wiredesignz / codeigniter-modular-extensions-hmvc

Overview

HTTPS https://bitbucket.org/wiredesignz/co

Last updated 2017-09-26 Language PHP Access level Read	11 Open PRs	783 Watchers
	1 Branch	155 Forks

Support development of Modular Extensions - HMVC

Donate

Modular Extensions - HMVC

Modular Extensions makes the CodeIgniter PHP framework modular. Modules are groups of independent components, typically model, controller and view, arranged in an application modules sub-directory that can be dropped into other CodeIgniter applications.

HMVC stands for Hierarchical Model View Controller.

Module Controllers can be used as normal Controllers or HMVC Controllers and they can be used as widgets to help you build view partials.

Features:

All controllers can contain an $autoload class variable, which holds an array of items to load prior to running the constructor. This can be used together with module/config /autoload.php, however using the $autoload variable only works for that specific controller.

```
<?php
class Xyz extends MX_Controller
{
    $autoload = array(
        'helper'    => array('url', 'form'),
        'libraries' => array('email'),
    );
}
```

The Modules::$locations array may be set in the application/config.php file. ie:

```
<?php
$config['modules_locations'] = array(
    APPPATH.'modules/' => '../modules/',
);
```

Modules::run() output is buffered, so any data returned or output directly from the controller is caught and returned to the caller. In particular, $this->load->view() can be used as you would in a normal controller, without the need for return.

Controllers can be loaded as class variables of other controllers using $this->load->module('module/controller'); or simply $this->load->module('module'); if the controller name matches the module name.

Any loaded module controller can then be used like a library, ie: $this->controller->method(), but it has access to its own models and libraries independently from the caller.

All module controllers are accessible from the URL via module/controller/method or simply module/method if the module and controller names match. If you add the _remap() method to your controllers you can prevent unwanted access to them from the URL and redirect or flag an error as you like.

Notes:

To use HMVC functionality, such as Modules::run(), controllers must extend the MX_Controller class.

To use Modular Separation only, without HMVC, controllers will extend the CodeIgniter Controller class.

You must use PHP5 style constructors in your controllers. ie:

```
<?php
class Xyz extends MX_Controller
{
    function __construct()
    {
        parent::__construct();
    }
}
```

Recent activity

- Add support http verb for ea...
  Pull request #25 unapproved in...
  Fuad Muhtaz · 2017-09-26
- Add support http verb for ea...
  Pull request #25 approved in wi...
  Fuad Muhtaz · 2017-09-26
- Add support http verb for ea...
  Pull request #25 created in wir...
  Fuad Muhtaz · 2017-09-26
- fixed for erro 'Call to undefin...
  Pull request #24 created in wir...
  Jay-r Simpron · 2017-09-03
- Fix Loader.php for CI 3.1.3
  Pull request #17 commented o...
  Salman Ali · 2017-08-27
- Loader.php edited online wit...
  Pull request #22 commented o...
  Ahila · 2017-08-17
- Loader.php edited online wit...
  Comment deleted from pull req...
  Ahila · 2017-08-17
- Loader.php edited online wit...
  Pull request #22 commented o...
  Ahila · 2017-08-17
- Loader.php edited online wit...
  Pull request #22 approved in wi...
  Cesar Vieira · 2017-07-10
- Fix Loader.php for CI 3.1.3
  Pull request #17 approved in wi...
  Cesar Vieira · 2017-07-10
- Fix Loader.php for CI 3.1.3
  Pull request #17 commented o...
  wolfgang1983 · 2017-07-01

codeigniter-modular-...

Overview

Source

Commits

Branches

Pull requests

Downloads