

Web API :: CRUD CI

Eri Haryanto, S.Kom.,
M.Kom.



Code Igniter + API

```
... mirror object to mirror_...  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
... selection at the end -add ...  
... _ob.select= 1  
... _ob.select=1  
... context.scene.objects.active  
... ("Selected" + str(modifier...  
... mirror_ob.select = 0  
... bpy.context.selected_ob  
... data.name = [obj.name]...  
... want( please select ...
```

```
... OPERATOR CLASSES ...  
... types.Operator):  
... X mirror to the selected  
... object.mirror_mirror_x"  
... ror X"
```



Why API in Code Igniter?

- ▶ Codeigniter is a Simple and Great PHP framework for web application development.
- ▶ There are many cases where Web based application needs communication with other application on web, then at the time we need RESTful API.
- ▶ If your web application has API then it can be very easily communicate with other application, because now API is an essential components of web based application.



What is REST API?

- ▶ We all know REST full form is Representational State Transfer.
- ▶ REST API is on type of Web services which has been use the HTTP different methods like POST, PUT, GET and DELETE for data exchange on the web.
- ▶ Dengan membangun API pada aplikasi web, dapat lebih mudah dalam mengembangkan aplikasi mobile untuk aplikasi tersebut.
- ▶ Aplikasi mobile dapat mengirim dan menerima data cukup menggunakan API.
- ▶ Dengan REST API web services menjadikan dua basis aplikasi (web dan mobile) memiliki database yang sama (satu). Sehingga ini yang menjadi kelebihan membangun API pada aplikasi web.



Step implementasi API pada CI

- Create Table
- Make Database Connection
- Create API Controller
- Create API Model
- Create Test API Controller
- Create View File for Output



Create Table

- Buat tabel untuk aplikasi CRUD

tbl_sample

```
CREATE TABLE `tbl_sample` (  
  `id` int(11) NOT NULL,  
  `first_name` varchar(250) NOT NULL,  
  `last_name` varchar(250) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
ALTER TABLE `tbl_sample`  
  ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `tbl_sample`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=1;
```

Make Database Connection

- Buka application/config/database.php dan definisikan mysql database configuration.

```
<?php
```

```
$active_group = 'default';  
$query_builder = TRUE;
```

```
$db['default'] = array(  
    'dsn' => "",  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => "",  
    'database' => 'testing',  
    'dbdriver' => 'mysqli',
```

```
    ...  
?>
```




Create API Controller

- ▶ Memulai RESTful API web services pada Codeigniter framework perlu dibuat Controller untuk **send** dan **received** API http request.
- ▶ Class Controller ini akan **received** HTTP request data dalam format JSON.
- ▶ Daftar method pada Controller:
 - ▶ `__construct()`
 - ▶ `index()`
 - ▶ `insert()`
 - ▶ `fetch_single()`
 - ▶ `update()`
 - ▶ `delete()`

application/controllers/Api.php (1)

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Api extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('api_model');
        $this->load->library('form_validation');
    }

    function index()
    {
        $data = $this->api_model->fetch_all();
        echo json_encode($data->result_array());
    }
}
```

application/controllers/Api.php (2)

```
function insert()
{
    $this->form_validation->set_rules("first_name", "First Name", "required");
    $this->form_validation->set_rules("last_name", "Last Name", "required");

    $array = array();
    if($this->form_validation->run())
    {
        $data = array(
            'first_name' => trim($this->input->post('first_name')),
            'last_name' => trim($this->input->post('last_name'))
        );
        $this->api_model->insert_api($data);
        $array = array(
            'success' => true
        );
    }
    else
    {
        $array = array(
            'error' => true,
            'first_name_error' => form_error('first_name'),
            'last_name_error' => form_error('last_name')
        );
    }
    echo json_encode($array, true);
}
```

application/controllers/Api.php (3)

```
function insert()
{
    $this->form_validation->set_rules("first_name", "First Name", "required");
    $this->form_validation->set_rules("last_name", "Last Name", "required");

    $array = array();
    if($this->form_validation->run())
    {
        $data = array(
            'first_name' => trim($this->input->post('first_name')),
            'last_name' => trim($this->input->post('last_name'))
        );
        $this->api_model->insert_api($data);
        $array = array(
            'success' => true
        );
    }
    else
    {
        $array = array(
            'error' => true,
            'first_name_error' => form_error('first_name'),
            'last_name_error' => form_error('last_name')
        );
    }
    echo json_encode($array, true);
}
```

application/controllers/Api.php (3)

```
function fetch_single()
{
    if($this->input->post('id'))
    {
        $data = $this->api_model->fetch_single_user($this->input->post('id'));
        foreach($data as $row)
        {
            $output['first_name'] = $row["first_name"];
            $output['last_name'] = $row["last_name"];
        }
        echo json_encode($output);
    }
}
```

application/controllers/Api.php (4)

```
function update()
{
    $this->form_validation->set_rules("first_name", "First Name", "required");
    $this->form_validation->set_rules("last_name", "Last Name", "required");
    $array = array();
    if($this->form_validation->run()) {
        $data = array(
            'first_name' => trim($this->input->post('first_name')),
            'last_name' => trim($this->input->post('last_name'))
        );
        $this->api_model->update_api($this->input->post('id'), $data);
        $array = array(
            'success' => true
        );
    } else {
        $array = array(
            'error' => true,
            'first_name_error' => form_error('first_name'),
            'last_name_error' => form_error('last_name')
        );
    }
    echo json_encode($array, true);
}
```

application/controllers/Api.php (5)

```
function delete() {  
    if($this->input->post('id')) {  
        if($this->api_model->delete_single_user($this->input->post('id')))  
        {  
            $array = array(  
                'success' => true  
            );  
        } else {  
            $array = array(  
                'error' => true  
            );  
        }  
        echo json_encode($array);  
    }  
}
```

Create API Model

Method pada Model untuk membangun API:

- `fetch_all()`
- `insert_api($data)`
- `fetch_single_user($user_id)`
- `update_api($user_id, $data)`
- `delete_single_user($user_id)`

application/models/Api_model.php (1)

```
<?php
class Api_model extends CI_Model{
function fetch_all() {
    $this->db->order_by('id', 'DESC');
    return $this->db->get('tbl_sample');
}
function insert_api($data) {
    $this->db->insert('tbl_sample', $data);
    if($this->db->affected_rows() > 0)
    {
        return true;
    }else{
        return false;
    }
}
}
```

application/models/Api_model.php (2)

```
function fetch_single_user($user_id){  
    $this->db->where("id", $user_id);  
    $query = $this->db->get('tbl_sample');  
    return $query->result_array();  
}  
function update_api($user_id, $data){  
    $this->db->where("id", $user_id);  
    $this->db->update("tbl_sample", $data);  
}
```

application/models/Api_model.php (3)

```
function delete_single_user($user_id) {  
    $this->db->where("id", $user_id);  
    $this->db->delete("tbl_sample");  
    if($this->db->affected_rows() > 0)  
    {  
        return true;  
    }else{  
        return false;  
    }  
}  
}
```

Create Test API Controller

- Setelah membangun RESTful API, perlu untuk dilakukan uji coba API tersebut untuk memastikan API berjalan sesuai dengan rancangan awal.
- URL testing → `http://localhost/api/api`



Resources

- Download, lakukan instalasi dan uji coba untuk melihat aplikasi API CRUD dengan Framework Code Igniter
- Project API CRUD → download di LMS