CX Framework

Version 0.0.2

My Common name spaces:

\cx\app\view – Views
\cx\model\NAME OF MODEL Class – Your DB Model
\cx\database\model – Generic Database Model – Object for use with PDO
\cx\app\main_functions::METHOD TO USE (Note :: for static methods!)
\cx\common\crypt::make_hash($data);

Config data:

Setting multiple things with data: cx_configure::set('KEY', array('Akey'=>'Data','key2'=>'D2'));

```
cx_configure::set('cx', array(
  'version' => '0.0.2',
  'login' => 'login_', // Name for login session variable, note: CX_SESS is will be added to the prefix
));
```

Getting data from an Array: cx_configure::a_get('KEY','ARRAY KEY NAME');
echo cx_configure::a_get('cx','version'); // Displays version #

Setting one item: cx_configure::set('AUTHOR','Bob');
Getting NON – ARRAY data: cx_configure::get('AUTHOR');

So, cx_configure::a_get('CONFIG KEY', 'Array keyname') is for getting the config key from an array, and get(CONFIG KEY) is for getting Non-Array Data or returns the entire Array of CONFIG KEY.

Creating a Controller:

```
class cx_loader_FOLDERNAME_FILENAME extends cx\app\app {
// ex: class cx_loader_app_testing extends cx\app\app {
// file: ProjectName/controllers/app/testing.php
        public function index() {
            // code….
// URL = http://localhost/PROJECTFOLDER/FOLDERNAME/FILENAME/METHODNAME.html
            // URL EX = http://localhost/newtoolz/app/testing/index.html
        }
}
```

Working with Session Data:

```
$this->session->set_session_var('credits', 5); // = $_SESSION[CX_SESS . 'credits'] = 5;
$this->session->session_var('credits'); // = $_SESSION[CX_SESS . 'credits'] ….
```

Working with Input:
See file: dealertoolz/cx/classes/cx/app/request.php
$this->request->METHOD... where METHOD is listed below:
is_ajax() // Returns true if AJAX request

get_var('fname'); // same as $_GET['fname']
post_var('fname'); // same as $_POST['fname']
request_var('fname'); // same as $_REQUEST['fname'] – Do not use as it incurs Security vulnerabilities
cookie_var('auth'); // Reads Cookie named auth, returns the content

clean_email($var); // Sanitizes Email addresses to be safe
encode_get_var($data); // Does an URL Encode and Sanitizes it, use before setting dynamic a URI GET Variable in page.
decode_get_var($this->request->get_var('fname')); // Un-encodes URI data, that was encoded...
encode_clean($content); // Uses htmlentities on the data, so it can be outputted in HTML page, this will prevent XSS Attacks!!!
decode_clean($data); // Uses html_entity_decode...

set_cookie_var($var, $content, $expires, "days"); // $var is name of Cookie, $content may be an Array, $expires is the number of Days/Hours..., “days” is for Days, “hours” is for Hours...

Unsets, all $_POST variables that are not in the exception array, see below:
set_allowed(array('year'=>'number','firstname','lastname'), 'post'); // Keeps ONLY: $_POST['year'], … firstname, and lastname !!!

Preventing Cross Site Attacks:
get_csrf_token(); // Return the CSRF Token for use in JavaScript POST...
echo do_csrf_token(); // Inserts an Hidden Input field into HTML with the Security Token
verify_csrf_token(); // Used in Controller to verify page is from a Valid User/Browser....

get_json_post_data($return_as_array); // $return_as_array, if true returns an JQUERY POST array, if false returns an JQUERY POST Object...

```
/*   To decode JQuery encoded objects, arrays, strings, int, bool types.
         The content must be of application/json. */
```

compair_it($var, $var2); // Checks if two strings are Equal
s_lowercase_trim($var); // Returns lowercase Trimmed String
is_valid_id($var); // Checks if ID is a number over the Value of Zero
is_not_valid_id($var) // Checks if ID is INVALID (Value below One)
is_int_set($var); // Checks if Number is NOT equal to -1
is_int_not_set($var); // Checks if Number is equal to -1
is_not_empty($var); // Checks if String is NOT Empty
is_set($var); // Checks if Variable was Set
is_empty($var); // Checks if Variable was NOT SET OR is Empty
is_not_set($var); // Checks if Variable was NOT Set

Setting View Output for Assets CSS/JS:

CX_BASE_REF – URI/URL to CX
PROJECT_BASE_REF – URI/URL to Project
Examples of usage:

```
$this->add_js(PROJECT_BASE_REF . '/assets/list_users.js'); // Adds Javascript
$this->add_css(CX_BASE_REF.'/assets/font-awesome/css/font-awesome.min.css'); // Adds CSS
```

```
// This next block of code, will include ssp_test.jsp → Note that this is JavaScript with PHP!!! Cool.
    $js_view = new \cx\app\javascript_view('app/testing/ssp_test'); // ssp_test.jsp
    $js = $js_view->fetch($this);
    $this->add_to_javascript($js);
```

Making a Short URI Link:

```
<a href="<?=$this->get_url("/app/". DEFAULT_PROJECT, "logout");?>”>Logout</a>
```

Note: get_url(“Route”,”Method”);

URL Re-Directing:

```
cx_redirect_url($this->get_url('/app/' . DEFAULT_PROJECT, 'main'));
// Route, Controller Method
```

Displaying a View:

```
// Controller Methods must be Public for them to work
  public function show_name() {
// Get input
    $fname = $this->request->post_var('fname'); // $_POST['fname']
// Grap View File if any
    $view = new \cx\app\view('app'.DS.'index'); // ProjectViewDIR/app/index.php
// Add content, manually if desired, IE to avoid making a view file
    $view->content = 'Hello, World!';
// Set the template, if desired output is an HTML Page
    $view->set_template('page'); // Project DIR + '/templates/' + page.tpl.php
// Set any Error messages via set_error
    $this->set_errors(array('api'=>array('Login'=>'Failed to get user/pwd.')));
// Set Message for next page to see
    \cx\app\main_functions::set_message('Bad Username/Password!', 'danger');
// Set variables to pass to the View file
    $view->set('firstname', $fname);
// Lastly, display the View
    $view->fetch($this); // Please always put $this  inside of fetch!!
  }
```

Notes:
If you do not use a template, please echo $this->scripts; // so JavaScript runs
Also, inside of your views: Do not use $this->VAR; Use: $local->VAR; or $local->Method();

Database:

```
    $this->load_model('users' . DS . 'users'); // Project Dir + “models/users/users.php”
    $users = new cx\model\users();
    $success = $users->is_user($username, $password); // Returns true if is user...
```

// OR without a Model do:

```
    $this->load_model(); // Does Require db.php and model.php for Database to work!!
    $db_options = array('table'=>'users', 'key'=>'id'); // Always, set table name and key field
    $users = new cx\database\model($db_options);
    $users->set_primary_key('username'); // Changes the field to load from key of 'id' to 'username'
    $username = 'bob_586';
    $password = 'FAKE_PWD';
    $users->load($username); // Load will do A Select Query * WHERE
                                          the Key is Equal to the Variable passed to the load method....
    // Note: get_member('name of field') // fetches data from the SELECT Query
    // Also, you may use get_members(); // Which returns the whole result set
    if ($users->get_member($db_options['key']) > 0 &&
            $users->check_pwd($password, $users->get_member('password'))) {
       echo 'Logged IN';
    } else {
       echo 'BAD PASSWORD or USERNAME';
    }
```

Saving Data to the Database:

```
// Do DB Code...
// Load....
    $users->set_member('fname', 'Bobby'); // Sets the field fname to Bobby
    $success = $this->save();  // Will Insert data if Primary key is empty, else Updates...
```

Example of a Database Model here we will create a testing model:

Note: Just change the class name to your name of the model file....

```
namespace cx\model;
use cx\database\model as the_model;
use cx\common\crypt as crypt;

class testing extends the_model {

  protected $table = 'test';
  protected $key = 'id';

  public function __construct($options = array()) {
    $options['table'] = $this->table;
    $options['key'] = $this->key;
    parent::__construct($options);
  }

  public function caching() {
    return false;
  }

  public function get_test($id) {
    $table = 'test';
    $options['fields'] = "*";
    $options['fetch'] = "row";

    $bind = array(":id" => $id);
    $options['where'] = "`id` = :id";

    $data = $this->database->select($table, $options, $bind); // See file: cx/classes/cx/database/db.php
    return $data['data'];
  }

}
```

CURL example:

```
public function test_curl() {
    $curl = $this->load_class('cx\app\cx_curl');
    $curl->hostname = 'localhost';
    $curl->port = 80;
    $curl->ssl = false;
    $curl->json_decode = true;
    cx_dump($curl->post(PROJECT_BASE_REF. '/api/app/testing/ajax_name', array('name'=>'bob',
'return'=>'json', 'debug'=>'true')));
  }
```

Basic Paginator

```
 public function generic() {
    $this->load_model();
    $db_options = array('table'=>'test', 'key'=>'id');
    $test = new cx\database\model($db_options);
//    $options['where'] = " BLAKey=BLAData ";
    $options['fields'] = "`test`.`id`, `test`.`data`";
    $options['paginator'] = 'true';
    $test->load("", $options);

    $view = new \cx\app\view('app/testing/generic');
    $allow_html = false;
    $view->set('rows', $test->get_members($allow_html));
    if ($test->get_paginator_object() === false) { // Useful, if you disable paginator
      $view->set('no_results', false);
      $view->set('paginator_items', "");
      $view->set('paginator_links', "");
      $view->set('paginator_entries', "");
    } else {
      $view->set('no_results', ($test->get_paginator_object()->items_total > 0) ? false : true);
      $view->set('paginator_items', $test->get_paginator_object()->display_items_per_page());
      $view->set('paginator_links', $test->get_paginator_links());
      $view->set('paginator_entries', $test->get_paginator_object()->get_entries());
    }
    $view->set_template('page');
    $view->fetch($this);
  }
```

View file:

```
<?php
if ($no_results === true) {

  $fade = true;
  echo \cx\app\main_functions::do_alert('No results!', 'info', $fade);

} else {

  echo $paginator_items . "<br><br>";

  foreach($rows as $row) {
    echo "ID#{$row['id']} : {$row['data']} <br>";
  }

  echo $paginator_links;
  echo $paginator_entries;
}
```