

Developer Basic Documentation

Main Work Flow:

- 1) User Visit sitename.com/folder/site/register
- 2) He fills up the form and submit.
- 3) Site.php controller (register method) processes the data and submit.
- 4) On completion of successful registration earning.php model process the instant payout (like level direct).
- 5) User gets his dashboard and that's all.

Important Files:

Controller/Cron.php => This file is very important as this is the door to run all the automated processes. It updates total node count of member, generate ROI and payouts and updates member hierarchical genealogy detail. However, this controller is set to run on completion of successful registration, but it is never advisable to run after each signup in case of a huge database. It is always advisable to run as a get/wget cron. Eg:

wget -q <http://sitename.com/folder/cron>

Model/Custom_income.php => This is another one important model. If you have a custom income rather than normal, matching/level/reward/roi or diirect income or anything else .. you can program here. One thing is to note that whatever method you want to call you have to initialize inside the index method. Suppose you want to generate board income. See below the example.

```
class Custom_income extends CI_Model
{
    public function __construct()
    {
        parent::__construct();
    }
    public function index()
    {
        $this->board_income();
    }
    private function board_income()
    {
        // Your Codes Here
    }
}
```

Some Functions:

We have our few functions that we use to ease our work, These may sometime save your few seconds.

DATABASE:

- 1) **`$this->db_model->select('field_name', 'table_name', 'where_condition as array');`**
eg: `echo $this->db_model->select('name', 'member', array('id'=>'1001'));` /// return John;
- 2) **`$this->db_model->select_multi('field_names', 'table_name', 'where_condition as array');`**
eg: `$data = $this->db_model->select_multi('name, phone', 'member', array('id'=>'1001'));` /// return array;
`// echo $data->name ==> John; echo $data->phone ==> 1234567890`
- 3) **`$this->db_model->sum('field_name', 'table_name', 'where_condition as array');`**
eg: `echo $this->db_model->sum('amount', 'earning', array('id'=>'1001'));` /// return sum eg: 100;
- 4) **`$this->db_model->count_all('table_name', 'where_condition as array');`**
eg: `echo $this->db_model->count_all('member', array('1'=>'1'));` /// return tota db count eg: 50;

COMMON:

To Send Email:

`$this->common_model->email(to_emailid, subject, html_msg);`

To Send SMS:

`$this->common_model->sms(to_phoneno, text_msg);`

Optionally you may post any post data to curl using below method:

`$post = key=value&key2=value2&key3=value3;`
`$this->common_model->sms(to_phoneno, text_msg, $post);`

To Call a URL via Curl:

`$this->common_model->curl($url, $post);`

Note: If Curl fails, it will fall to php file `get_contents()` automatically.

To Filter User Id (\$_GET, \$_POST, \$_REQUEST):

`$this->common_model->filter($_REQUEST['userid']);`

To Filter and Convert to a Float number (\$_GET, \$_POST, \$_REQUEST):

```
$this->common_model->filter($_REQUEST['data'], 'float');
```

To Filter and Convert to a Decimal number (\$_GET, \$_POST, \$_REQUEST):

```
$this->common_model->filter($_REQUEST['data'], 'decimal');
```

To Get Logged In Admin User ID:

```
$this->session->admin_id;
```

To Get Logged In Member User ID:

```
$this->session->user_id;
```

To Get Logged In Franchisee User ID:

```
$this->session->fran_id;
```

CONFIGURATION VARIABLE:

If you need to set more configuration variable, set it at “**config/global.php**” file.

How to Stop all income and Set Custom Income ?

Ans: If you are confused or want to stop all income and set your own income do the following steps.

STEP 1: at config/global.php set `$config['fix_income'] = "Yes";`

STEP 2: program your income logic at **Model/Custom_income.php** as described at top of the document.

STEP 3: run `http://sitename.com/folder/cron` as wget or curl at desired interval. (cron is a controller and url route).

How to Customize Existing Binary , ROI or Other Income Logic ?

Ans: You cannot change existing binary, ROI or Other Income Logic, but you may disable all of them and set your new binary and other custom income logic as mentioned above.

How to Show Additional Income Names at earning view ?

Ans: To do that thing go to config/config.php and insert additional income name as associative array. Eg:

```
$config['income_name'] = array(

    "Matching Income" => "Binary Income",
    "Referral Income" => "Referral Income",
    "Level Income"    => "Level Income",
    // Key => Value
);
```

Where key is Income name stores in database and value is income name display in UI.

How to Show Configure Matching Income Ratio ?

Ans: To do that go to config/config.php Line No 12, 13.

How to Customize Mobile Recharge API ?

Ans: To customize mobile recharge API or anything related, please go to views/member/recharge/new_recharge.php and design the desired recharge form with operator detail and change the controllers/recharge.php (recharge method) as per your own API documentation.

How to Customize Existing Theme ?

Ans: To customize existing theme please follow the below steps:

ADMIN: Go to views/admin/base.php and design your own theme.

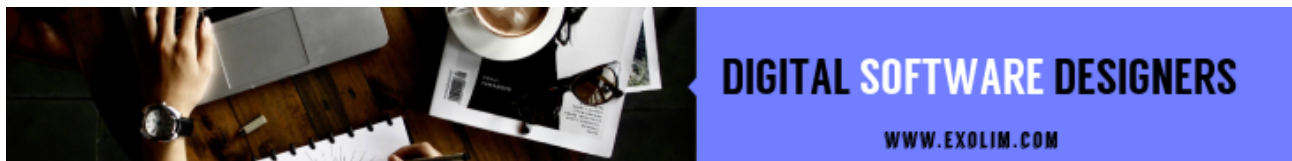
MEMBER: Go to views/member/base.php and design your own theme.

ADMIN: Go to views/franchisee/base.php and design your own theme.

LOGIN AND REGISTER FORM: Go to views/theme/default/base.php and design your own theme.

NOTE: Just remember that while changing themes, keep the href links and image links same.

As our software comes with nearly open source format (Other than few important files that is encoded for security) you can rest customize it as per your need. Our software follows loose MVC Pattern.



www.exolim.com