PACE 2.x - An Overview

Dan Trickey

2017

Contents

1 Introduction

PACE 2.x is a complete rewrite of the system and is not compatible with the old system. It has been written with future upgrades in consideration and is significantly better commented. I also have spent considerably more time on the database design to ensure reliability.

2 Technical Details

PACE 2.x is written in PHP and is based on the Laravel framework. The current framework version can be found in the Settings menu in the administrator area.

2.1 System Requirements

At the time of writing, the system requires the following:

- Minimum 512MB of RAM (with swap) . Recommended 1GB+.
- A *nix based system with Apache and MySQL server. Ubuntu Server LTS Edition is best.
- SFTP Upload for secure uploading of data.
- xGB of free space.
- PHP. Validated for 5.6.x, 7.0.x, 7.1.x.
- Composer Package manager
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension
- An email service (i.e SMTP, mailgun)

2.2 Prior Knowledge

In order to contribute to the system, you will need to have knowledge of PHP, or at the very least any C based language. You will need to understand object orientated programming, MVC (Model View Controller) frameworks, namespacing and relational databases. It will also be helpful if you have knowledge of unit testing (and the reasons for it), JSON-based APIs (for development of apps), REST, the HTTP protocol and common request types, GNU/Linux (for management), bash scripts, cron jobs, javascript, salting and hashing of passwords (bcrypt is used here). You will also need knowledge of Git to be able to contribute to the repository.

2.3 Development Requirements

If you wish to develop for the system, the following is required:

- PHP 5.6.x+, including the CLI extensions

- A basic webserver with MySQL (i.e XAMPP for Windows, LAMP stack, WAMP etc)

- Composer package manager

- A text editor (i.e gedit, emacs, notepad, notepad++)

- Git

The following is not required, but is recommended:

- Vagrant, with the laravel homestead box.

- The above utilises virtualisation, so Intel VT-d extensions or AMD-V extensions.

- GNU/Linux (The best OS). I recommend something based on debian. Ubuntu will do.

- A proper PHP IDE. PHPStorm is excellent and can be obtained for free if you present academic ID.

- A text editor with syntax highlighting, for example (Atom, Sublime text, notepad++)

- A GUI for Git. There are several options for this, but I prefer GitKraken.

2.4 Configuration

Most system configuration is done through the .env file:

```
1   APP_ENV=local
2   APP_KEY=
3   APP_DEBUG=true
4   APP_LOG_LEVEL=debug
5   APP_URL=http://localhost
6
7   DB_CONNECTION=mysql
8   DB_HOST=127.0.0.1
9   DB_PORT=3306
10  DB_DATABASE=homestead
11  DB_USERNAME=homestead
12  DB_PASSWORD=secret
13
14  BROADCAST_DRIVER=log
15  CACHE_DRIVER=file
```

```
16   SESSION_DRIVER=file
17   QUEUE_DRIVER=sync
18
19   REDIS_HOST=127.0.0.1
20   REDIS_PASSWORD=null
21   REDIS_PORT=6379
22
23   MAIL_DRIVER=smtp
24   MAIL_HOST=mailtrap.io
25   MAIL_PORT=2525
26   MAIL_USERNAME=null
27   MAIL_PASSWORD=null
28   MAIL_ENCRYPTION=null
29
30   PUSHER_APP_ID=
31   PUSHER_APP_KEY=
32   PUSHER_APP_SECRET=
```

3 Operation

This section will be completed later.

4 Updating

The project is now hosted on Github at https://github.com/trickeydan/Pace It is thus open source under the GPLv3 licence and anybody can contribute to it. I am the maintainer of the repository, meaning that all changes will have to be checked and approved by myself. All Pull Requests are subject to automatic testing using Travis CI and any PRs that fail to build will not be considered. Any new components must also use unit testing to ensure future reliability. The code on the server is automatically updated using a webhook on the repository. Thus, no direct changes can be made to the server files. Any changes will be automatically overwritten next time the repository is updated on the MASTER branch.

5 Future Support

This section will be completed later.