|  |  |
| --- | --- |
| **Date** | **Februari 7, 2018** |
| **Name** | **Deddy Lasmono Putro** | **Place** | **PIS – GG Gempol** |
| **Project** | **PROJ-SD-17.03.03-O** | | |
| **Task Number** | [**86434**](http://gggmpscdweb02/redmine/issues/86434) | | |
| **Subject** | Redmine V2 | | |

Contents

[1 Coding Guidelines 2](#_Toc505870465)

[1.1 Penamaan ID untuk UI 2](#_Toc505870466)

[1.2 Controller 2](#_Toc505870467)

[2 Design Information 5](#_Toc505870468)

[2.1 Front/Base Page 5](#_Toc505870469)

[2.1.1 Controller Information 5](#_Toc505870470)

[2.2 Sign In Page 5](#_Toc505870471)

[2.2.1 Controller Information 5](#_Toc505870472)

[2.2.1 Additional Information 6](#_Toc505870473)

# Coding Guidelines

## 1.1 Penamaan ID untuk UI

* Semua ID untuk UI dinamai dengan awalan 3 huruf yang menyatakan bentuk UI tersebut, sebagai contoh untuk button ‘Save’ maka diberi id ‘btn-save’.

Berikut adalah awalan-awalan yang dapat digunakan:

|  |  |
| --- | --- |
| **Widget** | **Nama** |
| Button | btn |
| Textfield | txt |
| Selectfield/combofield/combobox | cmb |
| Panel | pnl |
| Div | div |
| Table | tbl |
| Checkbox | cbx |
| List | lst |
| a/url | url |

## 1.2 Controller

Dalam Pembuatan controller terdapat 2 (dua) type controller, yaitu untuk data dan view. Dalam dokumen ini untuk type yang data akan kita beri tanda **API** pada table controller. Yang membedakan kedua controller dalam codingnya adalah kode awal, seperti dalam contoh berikut:

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

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

class sign\_in extends REST\_Controller {

function sign\_in(){

parent::\_\_construct();

$this->load->model(array('sign\_in\_model')); //load model yang digunakan

}

Code 1.2.1 Kode awal pada controller class API

Contoh kode di atas adalah template kode untuk api, dimana yang biasanya controller extends dari class CI\_Controller di ganti dengan REST\_Controller, dan sebelum penulisan class ditambahkan juga kode require APPPATH.'/libraries/REST\_Controller.php';

Controller API REST ini akan digunakan hanya untuk operasi data, dimana nantinya akan menampilkan data sesuai format yang diminta secara otomatis. Maka dari itu kode untuk menampilkan data pun juga berbeda dari biasanya, dari yang biasanya menggunakan echo atau print data berupa text maka diganti dengan cara seperti di bawah berikut:

function users\_get(){

$result = $this->users\_model->get\_users();

if(count($result) == 0)

$this->response(array(), 200);

else

$this->response($result);

}

**Code 1.2.2 kode fungsi dalam controller API**

Pada contoh diatas dicontohkan cara controller dalam menampilkan data user dengan REST API controller, data user sendiri ada pada variable $result yang berupa array/object. Dengan digunakannya cara ini nantinya maka fungsi tersebut bisa diakses dengan berbagai format. Sebagai contoh jika seandainya bila biasanya kita mengakses sebuah controller dengan **www.web.com/index.php/m\_user/users** maka hanya akan menampilkan data dengan format yang seusai dengan kode yang ada saja. Tetapi jika menggunakan seperti contoh diatas data bisa di akses dengan berbagai format, sebagai contoh **index.php/m\_user/users.json** maka data akan ditampilkan dengan format json, jika Anda panggil dengan **index.php/m\_user/users.jsonp?callback=fncallback** maka akan otomatis juga ditampilkan dengan format jsonp tanpa mengubah/menambah kode yang ada, begitu juga jika ingin mengambil data dalam bentuk xml, cukup ganti type nya dengan xml.

Saat ini ada tujuh (7) format yang telah disupport, format-format tersebut dapat dilihat pada tabel di bawah ini.

|  |  |
| --- | --- |
| **Format** | **Description** |
| .xml | Format data dalam XML |
| .json | Format data dalam JSON |
| .jsonp | Format data dalam JSONP, dengan parameter callback |
| .serialized | Format data dalam bentuk serialized PHP |
| .php | Format data dalam PHP |
| .html | Format data dalam HTML |
| .csv | Format data dalam CSV |

Table 1.2.1 Table format REST API

Pada contoh **Code 1.2.2** juga terdapat perbedaan dalam penamaan fungsi, yaitu adanya tambahan **\_get** pada nama fungsi. Hal ini merupakan type yang bisa digunakan dalam mengakses fungsi tersebut, dimana dalam mengaksesnya harus menggunakan **METHOD GET**. Terdapat empat (4) method yang dapat digunakan, seperti terdapat pada tabel berikut.

|  |  |
| --- | --- |
| **Method** | **Description** |
| GET | Untuk mendapatkan data/informasi |
| POST | Digunakan untuk mebuat data baru atau mengupdatenya |
| PUT | Sama seperti POST, perbedaannya adalah saat mengisi data yang sama maka data lama akan di overwrite |
| DELETE | Untuk menghapus data |

Tabel 1.2.2 Tabel REST method

Sedangkan pada controller selain API maka diharuskan terdapat kode berikut pada constructor nya.

/\*plugin system\*/

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

if($this->router->fetch\_class()."/".$this->router->fetch\_method()=="base/load"){

$libraries = directory\_map(APPPATH."libraries/".get\_class($this)."/index", 1, TRUE);

$folder = get\_class($this)."/index";

}

else{

$libraries = directory\_map(APPPATH."libraries/".$this->router->fetch\_class()."/".$this->router->fetch\_method(), 1, TRUE);

$folder = get\_class($this)."/".$this->router->fetch\_method();

}

$autoload = array();

$libs = array();

if($libraries)

foreach($libraries as $library)

{

if(!is\_array($library))

{

$pathinfo = pathinfo($library);

if(isset($pathinfo['extension']))

if($pathinfo['extension']=='php'){

$autoload[] = $folder.'/' . strtolower($pathinfo['filename']);

$libs[] = strtolower($pathinfo['filename']);

}

}

}

$this->load->library($autoload);

if(count($libs) > 0)

foreach($libs as $library){

if(method\_exists($this->{$library}, 'fn\_view')){

$view = $this->{$library}->fn\_view();

}

}

/\*end of plugin system\*/

Code 1.2.3 Kode plugin system yang harus ada pada controller selain API

# Design Information

## 2.1 Front/Base Page

### 2.1.1 Controller Information

|  |  |
| --- | --- |
| **Controller Name** | **Base.php** |
| **Function Name** | **Description** |
| index | * Untuk mengatur apakah akan mengirim page secara keseluruhan atau hanya content yang dibutuhkan (jika di panggil dengan ajax) * Jika dipanggil dengan ajax maka tampilkan sesuai fungsi yang dipangil dan proses plugin yang tersedia |
| **Function Name** | **Description** |
| not\_found | Tampilkan halaman 404 ‘not found’ jika halaman yang dituju tidak tersedia |
| **Function Name** | **Description** |
| load | * Menampilkan hanya content yang diminta jika dipanggil dengan ajax * Tampilkan halaman secara keseluruhan jika bukan ajax |
| **Parameter** | **Description** |
| segment | Segment dari controller yang akan di load |
| **Return** | |
| view/content dari controller yang dipanggil | |

## 2.2 Sign In Page

### 2.2.1 Controller Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Controller Name** | **api/Sign\_in.php** | **API** | |
| **Function Name** | **Description** | | |
| sign\_in\_post | Controller untuk melakukan login. | | |
| **Parameter (POST)** | **Description** | | |
| username | username dari user yang login | | |
| password | password dari user yang login | | |
| **Return** | **Success Value** | | **Failed/Error Value** |
| status | true | | false |
| message | Login Success. | | Login Failed. Please check your username or password! |
| **Function Name** | **Description** | | |
| forgot\_password\_post | Digunakan untuk mengirim email ke user link untuk melakukan perubahan password | | |
| **Parameter (POST)** | **Description** | | |
| username | Username dari orang yang lupa password | | |
| **Return** | **Success Value** | | **Failed/Error Value** |
| status | true | | false |
| message | Link to renew your password has been sent! | | Could not find your username! |
| **Controller Name** | **Sign\_in.php** | | |
| **Function Name** | **Description** | | |
| index | Menampilkan halaman sign in | | |

### 2.2.2 Additional Information

* jika login telah gagal 1 kali maka tampilkan link “Forgot Password”

### 2.2.3 View Information

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | |
| #btn-forgot-password | EVENT | click |
| TYPE | AJAX |
| controller | api/sign\_in/forgot\_password\_post |
| #btn-sign-in | EVENT | click |
| TYPE | AJAX |
| controller | api/sign\_in/sign\_in\_post |

## 2.3 My Page

### 2.3.1 Controller Information

|  |  |  |
| --- | --- | --- |
| **Controller Name** | **My\_page.php** | **API** |
| **Function Name** | **Description** | |
| dashboard\_get | Data tampilan dashboard dari user | |
| **Function Name** | **Description** | |
| save\_dashboard\_post | Save komposisi dashboard | |

|  |  |
| --- | --- |
| **Controller Name** | **My\_page.php** |
| **Function Name** | **Description** |
| index | Menampilkan halaman my page |

### 2.3.2 Additional Information

* Data yang didapatkan dari controller dashboard akan digunakan untuk menampilkan widget-widget pada dashboard.
* Untuk widget dashboard berupa query widget info terdapat pada controller/view query widget.
* Untuk widget lain hanya berupa tabel, yaitu reported tasks, watched tasks dan tasks assigned to me.
* Data untuk tabel-tabel tersebut dapat di ambil dari controller query widget.

## 2.4 Projects

### 2.4.1 Controller Information

|  |  |  |
| --- | --- | --- |
| **Controller Name** | **Projects.php** | **API** |
| **Function Name** | **Description** | |
| index\_get | Untuk mendapatkan data projects | |
| **Parameter (GET)** | **Description** | |
| iDisplayLength | Jumlah data yang diminta | |
| iDisplayStart | Posisi awal data yang diminta | |
| **Function Name** | **Description** | |
| index\_post | Untuk menyimpan/mengupdate project baru | |
| **Parameter (POST)** | **Description** | |
| Fields sesuai dengan tabel projects | Lihat info pada tabel projects | |
| Semua custom fields yang ada untuk projects | Lihat info pada tabel custom\_fields dan custom\_values | |