



Website: <https://codekerala.com>

Email: codekerala@gmail.com

Codecanyon: <https://codecanyon.net/user/codekerala/portfolio>

Github: <http://github.com/codekerala>

Youbute: <https://www.youtube.com/c/codekerala>

© 2019 Code Kerala

This source code is protected by International Copyright laws. Re-distribution of the source code without written permission of the owner is prohibited.

Information

For support email us at: codekerala@gmail.com

Getting Started

Follow the instruction below to setup Meridian in SAAS or Standalone mode.

System Requirements

- Linux or Mac Operating System
- Nginx / Apache
- MYSQL Version 5.7 or above
- PHP ≥ 5.4 , PHP < 7
- Mcrypt PHP Extension
- OpenSSL PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension

As of PHP 5.5, some OS distributions may require you to manually install the PHP JSON extension. When using Ubuntu, this can be done via apt-get install php5-json.

Meridian utilizes **Composer** to manage its dependencies. So, before using Meridian, make sure you have Composer installed on your machine.

Installation and Configuration

Public Directory

After installing Meridian, you should configure your web server's document / web root to be the **public** directory. The index.php in this directory serves as the front controller for all HTTP requests entering your application.

Directory Permissions

Directories within the storage and the **bootstrap/cache** directories should be writable by your web server or Meridian will not run.

We can give group ownership of our Meridian directory structure to the web group by typing:

```
sudo chown -R :www-data ./meridian
sudo chmod -R 775 ./meridian/storage
sudo chmod -R 775 ./meridian/bootstrap/cache
```

Mysql Configuration

Disable SQL mode ONLY_FULL_GROUP_BY in Mysql.

Step1: Run the command **SELECT @@sql_mode;** in you mysql client.

Step 2: Edit MySQL config file **my.conf** . Usually found in /etc/my.cnf or /etc/mysql/my.conf . If not sure, below command to find the file.

```
mysql --help | grep my.cnf
```

Step 3: Copy sql_mode values, remove ONLY_FULL_GROUP_BY and put the line in my.conf under [mysqld] section.

In my case it looks like this:

```
[mysqld]
sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
```

Restart the mysql server

```
sudo /etc/init.d/mysql restart
```

Application Configuration

Next create a new database and add your database credentials to your .env (copy content from .env.example) file:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret

DB_TEAM_HOST=127.0.0.1
DB_TEAM_PORT=3306
DB_TEAM_USERNAME=homestead
DB_TEAM_PASSWORD=secret
```

You will also want to update your website URL inside of the APP_URL and APP_NAME variable inside the .env file:

```
APP_NAME=Meridian
APP_URL=http://meridian.test
```

Production Settings

```
APP_ENV=production
APP_DEBUG=false
```

Email

Also setup your email server config in .env to send emails.

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

Use service like [Sparkpost](#) to sent email for free.

Follow the steps to configure sparkpost for sending emails.

Step 1: Complete the your email and domain verification in sparkpost dashboard.

Step 2: Change the following inside .env file.

```
MAIL_DRIVER=sparkpost
SPARKPOST_SECRET=<ecret key from sparkpost dashboard>
MAIL_FROM_ADDRESS="<example@your-company.com>"
MAIL_FROM_NAME="<Example Company Name>"
```

You may also want to configure a few additional components of Laravel, such as: Cache, Session, etc.

Stripe and Stripe Connect

You have to create Stripe Connect account in order for tenants to receive payments.

Open .env file and update the following:-

```
STRIPE_KEY=  
STRIPE_SECRET=  
  
STRIPE_CONNECT_KEY=  
  
STRIPE_WEBHOOK_SECRET=
```

Stripe can notify your application of a variety of events via webhooks.

Add the following to your stripe dashboard [<https://your-website.com>/meri/stripe/webhook](https://your-website.com/meri/stripe/webhook)

Run The Installer

Composer

Run `composer install`

SAAS Mode

Lastly, we can install meridian by running the below.

```
php artisan meridian:install
```

Standalone Mode

Lastly, we can install meridian by running the below.

```
php artisan meridian:install --standalone
```

```
Email: user@meridian.test
```

```
Password: password
```

The above command will also run `php artisan key:generate --force` and `php artisan migrate --force`

To access admin by default you should be able to login with the below email and password

```
Email: admin@meridian.test
```

```
Password: password
```

Cron Job

In order to make recurring invoices and exports work, you have to setup up cron job. Go to your terminal and run the below command

```
crontab -e
```

This will open server crontab file, paste this code inside, save it and exit.

```
* * * * * php /path-to-your-project/artisan schedule:run >> /dev/null 2>&1
```

This Cron will call the Meridian command scheduler every minute. When the `schedule:run` command is executed. Meridian will evaluate your scheduled tasks and runs the tasks that are due.

Pretty URLs

Apache

Meridian includes a `public/.htaccess` file that is used to provide URLs without the `index.php` front controller in the path. Before serving Meridian with Apache, be sure to enable the `mod_rewrite` module so the `.htaccess` file will be honored by the server.

If the `.htaccess` file that ships with Meridian does not work with your Apache installation, try this alternative:

```
Options +FollowSymLinks

RewriteEngine On


RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^ index.php [L]
```

Nginx

If you are using Nginx, the following directive in your site configuration will direct all requests to the `index.php` front controller:

```
location / {
    try_files $uri $uri/ /index.php?$query_string;
}
```

For developers

Meridian is a build with laravel 5.7 and Vue.js 2.6

This zip file only contains production files and does not contain `node_modules` directory. In order to customize you have to delete the `vendor` directory and run the following commands:

```
composer install
npm install
```

Directory Structure

Javascript files can be found in `resources/assets/js` and not in the `public` folder.

Internationalization

You can find the english translation inside `resources/lang/en`

For example to translate to language french fr

Set `APP_LOCALE=fr` inside `.env`

Add fr to langs array in `config/meridian.php`

```
'langs' => [  
    'en' => 'English',  
    'fr' => 'French'  
],
```

Make a folder fr inside `resources/lang/`

copy all the files from `resources/lang/en` to `resources/lang/fr`

Begin translation

Run `php artisan meridian:lang` to generate translation files

Customization

You can contact us at `codekerala@gmail.com` for support and paid customization requests.