CHARGING STATION MANAGEMENT SYSTEM ( CSMS )

HANDOVER DOCUMENT

DEVELOPER :

MUHAMMD HUSARY @ SASCHA HUSARY BIN BENDLIN

Contents

[TEMPLATE 4](#_Toc97582249)

[Original template url 4](#_Toc97582250)

[Genesis template url 4](#_Toc97582251)

[UI Toolkit ‘element-ui’ url 4](#_Toc97582252)

[WORKING CODE REPOSITORY 4](#_Toc97582253)

[JomCharge STAGING URL 4](#_Toc97582254)

[API 4](#_Toc97582255)

[JomCharge API – Apollo Team 4](#_Toc97582256)

[Diagrams 4](#_Toc97582257)

[Google Maps 5](#_Toc97582258)

[To Use Template 5](#_Toc97582259)

[Guide 5](#_Toc97582260)

[To run with ‘https’ 5](#_Toc97582261)

[To point to internal Mock API server 5](#_Toc97582262)

[To point to Jomcharge Api 6](#_Toc97582263)

[To use JomCharge API 7](#_Toc97582264)

[To Sign up 7](#_Toc97582265)

[To Sign In using OTP ( for normal user ) 7](#_Toc97582266)

[To verify OTP 7](#_Toc97582267)

[To sign In ( as Administrator ) 7](#_Toc97582268)

[To Get List of Organizations 8](#_Toc97582269)

[The Template 8](#_Toc97582270)

[Add new page 8](#_Toc97582271)

[Add route for new page 9](#_Toc97582272)

[Add new API in internal API Mock Server 9](#_Toc97582273)

# TEMPLATE

## Original template url

https://github.com/Armour/vue-typescript-admin-template

using VUE3 , Typescrit and UI Toolkit ‘element-ui’

## Genesis template url

https://github.com/PanJiaChen/vue-element-admin

## UI Toolkit ‘element-ui’ url

https://element-plus.org/en-US/

# WORKING CODE REPOSITORY

https://github.com/saschabendlin/csms

# JomCharge STAGING URL

https://staging.jomcharge.com/admins/sign\_in

## API

## JomCharge API – Apollo Team

https://jomcharge.itstartechs.com/jomcharge-main/docs#/

## Diagrams

Using web tools

url : <https://diagrams.net>

shared file :

<https://1drv.ms/u/s!AoTMammC29SvgP9T5EPo8IydZ14oKQ?e=xNEjoi>

or

with shared using email:

1. open diagrams.net
2. ‘File’ 🡪 ‘Open From’ 🡪 ‘OneDrive’ 🡪 ‘Shared’

# Google Maps

https://stackblitz.com/edit/vue-bujcvu?file=src%2Fcomponents%2FMarkers.vue

https://medium.com/@glavecoding/google-map-api-in-vue-js-925c7052a9b6

\*\* need a valid Google MAPS API key \*\*

# To Use Template

## Guide

1. Retrieve code from github repo
2. Open Visual Code
3. Open ‘Folder’ and point to the source code folder
4. Open ‘Terminal’
5. In ‘Terminal’, type
   1. npm install
   2. npm run serve

## To run with ‘https’

1. Open file ‘vue.config.vue’
2. Comment-out the line 22

// https: true,

## To point to internal Mock API server

By default, this template is provided with a internal mock server api, in order to simulate mock data request and response.

1. Open file ‘vue.config.vue’
2. Comment-out the line 34
3. Comment the line 33
4. Open file ‘users.ts’ in folder ‘src/api’
5. Comment-out the line 48-52
6. Comment the line 37-46

## To point to Jomcharge Api

The API provided by the Apollo Team

1. Open file ‘vue.config.vue’
2. Comment-out the line 33
3. Comment the line 34
4. Open file ‘users.ts’ in folder ‘src/api’
5. Comment-out the line 37-46
6. Comment the line 48-52
7. Test by goint to login page, refresh the page, and login
8. By using the ‘Inspect – Dev Tools’ , check the response received from Jomcharge API
   1. Will receive data with user token

# To use JomCharge API

url : https://jomcharge.itstartechs.com/jomcharge-main/docs#/

using Swagger

## To Sign up

1. Manually pass email and hp number to the Apollo Team to sign up

## To Sign In using OTP ( for normal user )

1. Use the ‘auth/signin/user/otp’ API with http verb ‘POST’
2. Click button ‘Try out’
3. Enter the registered hp number in ’Request Body’ space provided
4. {
5. "phone\_number": "+60123456789"
6. }
7. Click ‘Execute’
8. Will receive the response message "OTP has been sent to your phone number."
9. Will receive the OTP number in the registered hp

## To verify OTP

1. Use the ‘auth/signin/user/otp/verify’ API with http verb ‘POST’
2. Click button ‘Try out’
3. Organization Id will be provided by the Appolo Team
4. Enter the received *OTP, hp number and organizaton id* in ’Request Body’ space provided
5. Sample request data

{

"token": "635540",

"phone\_number": "+60178828844",

"organization\_id": "61a0ab6b-a586-44f4-83c7-06811da240fb"

}

1. Click ‘Execute’
2. Will receive response data ‘auth-token’ and ‘token\_type’

## To sign In ( as Administrator )

1. Use the ‘auth/signin/staff’ API with http verb ‘POST’
2. Click button ‘Try out’
3. Organization Id will be provided by the Appolo Team
4. Enter the received *email, pasword and organizaton id* in ’Request Body’ space provided
5. Sample request data

{

"email": " admin@jomcharge.com",

"password": " password",

"organization\_id": "61a0ab6b-a586-44f4-83c7-06811da240fb"

}

1. Click ‘Execute’
2. Will receive response data ‘auth-token’ and ‘token\_type’
3. Copy the auth-token data
4. At the top of the Swagger page, click ‘Authorize’ button
5. Paste in the copied auth-tkoen data into ‘value’ field
6. Click ‘Authorize’ and click ‘Close’

## To Get List of Organizations

1. Use the ‘/organization’ API with http verb ‘POST’
2. Click button ‘Try out’
3. Click ‘Execute’
4. Will receive a response with a list of organizations in the form of array

# The Template

## Add new page

In ‘src/views/ folder,

1. Create a file named ‘xxxx.vue’
2. Create vue template with some html codes
3. Then add route setting for the page

## Add route for new page

This new page will act as a canvas.

‘add’, ‘edit’ pages should be created as a component, and will be injected into the canvas at runtime.

In ‘src/route’ folder

1. Add the new page into ‘RouteConfig[]’ section accordingly
2. And save
3. In browser, the new page will appear in menu
4. Click the new page in the menu
5. Will display the new page

## Add new API in internal API Mock Server

In folder ‘/mock’

1. Create a file with the entity name, for example ‘xxxx.ts’
2. Key in the new APIs, using the ‘articles.ts’ file as reference
3. Import the new file into the file ‘api.ts’
4. Create new route in the ‘swagger.yaml‘ file , using the existing settings as reference
5. Save all modified files
6. Call the new API from the relevant ‘src/api’ folder file to test it, or even can use Postman to test