Carte Blanche

- Nos Activités - Nos projets -

Date de mise en ligne : mercredi 13 mars 2013

Les Ateliers Pierrot

The MVC of Les Ateliers, simple, light-weight and easy to use for developments.

See http://github.com/php-carteblanche/....

[sommaire]

Presentation

CarteBlanche is a PHP framework constructed as an MVC application ready to work. It uses a core library of classes that are enough to make it work and allows to develop some extra features quickly in this environement.

It holds `bundles`, some packages to execute a feature, that could be exported after development as a standalone PHP package (if so) ; it also proposes a set of `tools`, pretty simple PHP classes, one per feature, to help building views.

The rendering is integrated in a ready-to-use (but very simple) global template coded in HTML5.

Installation & Usage

Installation

The package to install is carteblanche/carteblanche.

Once you have downloaded or cloned the package sources, the installation is quite simple. It requires Composer to install its dependencies. Please see the "Dependencies" section below for more informations about Composer, what it is and what it does.

Installation steps :

1. put the whole CarteBlanche directory in your web-server files system ; you can download an archive at https://github.com/php-carteblanche... or clone GIT sources running in a command line terminal :

```
 ~$ git clone --recursive https://github.com/php-carteblanche/carteblanche path/to/CarteBlanche
// or, if you have an old version of GIT without the 'recursive' option:
~$ git clone https://github.com/php-carteblanche/carteblanche path/to/CarteBlanche
~$ cd path/to/CarteBlanche
~$ git submodule update --init
```

2. go to the CarteBlanche root directory :

```
~$ cd path/to/CarteBlanche
```

3. install dependencies running the `bin/build.sh` script :

```
~$ ./bin/build.sh install
// or, if you need to configure Composer or PHP binaries to use:
~$ ./bin/build.sh -h
```

4. CarteBlanche is ready !

Usage

As in many other frameworks, the PHP and binary sources are separated from the assets, web accessible stuffs. Basically, all the web files (javascripts, css, images and other assets) are stored in the `www/` sub-directory. All other files may NOT be web-accessible. The PHP sources are mostly stored in the `src/` sub-directory.

To access the framework's controller from a web-browser, just load the `index.php` file from the `www/` directory of CarteBlanche. A development version of this controller is predefined and accessible loading the `dev.php` file instead of `index.php`.

A terminal controller is also defined, for CLI use only, that you can access running the command :

```
~$ php bin/console
```

Documentation

See the `doc/` directory contents for documentations about configuration, structure and usage.

Rules & Dependencies

Coding Rules

To build a comprehensive and maintainable code, we try to follow the coding standards and naming rules most commonly in use :

- the PEAR coding standards
- the PHP Framework Interoperability Group standards.

Knowing that, all classes of the framework are named and organized in an architecture to follow the usage of the standard SplClassLoader.

Dependencies

As explained in the "Installation" section above, this package uses some other third-party packages to be compliant with the common standards of actual PHP development. In addition to Composer, required for its classic installation, CarteBlanche proposes a set of other external PHP packages used for development and a set of external assets packages used for the HTML5 rendering.

To install all PHP packages for development, just run :

```
 ~$ ./bin/build.sh --set=dev install
// or, using Composer itself
~$ composer install --dev
```

A PHP documentation tool is available generating it with Sami. To build or re-build it, you will need to run Composer in development environment (*just as shown above*) and then run :

```
 ~$ ./bin/build.sh render-doc
// or, using Sami itself
~$ php src/vendor/sami/sami/sami.php render config/vendor/sami.config.php
```