Generated Documentation

Powered by phpDocumentor

Contents

Package CodeIgniter Procedural Elements

Controller.php

CodeIgniter

An open source application development framework for PHP 5.1.6 or newer

- **Package** CodeIgniter
- **Author** ExpressionEngine Dev Team
- **Copyright** Copyright (c) 2008 - 2011, EllisLab, Inc.
- **Link** http://codeigniter.com
- **Since** Version 1.0
- **Filesource** Source Code for this file
- **License** http://codeigniter.com/user_guide/license.html

Model.php

CodeIgniter

An open source application development framework for PHP 5.1.6 or newer

- **Package** CodeIgniter
- **Author** ExpressionEngine Dev Team
- **Copyright** Copyright (c) 2008 - 2011, EllisLab, Inc.
- **Link** http://codeigniter.com
- **Since** Version 1.0
- **Filesource** Source Code for this file
- **License** http://codeigniter.com/user_guide/license.html

Package CodeIgniter Classes

Class CI_Controller

[line 30]

CodeIgniter Application Controller Class

This class object is the super class that every library in CodeIgniter will be assigned to.

- **Package** CodeIgniter
- **Sub-Package** Libraries
- **Author** ExpressionEngine Dev Team
- **Link** http://codeigniter.com/user_guide/general/controllers.html

CI_Controller::$instance

mixed = [line 32]

- **Static**
- **Access** private

Constructor *void* function CI_Controller::__construct() *[line 37]*

Constructor

- **Access** public

void function CI_Controller::get_instance() *[line 56]*

- **Static**
- **Access** public

Class CI_Model

[line 27]

CodeIgniter Model Class

- **Package** CodeIgniter
- **Sub-Package** Libraries
- **Author** ExpressionEngine Dev Team
- **Link** http://codeigniter.com/user_guide/libraries/config.html

Constructor *void* function CI_Model::__construct() *[line 34]*

Constructor

- **Access** public

void function CI_Model::__get($key) *[line 48]*

Function Parameters:

- *string* **$key**

__get

Allows models to access CI's loaded classes using the same syntax as controllers.

- **Access** private

Package SMAQC Procedural Elements

smaqc.php

smaqc.php

File containing the default CodeIgniter controller for SMAQC.

- **Package** SMAQC
- **Sub-Package** controllers
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Author** Aaron Cain
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

Package SMAQC Classes

Class Smaqc

[line 26]

CodeIgniter model for a SMAQC metric

- **Package** SMAQC
- **Sub-Package** controllers
- **Author** Aaron Cain
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0

Smaqc::$defaultenddate

mixed = [line 29]

Smaqc::$defaultstartdate

mixed = [line 28]

Constructor *void* function Smaqc::__construct() *[line 31]*
void function Smaqc::index() *[line 76]*
void function Smaqc::instrument($name, [$metric = NULL], [$start = NULL], [$end = NULL]) *[line 92]*

Function Parameters:

- **$name**
- **$metric**
- **$start**
- **$end**

- **Access** public

void function Smaqc::invaliditem([$requesteditemtype = NULL], [$name = NULL]) *[line 220]*

Function Parameters:

- **$requesteditemtype**
- **$name**

- **Access** public

instrumentmodel.php

instrumentmodel.php

File containing a CodeIgniter model for an instrument.

- **Package** SMAQC
- **Sub-Package** models
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

metricmodel.php

metricmodel.php

File containing a CodeIgniter model for a SMAQC metric.

- **Package** SMAQC
- **Sub-Package** models
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

Class Instrumentmodel

[line 24]

CodeIgniter model for an instrument

- **Package** SMAQC
- **Sub-Package** models
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0

Instrumentmodel::$averagedmetrics

object = [line 83]

The averaged metrics for the instrument over the provided date range.

The type is what is returned by a call to CI's Active Record db->get().

- **Access** private

Instrumentmodel::$definition

string = [line 40]

The definition of the instrument.

A string that is retrieved from a database.

- **TODO** Need/want this? Also needs implementing (the string is just set to lorem ipsum).
- **Access** private

Instrumentmodel::$enddate

string = [line 56]

The end date for grabbing metrics.

This should be a human readable string of the format m-d-Y. (Example: 12-12-2012)

- **Access** private

Instrumentmodel::$instrument

string = [line 30]

The name of the instrument.

- **Access** private

Instrumentmodel::$latestmetrics

object = [line 76]

The latest metrics for the instrument.

The type is what is returned by a call to CI's Active Record db->get().

- **Access** private

Instrumentmodel::$metricnames

array = [line 69]

A list (php array) of the metrics available in the database.

- **Access** private

Instrumentmodel::$startdate

string = [line 48]

The start date for grabbing metrics.

This should be a human readable string of the format m-d-Y. (Example: 11-11-2011)

- **Access** private

Instrumentmodel::$status

string = [line 63]

The status of the instrument (green/yellow/red)

- **TODO** Actually use/implement this
- **Access** private

Constructor *Instrumentmodel* function Instrumentmodel::__construct() *[line 95]*

Constructor

The contructor for Instrumentmodel simply calls the constructor for the base class (CI_Model). All initialization of the class must be done using the initialize function. The reasoning for this has to do with the way CI uses loads models (they cannot take arguments).

array|boolean function Instrumentmodel::initialize($instrument, $start, $end) *[line 154]*

Function Parameters:

- *string* **$instrument** The name of the instrument.
- *string* **$start** A human readable string for the start of the date range. Assumed to be in m-d-Y format. (Example: 11-11-2011)
- *string* **$end** A human readable string for the end of the date range. Assumed to be in m-d-Y format. (Example: 12-12-2012)

Initializer for the Instrument model

Gets all of the needed values for the class/model from the database. Calculates any values that need calculating.

- **Access** public

mixed function Instrumentmodel::__get($what) *[line 112]*

Function Parameters:

- *string* **$what** The member of the class that we are looking for.

__get

The custom __get() function is essentially a getter for our private members in the class. The use of the variable $CI and get_instance() allows us to access CI's loaded classes using the same syntax as in the controller.

Class Metricmodel

[line 24]

CodeIgniter model for a SMAQC metric

- **Package** SMAQC
- **Sub-Package** models
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0

Metricmodel::$average

float = 0 *[line 67]*

The average value of the metric over the provided date range.

- **Access** private

Metricmodel::$data

object = [line 74]

The results of querying the database for the metric values.

The type is what is returned by a call to CI's Active Record db->get().

- **Access** private

Metricmodel::$definition

string = [line 45]

The definition of the instrument.

A string that is retrieved from a database.

- **TODO** Needs implementing (the string is just set to lorem ipsum).
- **Access** private

Metricmodel::$enddate

string = [line 61]

The end date for grabbing metrics.

This should be a human readable string of the format m-d-Y. (Example: 12-12-2012)

- **Access** private

Metricmodel::$instrument

string = [line 30]

The name of the instrument.

- **Access** private

Metricmodel::$metric

string = [line 36]

The name of the metric.

- **Access** private

Metricmodel::$plotdata

string = [line 82]

A JSON encoded array of (x,y) values for jqplot to use.

The x value is a time/date in milliseconds. The type is what is returned by a call to CI's Active Record db->get().

- **Access** private

Metricmodel::$startdate

string = [line 53]

The start date for grabbing metrics.

This should be a human readable string of the format m-d-Y. (Example: 11-11-2011)

- **Access** private

Constructor *Metricmodel* function Metricmodel::__construct() *[line 94]*

Constructor

The contructor for Metricmodel simply calls the constructor for the base class (CI_Model). All initialization of the class must be done using the initialize function. The reasoning for this has to do with the way CI loads models in the controller (they cannot take arguments).

array|boolean function Metricmodel::initialize($instrument, $metric, $start, $end) *[line 154]*

Function Parameters:

- *string* **$instrument** The name of the instrument.
- *string* **$metric** The name of the metric.
- *string* **$start** A human readable string for the start of the date range. Assumed to be in m-d-Y format. (Example: 11-11-2011)
- *string* **$end** A human readable string for the end of the date range. Assumed to be in m-d-Y format. (Example: 12-12-2012)

Initializer for the Metric model

Gets all of the needed values for the class/model from the database. Calculates any values that need calculating.

- **Access** public

mixed function Metricmodel::__get($what) *[line 111]*

Function Parameters:

- *string* **$what** The member of the class that we are looking for.

__get

The custom __get() function is essentially a getter for our private members in the class. The use of the variable $CI and get_instance() allows us to access CI's loaded classes using the same syntax as in the controller.

footView.php

footView.php

File containing the ending code for all views in SMAQC.

- **Package** SMAQC
- **Sub-Package** views
- **Author** Aaron Cain
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

headView.php

headView.php

File containing the opening code for all views in SMAQC.

- **Package** SMAQC
- **Sub-Package** views
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Author** Aaron Cain
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

instrumentView.php

instrumentView.php

File containing the code for the instrumentView.

- **Package** SMAQC
- **Sub-Package** views
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

invaliditemView.php

invaliditemView.php

File containing the code for the invaliditemView.

- **Package** SMAQC
- **Sub-Package** views
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

leftMenuView.php

leftMenuView.php

File containing the code for the leftMenuView loaded on each page.

- **Package** SMAQC
- **Sub-Package** views
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

mainView.php

mainView.php

File containing the code for the default view for SMAQC.

- **Package** SMAQC
- **Sub-Package** views
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Author** Aaron Cain
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

metricView.php

instrumentView.php

File containing the code for the metricView.

- **Package** SMAQC
- **Sub-Package** views
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

topMenuView.php

topMenuView.php

File containing the code for the topMenuView loaded on each page.

- **Package** SMAQC
- **Sub-Package** views
- **Author** Trevor Owen < trevor.owen@email.wsu.edu>
- **Author** Aaron Cain
- **Version** 1.0
- **Copyright** TODO
- **License** TODO

Appendices

Appendix A - Class Trees

Package SMAQC

Instrumentmodel

- CI_Model **(Different package)**
  - Instrumentmodel

Metricmodel

- CI_Model **(Different package)**
  - Metricmodel

Smaqc

- CI_Controller **(Different package)**
  - Smaqc

Package CodeIgniter

CI_Controller

- CI_Controller
  - Smaqc

CI_Model

- CI_Model
  - Instrumentmodel
  - Metricmodel

Appendix C - Source Code

Package SMAQC

File Source for Controller.php

Documentation for this file is available at Controller.php

```
<?php     if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/**
 * CodeIgniter
 *
 * An open source application development framework for PHP 5.1.6 or newer
 *
 * @package        CodeIgniter
 * @author        ExpressionEngine Dev Team
 * @copyright    Copyright (c) 2008 - 2011, EllisLab, Inc.
 * @license        http://codeigniter.com/user_guide/license.html
 * @link        http://codeigniter.com
 * @since        Version 1.0
 * @filesource
 */

// ------------------------------------------------------------------------

/**
 * CodeIgniter Application Controller Class
 *
 * This class object is the super class that every library in
 * CodeIgniter will be assigned to.
 *
 * @package        CodeIgniter
 * @subpackage    Libraries
 * @category    Libraries
 * @author        ExpressionEngine Dev Team
 * @link        http://codeigniter.com/user_guide/general/controllers.html
 */
class CI_Controller {

    private static $instance;

    /**
     * Constructor
     */
    public function __construct()
    {
        self::$instance =&     $this;

        // Assign all the class objects that were instantiated by the
        // bootstrap file (CodeIgniter.php) to local class variables
        // so that CI can run as one big super object.
        foreach (is_loaded() as $var =>    $class)
        {
            $this->   $var =&     load_class($class);
        }

        $this->   load =&     load_class('Loader', 'core');

        $this->   load->   initialize();

        log_message('debug', "Controller Class Initialized"          );
    }

    public static function &    get_instance()
    {
        return self::$instance;
    }
}
// END Controller class

/* End of file Controller.php */
/* Location: ./system/core/Controller.php */
```

File Source for Model.php

Documentation for this file is available at Model.php

```
<?php    if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/**
 * CodeIgniter
 *
 * An open source application development framework for PHP 5.1.6 or newer
 *
 * @package        CodeIgniter
 * @author        ExpressionEngine Dev Team
 * @copyright    Copyright (c) 2008 - 2011, EllisLab, Inc.
 * @license        http://codeigniter.com/user_guide/license.html
 * @link        http://codeigniter.com
 * @since        Version 1.0
 * @filesource
 */

// ------------------------------------------------------------------------

/**
 * CodeIgniter Model Class
 *
 * @package        CodeIgniter
 * @subpackage    Libraries
 * @category    Libraries
 * @author        ExpressionEngine Dev Team
 * @link        http://codeigniter.com/user_guide/libraries/config.html
 */
class CI_Model {

    /**
     * Constructor
     *
     * @access public
     */
    function __construct()
    {
        log_message('debug', "Model Class Initialized"          );
    }

    /**
     * __get
     *
     * Allows models to access CI's loaded classes using the same
     * syntax as controllers.
     *
     * @param    string
     * @access private
     */
    function __get($key)
    {
        $CI =&     get_instance();
        return $CI->   $key;
    }
}
// END Model Class

/* End of file Model.php */
/* Location: ./system/core/Model.php */
```

Appendix D - Todo List

In Package SMAQC

In **Instrumentmodel::$definition**

- Need/want this? Also needs implementing (the string is just set to lorem ipsum).

In **Metricmodel::$definition**

- Needs implementing (the string is just set to lorem ipsum).

In **Instrumentmodel::$status**

- Actually use/implement this

Index