Welcome to LucidFrame

LucidFrame is a micro application development framework - a toolkit for PHP users. It provides several general purpose helper functions and logical structure for web application development. The goal is to provide a structured framework with small footprint that enables rapidly robust web application development.

LucidFrame is simple, fast and easy to install. The minimum requirements are a web server and a copy of LucidFrame.

License

LucidFrame is licensed under the MIT license. This means that you are free to modify, distribute and republish the source code on the condition that the copyright notices are left intact. You are also free to incorporate LucidFrame into any Commercial or closed source application.

Prerequisites

* Web Server (For example, Apache with mod\_rewrite enabled)
* PHP version 5.1.6 or newer (mcrypt extension enabled, but by no means required.)
* MySQL 5.0+ with MySQLi enabled.
* jQuery (LucidFrame provides AJAX Form and List APIs which require [jQuery](http://jquery.com/))

Installation

You can get a fresh copy of LucidFrame on GitHub: <https://github.com/cithukyaw/LucidFrame>. You can also clone the repository using git (<https://git-scm.com>)

git clone https://github.com/cithukyaw/LucidFrame.git

Regardless of how you downloaded it, place the code inside of your DocumentRoot.

In **production**, your directory setup may look something like the following so that it can be accessible via <http://www.example.com>.

|  |
| --- |
| /path\_to\_webserver\_document\_root  /app  /business  /css  /files  /helpers  /i18n  /images  /inc  /js  /tests  /vendors  /xhr  .htaccess  index.php  LICENSE.txt  README.md  robots.txt |

In this case, the configuration file at /inc/config.php should look like this:

|  |  |
| --- | --- |
| 88  89  90 | # $lc\_baseURL: No trailing slash (only if it is located in a sub-directory)  # Leave blank if it is located in the document root  $lc\_baseURL = ''; |

In **development**, you could have a directory name, for example, LucidFrame. Then your directory setup may look something like the following so that it can be accessible via http://localhost/LucidFrame

|  |
| --- |
| /path\_to\_webserver\_document\_root  /LucidFrame  /app  /business  /css  /files  /helpers  /i18n  /images  /inc  /js  /tests  /vendors  /xhr  .htaccess  index.php  LICENSE.txt  README.md  robots.txt |

In this case, the configuration file at /inc/config.php should look like this:

|  |  |
| --- | --- |
| 88  89  90 | # $lc\_baseURL: No trailing slash (only if it is located in a sub-directory)  # Leave blank if it is located in the document root  $lc\_baseURL = 'LucidFrame'; |

You can name the directory as you preferred, but you must change $lc\_baseURL respective to your directory name.

Application Framework Structure

| Directory | Description |
| --- | --- |
| app | This directory structure contains the application files and folders of your site. |
| app/helpers | The helpers mapping to the system core helpers should be placed in this directory. They are auto-loaded. For example, the custom validation helper (validation\_helper.php) should be placed in this directory and it is auto-loaded across the site. The following helper files are allowed:   * auth\_helper.php * db\_helper.php * pager\_helper.php * session\_helper.php * utility\_helper.php * validation\_helper.php |
| app/inc | The directory can include the site template files and site configuration file.   * site.config.php (inherited to /inc/site.config.php) * head.php (overwritable to /inc/head.php) * 404.php (overwritable to /inc/404.php) * header.php (overwritable to /inc/header.php if you have) * footer.php (overwritable to /inc/footer.php if you have) |
| app/js | The application javascript files should be placed in this directory. |
| business | This directory should be used to place the files which have the business logic functions or classes. They usually do the direct operations to the database layer. |
| css | This directory contains the application CSS files. |
| files | This directory contains the files and folders of your site uploaded data.  For example, sessions, photos, cache, etc. |
| helpers | This directory is reserved for core helper files. Custom and overwritten helpers should be placed in their own subdirectory of the app/helpers or app/subsite/helpers directory. |
| i18n | This directory should be used to place .po language files and custom multi-lingual content files, for example,   * en.po * zh-CN.po |
| i18n/ctn | This directory should contain sub-directories to place custom multi-lingual content files.For example,   * en/ * zh-CN/ |
| i18n/ctn/en | This directory should be used to place custom multi-lingual content files with the directory name extension, for example,   * about-us.en * privacy-policy.en |
| images | This directory contains the images of the application. |
| inc | This directory structure contains the settings and configuration files of the application. The following files are overwritable or inherited by the app/inc or app/subsite/inc directory.   * site.config.php (inherited by /app/inc/site.config.php) * head.php (overwritable by /app/inc/head.php) * 404.php (overwritable by /app/inc/404.php) |
| js | This directory contains the system core javascript files which should not be hacked. Your application javascript files should be placed in /app/js |
| vendors | This directory should be used to place downloaded and custom modules and third party libraries which are common to all sites. |
| xhr | This directory structure contains the files which are requested by AJAX and which are common to all sites. |

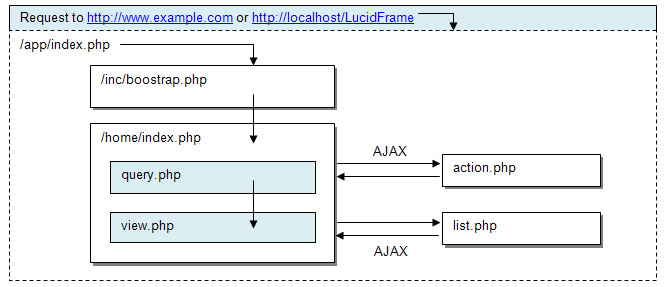
Page Structure

LucidFrame encourages a uniform and structural page organization though it is by no means required. It expects a directory comprised at least two files: **index.php** and **view.php**. As an example, you can see the directory /app/home/ of the LucidFrame release you downloaded.

|  |
| --- |
| /path\_to\_webserver\_document\_root  /LucidFrame  /app  /home  action.php  index.php  query.php  view.php |

1. The **index.php** (required) serves as the front controller for the requested page, initializing the base resources needed to run the page.
2. The **action.php** (optional) handles form submission. It should perform form validation, create, update, delete of data manipulation to database. By default, a form is initiated for AJAX and action.php is automatically invoked if the action attribute is not given in the <form> tag.
3. The **query.php** (optional) should retrieve and process data from database and make it available to view.php.
4. The **view.php** (required) is a visual output representation to user using data provided by query.php. It generally should contain HTML between <body> and </body>.
5. The **list.php** (optional) is a server page requested by AJAX, which retrieves data and renders HTML to the client. It is normally implemented for listing with pagination.

Page Workflow



URL Routing

LucidFrame typically requires mod\_rewrite enabled on web server. It also requires home page routing configured in /inc/config.php. By default, it is set to home which is mapped to the directory /app/home/ and it is accessible via <http://localhost/LucidFrame> or <http://www.example.com>.

|  |  |
| --- | --- |
| 96  97 | # $lc\_homeRouting: Home page routing; if it is not set, default is 'home'  $lc\_homeRouting = 'home'; |

If you have home.php instead of a directory, you could change $lc\_homerouting to home.php.

Accessing URL

You can get the current routing path using a function \_r()or route\_path() and you can get a component of the current path using \_arg().

|  |
| --- |
| // url is www.example.com  echo \_r(); // home  echo \_arg(0); // home  // url is www.example.com/user/1  echo \_r(); // user/1  echo \_arg(0); // user  echo \_arg(1); // 1 |

LucidFrame also provides to use URL component key preceding by a dash (-). For example, <http://www.example.com/posts/-page/1> which reflects [http://www.example.com/posts?page=1](http://www.example.com/listing?page=1)

|  |
| --- |
| // url is www.example.com/posts/-page/1/-sort/title/asc  echo \_r(); // posts/-page/1/-sort/title  echo \_arg(0); // posts  echo \_arg(1); // -page  echo \_arg(2); // 1  echo \_arg(3); // -sort  echo \_arg(4); // title  echo \_arg(5); // asc  // The following is a formal way of getting the URI component ”page”  echo \_arg(‘page’); // 1  // The following is a formal way of getting the URI component ”sort”  \_pr(\_arg(‘sort’)); // array( ‘title’, ‘asc’ )  // \_pr() is an convenience method for print\_r. |

Creating and Getting URL

You can use the function \_url()or route\_url() to make a URL.

|  |
| --- |
| echo \_url(‘user’, array(1));  // http://www.example.com/user/1  echo \_url(‘posts’, array(‘page’=>1, ‘sort’=>array(‘title’,’asc’));  // http://www.example.com/posts/-page/1/-sort/title/asc  echo \_url();  // it would return the current URL |

Redirecting URL

You can use the function \_redirect() to redirect to a URL.

|  |
| --- |
| \_redirect(‘home’);  // redirect to the home page according to $lc\_homeRouting in /inc/config.php  // ‘home’ is a constant whatever you defined for $lc\_homeRouting  \_redirect(‘user’, array(1));  // redirect to http://www.example.com/user/1  \_redirect(‘posts’, array(‘page’=>1, ‘sort’=>array(‘title’,’asc’));  // redirect to http://www.example.com/posts/-page/1/-sort/title/asc  // assuming that the current URL is http://www.example.com/posts/-page/1/-sort/title/asc  // you can redirect to the current page itself by updating the query strings ‘page’ and ‘sort’  // in this case, you can use NULL or an empty string for the first parameter to \_redirect()  \_redirect(NULL, array(‘page’=>2, ‘sort’=>array(‘title’,’desc’));  // redirect to http://www.example.com/posts/-page/2/-sort/title/desc |

Check more details in /helpers/utility\_helper.php and /helpers/route\_helper.php

URL Rewrite

Sometimes, you may need to write RewriteRule in .htaccess of the root directory.

|  |
| --- |
| # www.example.com/en/99/foo-bar to ~/app/post/?lang=en&id=99&slug=foo-bar  # www.example.com/zh-CN/99/foo-bar to ~/app/post/?lang=zh-CN&id=99&slug=foo-bar  RewriteRule ^(([a-z]{2}|[a-z]{2}-[A-Z]{2})/)?([0-9]+)/(.\*)$ app/index.php?lang=$1&id=$3&slug=$4&route=post [NC,L] |

As the default routing name of LucidFrame is “route” and according to the RewriteRule above, route=post will map to the file /app/post/index.php or /app/post.php given the three URI components – lang, id and slug. For example, if the requested URL is [www.example.com/en/99/foo-bar](http://www.example.com/en/99/foo-bar), this will be rewritten to /app/post/index.php?lang=en&id=99&slug=foo-bar or /app/post.php?lang=en&id=99&slug=foo-bar

Core Defined Constants & Variables

The following PHP constants are available across LucidFrame.

|  |  |
| --- | --- |
| PHP Constant | Description |
| APP\_ROOT | File system path to the application’s directory |
| ROOT | File system path to the “root” directory |
| INC | File system path to the “inc” directory of the root directory |
| HELPER | File system path to the “helper” directory of the root directory |
| I18N | File system path to the “i18n” directory of the root directory |
| VENDOR | File system path to the “vendors” directory of the root directory |
| BUSINESS | File system path to the “business” directory of the root directory |
| FILE | File system path to the “files” directory of the root directory |
| CACHE | File system path to the “cache” directory under the “files” directory |
| LC\_NAMESPACE | Namespace according to the site directories, for example, if you have www.example.com/admin, you may have a namespace “admin”. |
| WEB\_ROOT | URL to the application root, e.g., http://www.example.com or http://localhost/LucidFrame/ |
| CSS | Web-accessible path to the “css” directory., e.g., http://www.example.com/css/ |
| JS | Web-accessible path to the “css” directory., e.g., http://www.example.com/js/ |
| WEB\_VENDOR | Web-accessible path to the “vendors” directory., e.g., http://www.example.com/vendors/ |
| HOME | The home page URL |

LucidFrame has a global object variable in Javascript – LC. The following Javascript global variables of LC are available to use.

|  |  |
| --- | --- |
| Javascript Variables | Description |
| LC.root | URL to the application root. It could also be accessible as WEB\_ROOT |
| LC.self | The current URL |
| LC.lang | The current language code |
| LC.baseURL | The sub-directory name if your application is wrapped in. It would be blank if your application is located in the web server document root. |
| LC.route | The current route path |
| LC.namespace | Namespace according to the site directories, for example, if you have www.example.com/admin, you may have a namespace “admin”. |

You can also extend any global variable from LC by using a hook \_\_script() in /app/helpers/utility\_helper.php

|  |
| --- |
| /\*\*  \* This function is a hook to the core utility function \_script()  \*/  function \_\_script(){  global $lc\_cleanURL;  ?>  LC.cleanURL = <?php echo (int)$lc\_cleanURL; ?>;  <?php  } |

Database Setting Configuration

You can configure your database setting in /inc/config.php.

|  |  |
| --- | --- |
| 71  72  73  74  75  76  78  79  80  81  82  83 | # $lc\_databases: The array specifies the database connection  $lc\_databases = array(  'default' => array( // default database; you could also have other database settings here  'engine' => 'mysqli', // database engine  'host' => 'localhost', // database host  'database' => '', // database name  'username' => '', // database username  'password' => '', // database password  'prefix' => '', // table name prefix  'collation' => 'utf8\_general\_ci' // database collation  )  ); |

Forms

You can implement a form in two ways – using AJAX and without using AJAX. LucidFrame provides AJAX form submission by default. You can prevent AJAX form submission using class=”no-ajax” in the <form> tag.

Creating Form Using AJAX

To be continued…..