

# VOBO CLOUD - SASS

Now the era of writing code for classic "Corporate Site" and similar jobs is over

"Vobo Cloud " is powered by the " Fix Framework " developed specifically for the sass backbone structure. It is used quickly and functionally thanks to its easy-to-access encoded infrastructure in plain language.

With the sass feature, you can define individual plugins for each site that opens and manage components as you wish. For registered sites, the registration limit is set separately.

With the developed plug-in infrastructure, fully integrated plug-ins can be developed on the system.

It has a positive effect on users with its easy-to-use interface and operations.

Api documentation is automatically prepared for the created components, it creates a sample code fragment, data connection, and process links in the created document.

## Form Elements

1. Text input
2. Tags input
3. Select - Option
4. Radio – Option
5. Checkbox - Option
6. Number input
7. Photo manager
8. Text editor
9. Color select
10. Multi Select – Option
11. Textbox
12. Input mask
13. Random
14. Date

## Project Requirements

- 1) Os : Centos 7
  1. Ram : 2 Gb
  2. Cpu : 4 Core
  3. Disk : 20 Gb
  4. Network : 100Mbit
- 2) Engine : Nginx & Mysql & Php
- 3) Technical information little. for install

## INSTALLATIONS STEPS

### 1. Copy files to server

Upload the files required for installation to the server completely.

If you using apache : use .htaccess

If you using nginx :

set nginx index file : fix.php

set nginx router :

```
if (!-e $request_filename){  
    rewrite ^/([^\?]*)$ /fix.php?url=$1 last; break;  
}
```

### 2. Go to your install domain. domain.com/install

Enter your database information to be installed

**"Database Server", "Database Name", "Database User Name", "Database Password"**

Default Root Account : [demo@demo.com](mailto:demo@demo.com) | 123

### 3. Import Sample Data

You can see the working structure of the system by uploading the sample data.

Login > Settings > System Management and click Import (Go)

## Project Tree

- Project
  - App
    - Main
      - Controller
        - System routing controller layer
      - Hook
        - System permissions settings
      - Model
        - System models and db process
      - Plugin
        - System external plugins
      - Router
        - Url routing main.php
      - View
        - System interface layer files includes
    - Assets
      - Template css,js and images
    - Fix
      - System kernel layer
    - Upload
      - File upload dir
    - 3thparty
      - Filemanager
    - composer.json
    - config.php
      - System config file. Database, install config
    - fix.php
      - System running main file
    - robot.txt

## Create Module

Sign in with the root account.

Go to Site Management > site (details) > Create Module.

Note: If you create multiple data, select "**Multiple Record**" or if you are updating a Single data like "**About Us**", select "**Single Record**"

Now select the elements you want to use in the recording form area.

### Type 1

**TEXT INPUT** Remove

### Type 2

**SINGLE OPTION (SELECT BOX)** Remove

### Type 3

**TEXT INPUT** Remove

☒ X0X0-X0X0X0-X0X0-00 19 Character  
☐ AAAAAAAAAA 10 Character  
☐ 1234567890 10 Character

**Please pay attention here.**

Input Key	Do not use "Space, Special Character" in this field. This field must be unique and is used for api hooks.
Input Title	This field is used to recognize the input in the form. It should be short and meaningful.
Input Placeholder	This field is used to set an example for the input in the form. It is a preview text.
Requirement	It will select whether the data to be entered in this field will be compulsory recording.
Table	It is the option to display in the data record list in multiple data records.
Data (Options)	This field is the field where the data to be selected for multiple or single selections are entered. In this field, enter the data you want to be selected by separating them with a comma.
Random	Thanks to this field, it is used for automatic registration definition code assignment in your multiple records.

**Multiple record slug action (Auto Sef-link)**

Title

Blog

Menu Selection

Passive

Url Slug

Active

Operation Type

Multiple Record

Slug Module Key

title

Menu Row Number

0

TEXT

TAGS

OPTION (SELECT BOX)

OPTION (RADIO)

NUMBER

PHOTO

TEXT EDITOR

COLOR

OPTION MULTIPLE (CHECK BOX)

TEXT (TEXTAREA)

INPUT MASK (MASK)

RANDOM

COMPONENT

DATE

TEXT INPUT

Remove

title

Title

Title

Requirement active

Table active

## Module Settings :

- **Operation Type** : Multiple Record
- **Slug Search** : Active
  - **Slug Key** : Input Key of Slug Key

It is the use of the "TEXT" component recommended in this area.

<b>Input Key</b>	It should be the same as the " <b>Input Key of Slug Key</b> " assigned to the module.
<b>Requirement</b>	active
<b>Table</b>	active

After the desired components are installed, click the "**SAVE**" button and the module will be ready to use.

There will be "**Multiple Module**" on the left menu.

## **User Permission**

User management permission controlling action page.

Name	Account owner first name						
Surname	Account owner surname						
Phone	Account access phone						
E-Mail	Account access login email						
Password	Account access login password						
User Types	Account owner permission type <table><tr><td>Admin</td><td>All in one permission.</td></tr><tr><td>User</td><td>User is limitation permissions.</td></tr><tr><td>Integration</td><td>Developer permission limit.</td></tr></table>	Admin	All in one permission.	User	User is limitation permissions.	Integration	Developer permission limit.
Admin	All in one permission.						
User	User is limitation permissions.						
Integration	Developer permission limit.						
User Permissions	Account owner module and access permission.  This is dynamic permissions options has been accepting						

Authorized Information

Name

Test

Surname

User

Phone

00000000000000

E-Mail

test@test.com

Password

User Types

Admin

User Permissions

Demo

Logo Settings

Contact Settings

Blog Management

Product Management

Reading

Remove

Cancel

Update

## API Documentation

Log in to the account you want to integrate with "Admin" or "Integration" privileges.

Dashboard > Integration (Api Management)

Step 1 : Create api key with customized authority.

Step 2 : Select to “**Module**” > “**Details**” > “All Parameters and Integration Training is inside”

Step 3 : Complete the process of integrating into your software.

## #PHP CLASS TUTORIAL

\$API\_ENDPOINT = SASS ENDPOINT LINK

\$SECRET = PRIVATE SITE CODE

\$TOKEN = CUSTOM UNIQUE CODE

Hooks of class :

addRecord	(\$module = null, array \$data = [])	Module : Hook Code Data : [ “Input Key” => “data” ]
updateRecord	(\$module = null, \$uuid = null , array \$data = [])	Module : Hook Code Uuid : Record Id Data : [ “Input Key” => “data” ]
deleteRecord	(\$module = null, \$uuid = null)	Module : Hook Code Uuid : Record Id
getRecords	(\$module = null, \$orderBy = "DESC", \$orderLimit = 9999)	Module : Hook Code OrderBy : Sorting (ASC DESC) OrderLimit : Show data count
getCategories		
getRecordWithId	(\$module = null, \$uuid = null)	Module : Hook Code Uuid : Record Id
getRecordWithSlug	(\$module = null, \$slug = null)	Module : Hook Code Slug : Slug name
getRecordPagination	(\$module = null, \$page = 1, \$orderBy = "DESC", \$perPageLimit = 10)	Module : Hook Code Page : Show data of page OrderBy : Sorting (ASC DESC) PerPageLimit : Show data count
getPage	(\$module = null)	Module : Hook Code
updatePage	(\$module = null, array \$data = [])	Module : Hook Code Data : [ “Input Key” => “data” ]