ICO DASH
INSTALLATION DOCUMENT

Setting up server for installation of source code

Recommended Requirements: Ubuntu 16.0 PHP 7.2^ MYSQL 5.7.0^ and Composer

Since different hosting companies have different methods of installing the OS and the dependencies, please follow your hosting providers instructions to set up the above mentioned **Recommended requirements.**

Install PHP:

Here we will show how to install php 7.2 version (you can install php7.2 or above) because our script is developed on laravel 5.6v

sudo apt-get update && apt-get upgrade
sudo apt-get install python-software-properties
sudo apt-get-repository ppa:ondrej/php
sudo apt-get update
sudo apt-get install php7.2

To check if PHP 7.2 is iinstalled on your server, run the following command

php -v

Now install php and apache2 modules, run the following commands

```
apt-get install php-pear php7.2-curl php7.2-dev php7.2-gd php7.2-
mbstring php7.2-zip php7.2-xml php-mcrypt php7.2-common
```

Install Apache2 Server :

First you have to install apache server which will serve the app from browser. Run the below Command,

sudo apt-get update && apt-get upgrade
sudo apt-get install apache2 libapache2-mod-php7.2 php7.2-fpm

After apache installed successfully, you can check via command :

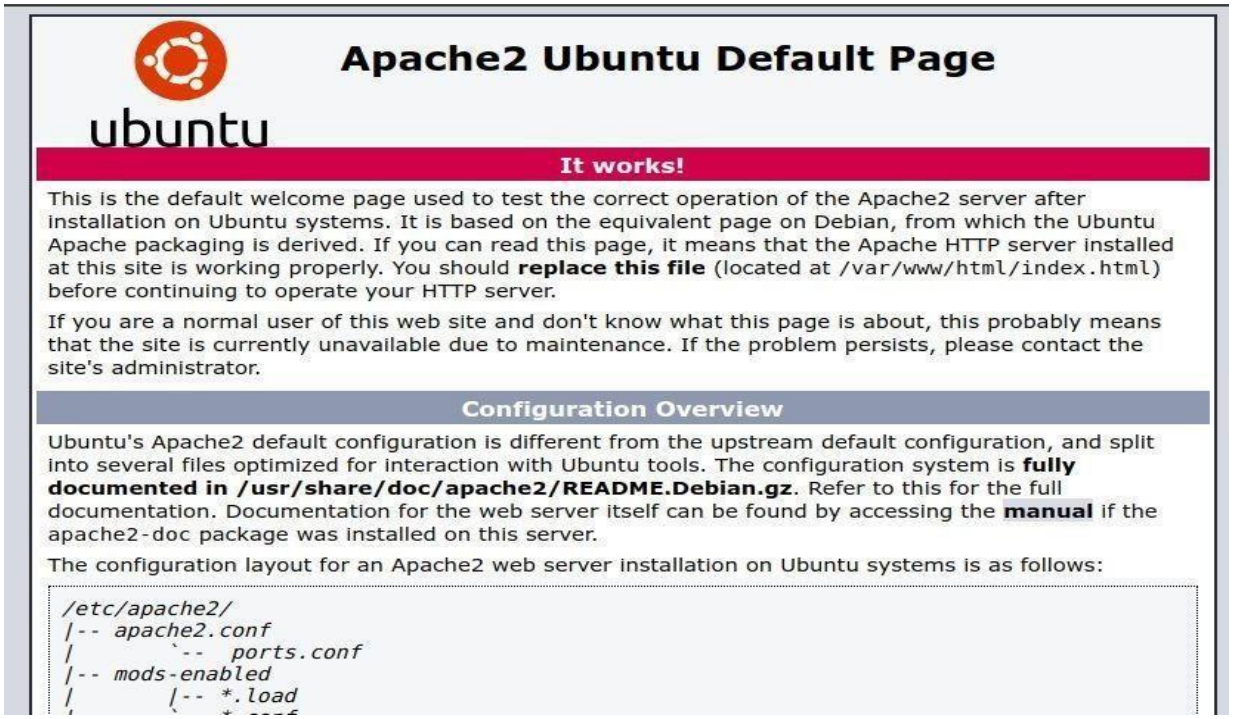
apache2 -v

For more information visit : https://tutorials.ubuntu.com/tutorial/install-and-configure-apache#1

Now you need to enable apache2 rewrite module so that laravel pretty url works.

sudo a2enmod rewrite
sudo service apache2 restart

Now type your server ip on browser and hit enter, You will see something like this page.



Install Mysql DB Server and phpmyadmin:

Install mysql-server packages for MySQL database. Also, install php-mysql package to use MySQL support php and also phpmyadmin to access mysql db from browser. Use the following command to install.

sudo apt-get install mysql-server php7.2-
mysql sudo mysql_secure_installation sudo
apt-get install phpmyadmin

Now restart the system apache2 and mysql services

sudo systemctl restart apache2.service

sudo systemctl restart mysql.service

For more information visit link :

https://tecadmin.net/install-apache-mysql-php-lamp-stack-on-ubuntu-16-04/

Now navigate to phpmyadmin via browser :

http://{your-server-ip}/phpmyadmin

Login in phpmyadmin with your username and password that you setup while installing. Create empty database. If you don't know, then google how to create new databse using phpmyadmin.

Install Composer:

Follow the the link to install composer
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-composer-on-ubuntu-16-04

UPLOADING THE SOURCE CODE PACKAGE

- Unzip the contents of the Source Code Zip package in a new folder, for example, let's call it **icodash.**

Assuming you want to install it under the root domain.

Now, use an FTP program such as FileZilla to upload the contents of the **icodash** folder under the **/var/www/html** public folder.

Now navigate to the **/var/www/html/icodash** folder in terminal and update the laravel dependencies, run the following command

sudo composer update

Or if above command at end gives error then try,

sudo composer update --no-scripts

*give 777 permission for **storage** and **bootstrap** folder
* give 755 permission to the project folder(**icodash** folder)

Open the .env file under:

/icodash/.env
[Change the DB connection settings, mail, Google login settings,]

After this run following commands

sudo php artisan migrate
sudo php artisan db:seed
sudo php artisan storage:link
php artisan passport:install --force

Now to run the app from browser, you need to point you {server-ip} to **/var/www/html/icodash/public** folder, to do this follow this steps

Open in editor :
/etc/apache2/sites-available/000-default.conf file

DocumentRoot /var/www/html >> change this to
DocumentRoot /var/www/html/icodash/public

In /etc/apache2/apache2.conf edit
<Directory /var/www/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>

Change AllowOverride None >> **AllowOverride All**

Now restart the apache2 server, run following command :
sudo apache2 reload
sudo apache2 restart

***now you can navigate you browser with {server-ip} to check website is working or not .**

Now setup your wallet in admin panel :

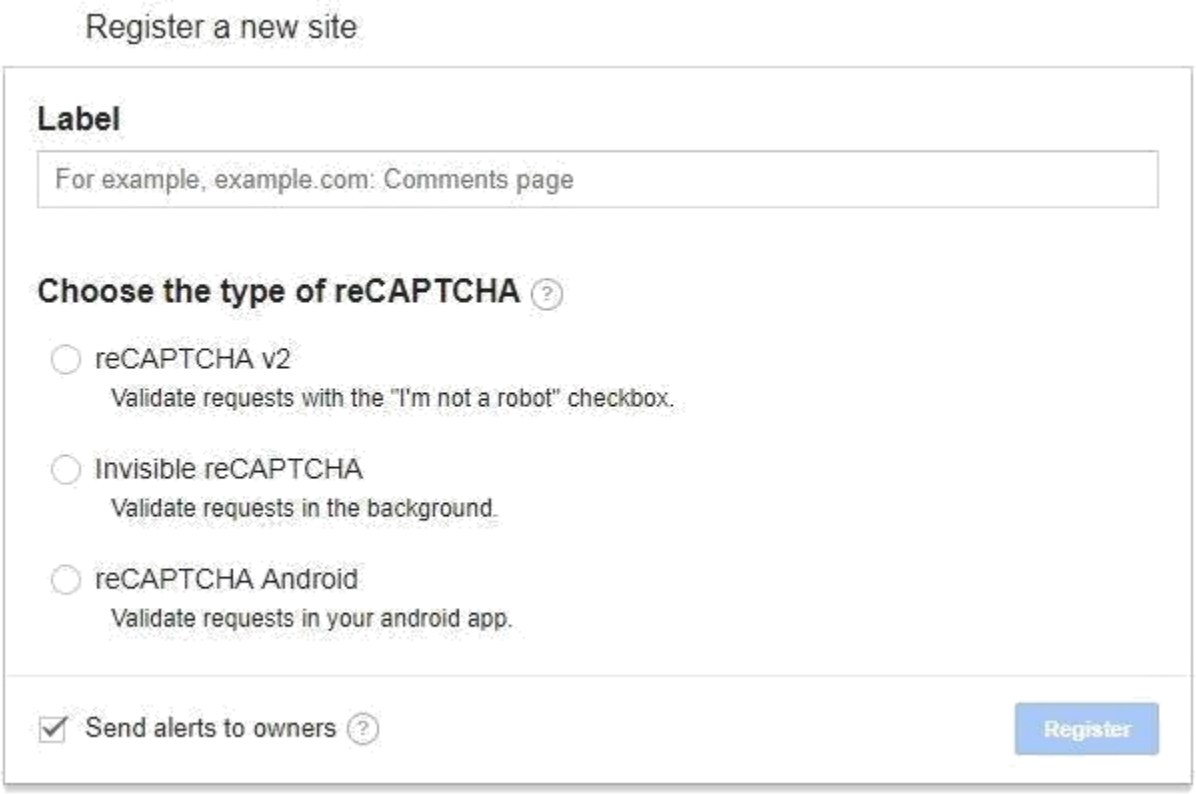
1) Navigate to admin panel {server-ip}/admin/login Default admin user : admin@mailinator.com Password : 123456
2) Goto account page by clicking left side menu
3) Then in wallet section setup following fields
   Transaction hash,
   Contract Address,
   Contract name,
   Contact ABI

Setting up the Basics

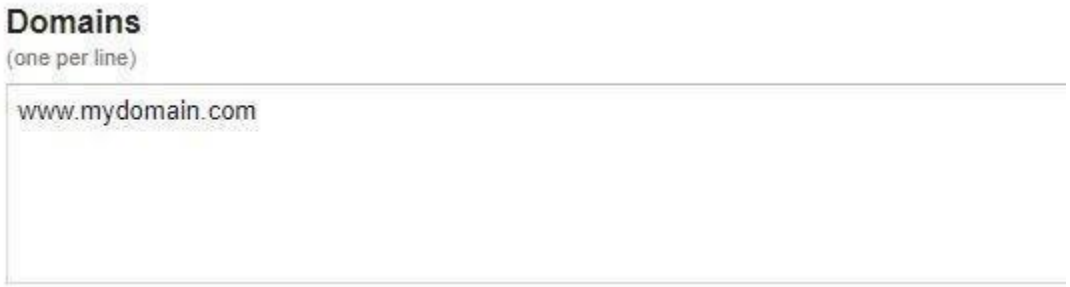
Google Captcha Key in the Admin panel

Signup for the Google Recaptcha key and paste it in the admin panel

https://www.google.com/recaptcha/intro/v3beta.html



Choose **reCaptcha v2** and then add the domains you would want to enable reCaptcha functionality. It could be domain or an ip address (provided it is a static one, do not choose this option on a Shared hosting as the ip keeps changing regularly.



Now select the API key for the client side.

URL: https://www.google.com/recaptcha/api/siteverify

secret (required)	[redacted]
response (required)	The value of 'g-recaptcha-response'.
remoteip	The end user's ip address.

Copy the Captcha key, Private and Public keys; paste them in the account setting page.

HOW TO USE THE ICO DASHBOARD ADMIN

SENDING COINS TO YOUR USERS

Step 1: Login to your Dashboard / Coin Requests page,

In the table you will see list of people who have purchased the coins in different currencies.

It will show Name of the person, the Amount they have paid and the number of coins they want, the status (Pending or Coins sent) and the action.

If the coin request is new and you haven't sent any coins to the user it will show as pending and the button in the actions tab will show as **SEND COINS**

If you have sent the users some coin, it will show as **Coin Sent** in the status and under the Actions tab, it will display **View Details** where you can click and see the details of the completed transaction.

To send coins, click on **SEND COINS** under the actions tab. It will open the transaction details page.

Coin Requests

You can view and send coins to people who have purchased the coins here.

- All your Coin requests are listed here. In the Status, if it shows as **PENDING** it means that people have paid and are waiting for you to transfer the coins to them.

- See how much they paid and transfer the exact amount of coins they paid for.

- You would need to **install MetaMask** for sending coins to those who have paid.

- To send coins to users, click on **SEND COINS** under the Action tab.

Balance : **99999901000000000000000000**

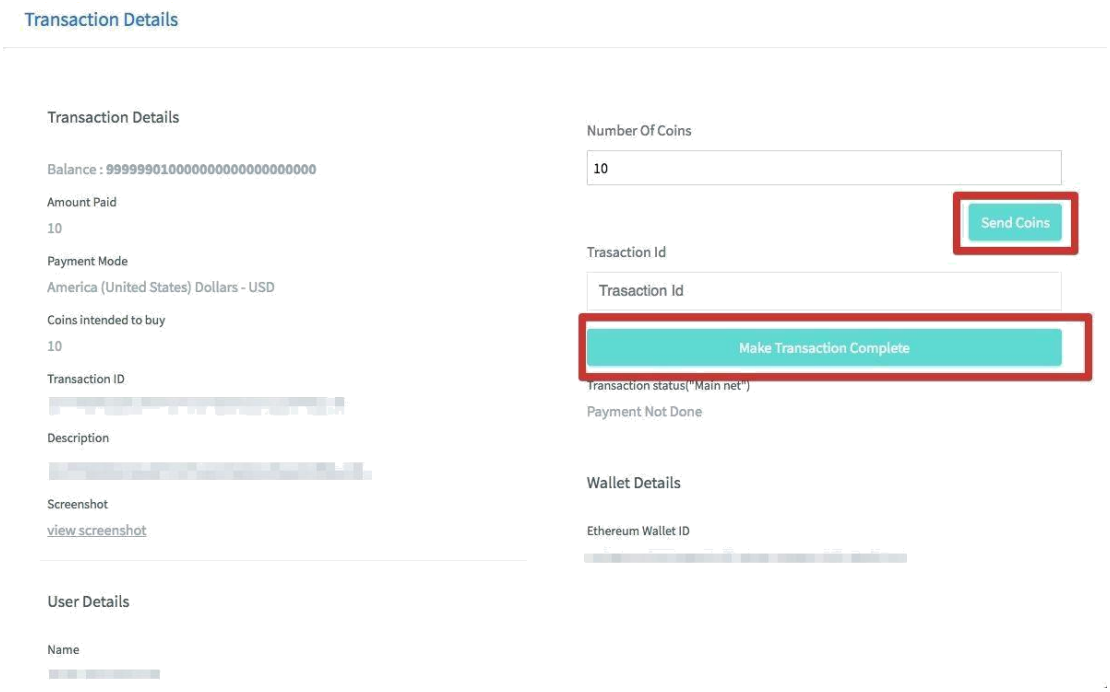
Download Transactions

Copy CSV Excel PDF Print

Search:

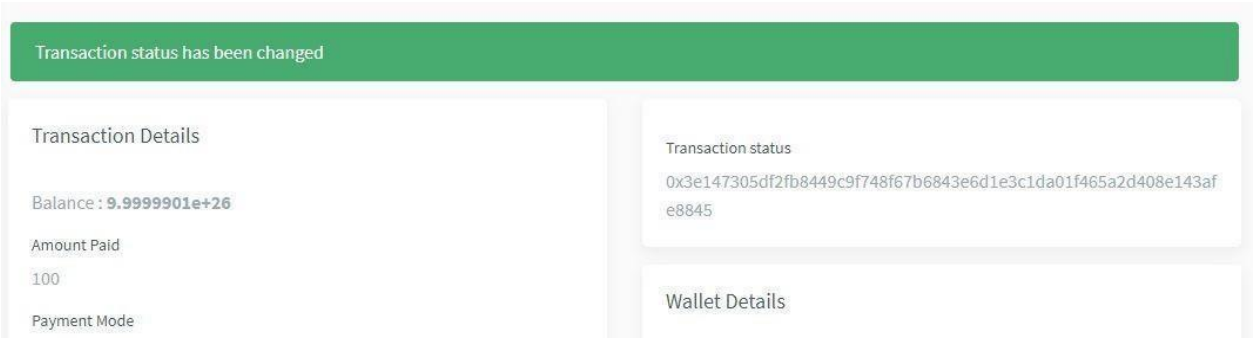
#	Name	Amount Sent	Equivalent	Status	Action
1	[redacted]	100	100 My Coin	Pending	Send Coins
2	[redacted]	100	1000 My Coin	Pending	Send Coins
3	[redacted]	750 America (United States) Dollars - USD	26354 My Coin	Pending	Send Coins

Now.before you send Coins to the users who have paid, you will have to first verify in your bank / crypto wallet account that the transaction details entered by the customer is true.



Then click on the **Send.**
This popup will show up with all the details of the Ether wallet the user has entered. As an admin you don't need to enter it manually.

This will show as Completed transaction on the User Side.



ICO STATUS BAR

The statuses you add here are displayed on the User dashboard. You can display your Project goal, Roadmap, the ICO status or any event

To add a Status,

- Enter the Title, Hints, and a date for the goal to complete.
- Click **Add status**

The ICO status you entered will be saved. Once it is completed, **select Mark as Achieved** to display as status Complete.

NOTIFICATIONS

Send Notifications to your user on the Dashboard. When you send a notification, your users will the message once they login to the dashboard.

To set a notification, go to the **Notifications** tab on the left menu.

- Enter the title of the notification (Example: Announcement)
- Then, enter the notification message. (Example: Pre-ICO sales offer coming up soon) Click **SEND** to display the notification immediately.

You can later remove the notifications from the Notification history section of the notifications page.

Account Page

Your Accounts Page is where all the site details, Wallet addresses, your token's value, Google Recaptcha etc can be changed.

Here you can also change the ETH contract address. Please do not change this without expert help, it may stop the site from functioning properly.

Footer Menu Link

In the Footer Menu Link page, you can add links in the footer of the user Dashboard.

USER DASHBOARD

How to Buy Coins.

As a User, choose a payment method that you want to use for paying and purchasing the Tokens from the admin.

Once you have made the payment, On the Dashboard, click on **Enter Transaction Details** And enter the details of your transaction.

You would need to have an Ether wallet to receive the coins as a user.

Once the admin verifies the transaction, the user will receive the admin panel (As an admin, see how to send coins above)

PayPal Payment Configuration

```php
<?php
return array(
/** set your paypal credential **/
'client_id' =>'your_paypal_client_id',
'secret' => 'your_paypal_secret',
/**
* SDK configuration
*/
'settings' => array(
/**
* Available option 'sandbox' or 'live'
*/
'mode' => 'sandbox',
/**
* Specify the max request time in seconds
*/
'http.ConnectionTimeOut' => 30,
/**
* Whether want to log to a file
*/
'log.LogEnabled' => true,
/**
* Specify the file that want to write on
*/
'log.FileName' => storage_path() . '/logs/paypal.log',
/**
* Available option 'FINE', 'INFO', 'WARN' or 'ERROR'
*
* Logging is most verbose in the 'FINE' level and decreases as you
* proceed towards ERROR
*/
'log.LogLevel' => 'FINE'
),
);
```

Add **CLIENT_ID** and **SECRET** in file ***=> /Config/Paypal.php***

TWILIO SMS INTEGRATION:

```
TWILIO_ACCOUNT_SID=your_twilio_sid
TWILIO_AUTH_TOKEN=your_twilio_auth_token
TWILIO_PHONE_NUMBER=your_twilio_phone_number
```

Add **TWILIO_ACCOUNT_SID**, **TWILIO_AUTH_TOKEN, TWILIO_PHONE_NUMBER** in .env file

MAILGUN EMAIL INTEGRATION:

Add **MAIL_USERNAME, MAIL_PASSWORD, MAILGUN_DOMAIN, MAILGUN_SECRET in**

. env File

```
MAIL_DRIVER=mailgun
MAIL_HOST=smtp.mailgun.org
MAIL_PORT=587
MAIL_USERNAME=mail_username
MAIL_PASSWORD=mail_password
MAIL_ENCRYPTION=tls

MAILGUN_DOMAIN=your_mailgun_domain
MAILGUN_SECRET=your_mailgun_secret_key
```

Add ***Mailgun*** : **domain** and **secret** in file **=> */Config/services.php***

```
'mailgun' => [
    'domain' => 'your_mailgun_domain',
    'secret' => 'your_mailgun_secret_key',
],
```

Update **BASE_URL** and **ERC-20 Token API URL** :

File => **App/Http/Controllers/Basecontroller.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Admin;

define('BASE_URL', 'your_domain_url');
define('API_TOKEN_URL', 'Api_url_of_ERC20');
```

Setup cron:

Next, we need to setup cron for campaign scheduling. Enter following command in your terminal:

crontab –e

* *** * * * cd /var/www/html/ico-dashboard-laravel/ && php artisan schedule:run >> /dev/null**

2>&1 Exit the file by pressing Control + X.

Then, restart cron by using the below

command: Sudo service cron restart