pegawai sebagai parameteranya.

Sebelumnya, sekita perlu menambahkan case "DELETE" ke dalam struktur switch ...case, seperti berikut ini:

```
1 case 'DELETE':
2 // Delete Product
3 $id=intval($_GET["id"]);
4 delete_employee($id);
5 break;
```

Metode `delete_employee()` di dalam file `employees.php` bertanggungjawab menangani penghapusan data pegawai tertentu. Field `employee_id` digunakan sebagai kunci penghapusan tersebut.

```
1 function delete_employee($id) {
2     global $connection;
3     $query="DELETE FROM tb_employee WHERE id=".$id;
4
5     if(mysqli_query($connection, $query)) {
6         $response=array(
7             'status' => 1,
8             'status_message' =>'Employee Deleted Successfully.'
9         );
10    }
11    else {
12        $response=array(
13            'status' => 0,
14            'status_message' =>'Employee Deletion Failed.'
15        );
16    }
17
18    header('Content-Type: application/json');
19    echo json_encode($response);
20 }
```

Kesimpulan

Kita telah membuat suatu RESTful API untuk mengambil informasi semua pegawai, mengambil satu pegawai saja, menambahkan entri pegawai baru, mengupdate data pegawai dan menghapus data pegawai yang telah ada dari database MySQL. Contoh yang diperlihatkan masih sederhana sekali. Kita masih dapat menambahkan keamanan (*security*) seperti token dan membatasi akses REST API dari pengguna yang tidak sah.