[Overview](http://docs.google.com/index.html)

### Packages

* [CodeIgniter](http://docs.google.com/package-CodeIgniter.html)
  + [Libraries](http://docs.google.com/package-CodeIgniter.Libraries.html)
  + [Rest](http://docs.google.com/package-CodeIgniter.Rest.html)
* [None](http://docs.google.com/package-None.html)

### Classes

* [Example](http://docs.google.com/class-Example.html)
* [Format](http://docs.google.com/class-Format.html)
* [Key](http://docs.google.com/class-Key.html)
* [REST\_Controller](http://docs.google.com/class-REST_Controller.html)
* [Rest\_server](http://docs.google.com/class-Rest_server.html)
* [Welcome](http://docs.google.com/class-Welcome.html)
* [Overview](http://docs.google.com/index.html)
* Package
* Class

[1:](#gjdgxs) <?php  
 [2:](#30j0zll)   
 [3:](#1fob9te) defined('BASEPATH') OR exit('No direct script access allowed');  
 [4:](#3znysh7)   
 [5:](#2et92p0) // This can be removed if you use \_\_autoload() in config.php OR use Modular Extensions  
 [6:](#tyjcwt) require APPPATH . '/libraries/REST\_Controller.php';  
 [7:](#3dy6vkm)   
 [8:](#1t3h5sf) /\*\*  
 [9:](#4d34og8)  \* This is an example of a few basic user interaction methods you could use  
 [10:](#2s8eyo1)  \* all done with a hardcoded array  
 [11:](#17dp8vu)  \*  
 [12:](#3rdcrjn)  \* @package CodeIgniter  
 [13:](#26in1rg)  \* @subpackage Rest Server  
 [14:](#lnxbz9)  \* @category Controller  
 [15:](#35nkun2)  \* @author Phil Sturgeon, Chris Kacerguis  
 [16:](#1ksv4uv)  \* @license MIT  
 [17:](#44sinio)  \* @link https://github.com/chriskacerguis/codeigniter-restserver  
 [18:](#2jxsxqh)  \*/  
 [19:](#z337ya) class Example extends REST\_Controller {  
 [20:](#3j2qqm3)   
 [21:](#1y810tw)  function \_\_construct()  
 [22:](#4i7ojhp)  {  
 [23:](#2xcytpi)  // Construct the parent class  
 [24:](#1ci93xb)  parent::\_\_construct();  
 [25:](#3whwml4)   
 [26:](#2bn6wsx)  // Configure limits on our controller methods  
 [27:](#qsh70q)  // Ensure you have created the 'limits' table and enabled 'limits' within application/config/rest.php  
 [28:](#3as4poj)  $this->methods['user\_get']['limit'] = 500; // 500 requests per hour per user/key  
 [29:](#1pxezwc)  $this->methods['user\_post']['limit'] = 100; // 100 requests per hour per user/key  
 [30:](#49x2ik5)  $this->methods['user\_delete']['limit'] = 50; // 50 requests per hour per user/key  
 [31:](#2p2csry)  }  
 [32:](#147n2zr)   
 [33:](#3o7alnk)  public function users\_get()  
 [34:](#23ckvvd)  {  
 [35:](#ihv636)  // Users from a data store e.g. database  
 [36:](#32hioqz)  $users = [  
 [37:](#1hmsyys)  ['id' => 1, 'name' => 'John', 'email' => 'john@example.com', 'fact' => 'Loves coding'],  
 [38:](#41mghml)  ['id' => 2, 'name' => 'Jim', 'email' => 'jim@example.com', 'fact' => 'Developed on CodeIgniter'],  
 [39:](#2grqrue)  ['id' => 3, 'name' => 'Jane', 'email' => 'jane@example.com', 'fact' => 'Lives in the USA', ['hobbies' => ['guitar', 'cycling']]],  
 [40:](#vx1227)  ];  
 [41:](#3fwokq0)   
 [42:](#1v1yuxt)  $id = $this->get('id');  
 [43:](#4f1mdlm)   
 [44:](#2u6wntf)  // If the id parameter doesn't exist return all the users  
 [45:](#19c6y18)   
 [46:](#3tbugp1)  if ($id === NULL)  
 [47:](#28h4qwu)  {  
 [48:](#nmf14n)  // Check if the users data store contains users (in case the database result returns NULL)  
 [49:](#37m2jsg)  if ($users)  
 [50:](#1mrcu09)  {  
 [51:](#46r0co2)  // Set the response and exit  
 [52:](#2lwamvv)  $this->response($users, REST\_Controller::HTTP\_OK); // OK (200) being the HTTP response code  
 [53:](#111kx3o)  }  
 [54:](#3l18frh)  else  
 [55:](#206ipza)  {  
 [56:](#4k668n3)  // Set the response and exit  
 [57:](#2zbgiuw)  $this->response([  
 [58:](#1egqt2p)  'status' => FALSE,  
 [59:](#3ygebqi)  'message' => 'No users were found'  
 [60:](#2dlolyb)  ], REST\_Controller::HTTP\_NOT\_FOUND); // NOT\_FOUND (404) being the HTTP response code  
 [61:](#sqyw64)  }  
 [62:](#3cqmetx)  }  
 [63:](#1rvwp1q)   
 [64:](#4bvk7pj)  // Find and return a single record for a particular user.  
 [65:](#2r0uhxc)   
 [66:](#1664s55)  $id = (int) $id;  
 [67:](#3q5sasy)   
 [68:](#25b2l0r)  // Validate the id.  
 [69:](#kgcv8k)  if ($id <= 0)  
 [70:](#34g0dwd)  {  
 [71:](#1jlao46)  // Invalid id, set the response and exit.  
 [72:](#43ky6rz)  $this->response(NULL, REST\_Controller::HTTP\_BAD\_REQUEST); // BAD\_REQUEST (400) being the HTTP response code  
 [73:](#2iq8gzs)  }  
 [74:](#xvir7l)   
 [75:](#3hv69ve)  // Get the user from the array, using the id as key for retreival.  
 [76:](#1x0gk37)  // Usually a model is to be used for this.  
 [77:](#4h042r0)   
 [78:](#2w5ecyt)  $user = NULL;  
 [79:](#1baon6m)   
 [80:](#3vac5uf)  if (!empty($users))  
 [81:](#2afmg28)  {  
 [82:](#pkwqa1)  foreach ($users as $key => $value)  
 [83:](#39kk8xu)  {  
 [84:](#1opuj5n)  if (isset($value['id']) && $value['id'] === $id)  
 [85:](#48pi1tg)  {  
 [86:](#2nusc19)  $user = $value;  
 [87:](#1302m92)  }  
 [88:](#3mzq4wv)  }  
 [89:](#2250f4o)  }  
 [90:](#haapch)   
 [91:](#319y80a)  if (!empty($user))  
 [92:](#1gf8i83)  {  
 [93:](#40ew0vw)  $this->set\_response($user, REST\_Controller::HTTP\_OK); // OK (200) being the HTTP response code  
 [94:](#2fk6b3p)  }  
 [95:](#upglbi)  else  
 [96:](#3ep43zb)  {  
 [97:](#1tuee74)  $this->set\_response([  
 [98:](#4du1wux)  'status' => FALSE,  
 [99:](#2szc72q)  'message' => 'User could not be found'  
[100:](#184mhaj)  ], REST\_Controller::HTTP\_NOT\_FOUND); // NOT\_FOUND (404) being the HTTP response code  
[101:](#3s49zyc)  }  
[102:](#279ka65)  }  
[103:](#meukdy)   
[104:](#36ei31r)  public function users\_post()  
[105:](#1ljsd9k)  {  
[106:](#45jfvxd)  // $this->some\_model->update\_user( ... );  
[107:](#2koq656)  $message = [  
[108:](#zu0gcz)  'id' => 100, // Automatically generated by the model  
[109:](#3jtnz0s)  'name' => $this->post('name'),  
[110:](#1yyy98l)  'email' => $this->post('email'),  
[111:](#4iylrwe)  'message' => 'Added a resource'  
[112:](#2y3w247)  ];  
[113:](#1d96cc0)   
[114:](#3x8tuzt)  $this->set\_response($message, REST\_Controller::HTTP\_CREATED); // CREATED (201) being the HTTP response code  
[115:](#2ce457m)  }  
[116:](#rjefff)   
[117:](#3bj1y38)  public function users\_delete()  
[118:](#1qoc8b1)  {  
[119:](#4anzqyu)  $id = (int) $this->get('id');  
[120:](#2pta16n)   
[121:](#14ykbeg)  // Validate the id.  
[122:](#3oy7u29)  if ($id <= 0)  
[123:](#243i4a2)  {  
[124:](#j8sehv)  // Set the response and exit  
[125:](#338fx5o)  $this->response(NULL, REST\_Controller::HTTP\_BAD\_REQUEST); // BAD\_REQUEST (400) being the HTTP response code  
[126:](#1idq7dh)  }  
[127:](#42ddq1a)   
[128:](#2hio093)  // $this->some\_model->delete\_something($id);  
[129:](#wnyagw)  $message = [  
[130:](#3gnlt4p)  'id' => $id,  
[131:](#1vsw3ci)  'message' => 'Deleted the resource'  
[132:](#4fsjm0b)  ];  
[133:](#2uxtw84)   
[134:](#1a346fx)  $this->set\_response($message, REST\_Controller::HTTP\_NO\_CONTENT); // NO\_CONTENT (204) being the HTTP response code  
[135:](#3u2rp3q)  }  
[136:](#2981zbj)   
[137:](#odc9jc) }  
[138:](#38czs75)

API documentation generated by [ApiGen](http://apigen.org)