

Documentation

Developer : Bastisapp

Author name : bastikikang

Email : bastikikang@gmail.com

Author Link : <https://codecanyon.net/user/bastikikang/portfolio>

Software Framework : Yii, Vuejs 3, Pinia vuejs, Capacitor , Mysql and PHP

Date of release : March 03, 2022



**KARENDERIA
DRIVER APP
DELIVERY APP**

FEATURES!

- Android - Google play
- PWA - progressive web apps
- Dark and light mode
- Real time notification
- Wallet / Earnings / Cashout
- Auto assign features
- Complete delivery management

Logos: Progressive PWA Web App, Quasar, capacitor, android

Smartphone Screen Display:

New Order #10213

Map showing route from Los Angeles to Long Beach.

Delivery Details:

- 1. Pickup: McDonald's, 2910 South Figueroa Street, Los Angeles, CA, USA
- 2. Drop-off: Baskin-Robbins, 1 World Way, Los Angeles, CA 90048, USA

Order total: \$649.80

Buttons: Accept, Decline

Table of contents

- A. Requirements
- B. How to activate license
- C. Where can I find the purchase code
- D. How to install addons
- E. How to setup Firestore database
- F. Running cron jobs
- G. How to build PWA
- H. How to build Android
- I. Configuring your android app
- J. Configuring PWA
- K. How to download the latest version
- L. How to upgrade to the latest version
- M. Quick guide full setup of your single website
- N. PWA app Customization
- O. Android Customization
- P. Troubleshooting
- Q. Video tutorials
- R. Change logs
- S. Source and Credits
- T. Finally

Requirements

Server requirements

- App can be installed in subdomain or main domain
- If you want to install on localhost use XAMPP
- SSL is required for PWA
- Server minimum requirements
 - PHP version 7.2 + (karenderia is build and stable in 7.4.27)
 - Mysql 5.6+
 - Mod_rewrite Apache
 - JSON php extension
 - MBstring php extension
 - PDO PHP extension
 - Ziparchive extension
 - Fileinfo Extension
 - GD Library (>=2.0) OR Imagick PHP extension (>=6.5.7)

In most server, these extensions are enabled by default, but you should check with your hosting provider.

How to activate your license

Steps are:

- Open this link in your browser <http://bastisapp.com/activation/>
- In the respective fields, put your Name, E-mail, CodeCanyon Username, Purchase Key and your domain
- Click activate
- You can change the activation later from this same form. Activating a Regular license again with another domain name will remove the activation of the previous domain
- If you're doing a test in your **localhost** enter a domain **localhost.com**
- If you're ready to go live you need to update your domain registration
<http://bastisapp.com/activation/index.php/index/updatedomain>

Where can I find the purchase code

- Login into your envato market account
- Hover the mouse over your username at the top of the screen.
- Click 'Downloads' from the drop-down menu.
- Click 'License certificate & purchase code' (available as PDF or text file).

If you still cannot find it envato has a video and image preview where to get it

Go to this link <https://help.market.envato.com/hc/en-us/articles/202822600-Where-Is-My-Purchase-Code->

How to install addons

To install the addons follow these simple steps below.

- First Extract the zip file **Karenderia Driver App.zip**
- To install addons login to your admin panel and navigate to **addon manager**
- Click on **install/update addon**
- Locate the **DriverModules.zip** that you extracted
- Click **install/update**
- **Finish**

Setup Firestore database

To make sure that you can track driver location in real time you need to create your firestore database.

- First step is go to your firebase console <https://console.firebase.google.com/>
- Create your first project if you don't have one yet or select your existing firebase project.
Make sure you only use 1 firebase project for your all apps
- In your firebase console locate the menu Firestore database, you can this either in your short cut menu or in build menu
- Create your firestore database if you see message saying
This project is set up to use Cloud Firestore in Datastore mode. This mode can only be accessed from the Google Cloud Console

Just click on Go to google cloud console, after that you will be redirected to new page and Click to switch to native mode button and then close window and go back to firebase console And you will see the your database is now created.

- Now click on Roles tab this menu is next to data and under edit rules
Put this rules

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write;
    }
  }
}
```

- Now we get our firestore database credentials by going to project overview, on main dashboard click web icon and register app
- Once you get the firebase configuration you need to set this credentials in admin panel -> delivery management -> firebase settings, make sure you copy the correct data and watch out for spaces
- Also we need to set the firebase store configuration in our Driver app , open the file src/boot/firebase.js

Replace this part with your firebase store credentials

```
{
  apiKey: "AIzaSyDzLT9hE-NQV0-JXzYWPi93LwHdidLnrC4",
  authDomain: "kmrs-demo.firebaseio.com",
  databaseURL: "https://kmrs-demo.firebaseio.com",
```

```
projectId: "kmrs-demo",
storageBucket: "kmrs-demo.appspot.com",
messagingSenderId: "46158489047",
appId: "1:46158489047:web:6c1442a9e7ac9e501e1792",
measurementId: "G-G326YZHY21",
};
```

Runing cron jobs

Cron jobs is required in driver app , make sure you run cron jobs found in your admin panel -> Delivery management -> settings -> cron jobs

This cron jobs is to handle failed run actions for driver app.

How to build PWA

To build your **PWA** you need to install first **Node.js** and **NPM** first and **Visual studio code**.

Quasar framework requires Node 14+ for Quasar CLI with Vite

Guide for **node.js** and **npm** installation <https://phoenixnap.com/kb/install-node-js-npm-on-windows>

Download visual studio here <https://code.visualstudio.com/download>

Important: PWA works only with **SSL** so make sure your site has **SSL** installed example

<https://yourserver.com>

- First we need to install some extension for visual studio code, in your visual studio code
Under File -> preferences -> extension -> search and install the following
 - **Vetur** by Pine Wu
 - **ESLint**
 - **Vue Volar extension Pack**
- Extract the **DriverVue.zip**
- In your visual studio code add the folder **DriverVue** to workspace

- Open new terminal in your visual studio code on the project **DriverVue**
- In your terminal run this command **npm install -g @quasar/cli**
- Run this command **npm install**
- Login to admin panel and locate **Delivery management** on the sub menu **Settings => API**

Access click on **Generate api keys**.

Copy the **Mobile API URL, Payment API URL and Mobile API keys**, now open up your vue files

And open the file **src/api/config.js**

And simply set the value for the following

api_base_url = API URL

api_payment_url = Payment API URL

api_token = Mobile API Keys

- It's very important that don't change the value of **topic** and **channel**
- Finally we build the PWA app by running this command in your terminal **quasar build -m pwa** or if you want to run in debug mode for development run **quasar dev -m pwa**
- Once the build is success you will find a folder in **DriverVue/dist/pwa**
copy the files inside **pwa** to your website host and you can simply now access your pwa app
example <https://yourserver.com> . please use sub domain and don't copy the **pwa** files to sub folder. Do not copy **pwa** folder, copy all the files inside **DriverVue/dist/pwa** folder
example of sub domain <https://pwa.yourserver.com>
- **Finish**

Note : if command quasar command does not work you need to open up your command prompt and run it as administrator and execute this command

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

If you still cannot make the command work read more solution on this link

<https://stackoverflow.com/questions/47023796/visual-studio-code-unsigned-powershell-scripts>

How to build Android

To build android app you need to install android studio first always install the latest android studio. Follow this guide <https://quasar.dev/quasar-cli-vite/developing-capacitor-apps/preparation>

Note: follow only the guide step 1 installation

Before doing this make sure your already set your **api_base_url** , **api_payment_url** and **api_token** In **src/ src/api/config.js** under **build pwa** section.

This guide below only to show how to quick build the app in android, to fully complete build you need to change all the configuration to yours such as providing firebase project for push, google login , social login etc. follow the guide under **configuring your android application**.

- In your Visual Studio code open a terminal with your project **DriverVue** and cd to **src-capacitor** folder by typing **cd src-capacitor** once you inside the **src-capacitor** folder and run the command **npm install**
- To quick run the app in android we first run it with development mode
By running this command **quasar dev -m capacitor -T android**
This will open up your android studio if you already have configure your device emulator just click play button to run the app in your android emulator.
Also you can run it with your actual mobile phone by connecting your device to your computer
And make sure the developer options is enabled in your phone, you can find many video how to turn on developer options in your phone such as this
<https://www.youtube.com/watch?v=gn4bRTFicZw>
- Finally once your app is completed you can build the app by running this command
quasar build -m capacitor -T android -d
once build is done open up your android studio and open the project **src-capacitor/android**, or if you already run the dev command you probably already open the files, all of you have to do is sign the app, go to your android studio and locate **Build -> generate signed bundle / apk** and provide your keystore on next page.
- **Finish**

Configuring your android app

- **Build your app**

https://youtu.be/9-_CkcVEVdU

- **Configure firebase project**

<https://youtu.be/lvQGqTJFH4M>

- **Create keystore**

https://youtu.be/_5SoV6dsMDo

- **Change push sounds notifications**

To change the push notifications sounds just replace the file in with same filename in

src-capacitor\android\app\src\main\res\raw\notify.mp3

- **Change android package ID and package name**

https://youtu.be/2owVmp2EX_w

Configuring PWA

- **Build your PWA app**

<https://youtu.be/nkhVzG2pGnc>

How to download the latest version

- Login to your Codecanyon account.
- Hover over your username from the top right corner and click 'Downloads' from the drop-down menu
- The downloads section displays a list of all the items purchased using your account Click the 'Download' button next to the item and select 'Main File(s)' which contains all
- The downloads section displays a list of all the items purchased using your account Click the 'Download' button next to the item and select 'Main File(s)' which contains all files, or 'License Certificate and Purchase Code' for the item license information only.

How to upgrade to the latest version

To update to latest version process is still the same as installation

You need to upload the **modules.zip** under **admin panel -> addon manager**

And to use the pwa vue you need to use the new files and compile the web app again.

Using **quasar build -m pwa** command

Quick guide full setup of your PWA and Driver app

In this section I will guide you to the quick minimal setup information that needed to run your pwa app and mobile app.

Login to your merchant panel

- **Fill your website details**
 - Go to site merchant -> information and fill merchant nformation tab and address
- **Set your google API keys**
 - Merchant -> settings -> Map api keys
Choose google maps as provider and enter your Geocoding API Key and Google Maps JavaScript API.
You can go here <https://developers.google.com/maps/documentation/places/web-service/get-api-key> on how to get your google api key

You can create or start your google projects here

<https://console.developers.google.com>

The following API should be enabled in your google projects

1. Distance Matrix API
2. Geocoding API
3. Maps Javascript API
4. Places API
5. Maps Static API

Things to note when creating google api:

1. you cannot restrict the api key for Geocoding API Key as this key does not show anywhere in your website.
2. For Google Maps JavaScript API you can restrict this to your domain or server I.p address
3. Make sure you enabled your google billing account.
4. In the new version of karenderia the call to google api is lessen compared to old version. Only the api will be called if necessary.

- **Merchant -> Search mode**
In this section you can set your default country
- **Merchant -> Settings -> Login & Signup**
In this section you set how your customer register to your website
- **Merchant -> -> Settings -> Phone settings**

Set your default phone country code

- **Merchant -> Settings -> Social settings**

In here you can allow your customer to easily login using social login like facebook and google

- **Merchant -> notifications settings**

Set the email address and phone number that will receive notifications if there is new order

- **Payment Gateway**

To receive order from customer you need to add or enabled payment gateway.

Navigate to admin panel -> Payment gateway and from here you can see the list of available Payment gateway for karenderia

- Choose one of the payment gateway example stripe click edit on the pencil icon ,
Once edit you will see the form that ask for Payment API credentials just set your stripe api like Secret key Publishable key and webhook signing secret key
- **Enabled payment gateway to merchant .**
In order to enabled a payment gateway to merchant navigate to Merchant -> list and choose the merchant you need to edit and go to Payment settings -> enabled your desire payment gateway and click save.

Things to note: karenderia has 2 different types of merchants

Commission and membership

- **Commission**
This type of merchant will sale a item in your site and all payment will go to your account like stripe paypal etc, in short all payments will be collected by website owner.
- **Membership**
Merchant will pay a plan subscription like monthly yearly etc
And all payments will go straight to merchant account.

A video will tutorial can be found for this in video tutorial sections on how to configure payment gateway.

- **Finish**

PWA app Customization

- **Changing PWA app title**

- To change the pwa application name locate the file **src-pwa/manifest.json**
Change the **name** and **short name**

- **Change the pwa app default icon**

- To change the pwa app icon you have to options
 1. Manually change the app icon found in public/icons
 2. Auto generate icons using icon genie
By executing the command

icongenie generate -m pwa -i /path/to/source/icon.png [-b /path/to/background.png --splashscreen-color fff4ef]

more information how to use icon genie cli

<https://quasar.dev/quasar-cli-vite/developing-pwa/app-icons-pwa>

- **Change the color of pwa and mobile app**

- To change the color simply change the color in **src/css/ quasar.variables.scss**
And main css **app.scss**

Android app Customization

- Change app icon and splash screen

- To change the android app icon you have to options
 1. Manually change the app icon found in src-capacitor\android\app\src\main\res
 2. Auto generate icons using icon genie
By executing the command

icongenie generate -m capacitor -i /path/to/source/icon.png --padding 10,10 [-b b /path/to/background.png --splashscreen-color fff4ef]

more information how to use icon genie cli

<https://quasar.dev/quasar-cli-vite/developing-capacitor-apps/app-icons-capacitor>

Troubleshooting

Here are the common mistake when single web app does not work

- **some server does not return bearer token**
 - Solution is to edit the .htaccess and add RewriteCond %{HTTP:Authorization} ^(.*)
 - RewriteRule .* - [e=HTTP_AUTHORIZATION:%1] after RewriteEngine Onexample

```
RewriteEngine On
RewriteCond %{HTTP:Authorization} ^(.*)
RewriteRule .* - [e=HTTP_AUTHORIZATION:%1]
```
- **API blocking request**
 - When generating api keys make sure you're in your website domain url
- **Cannot execute quasar command**
 - Make sure you installed NPM and quasar cli in your local computer, quasar requires node+12More information in <https://quasar.dev/start/pick-quasar-flavour>

Video tutorials

- <https://www.youtube.com/channel/UCMSzf36wnBxCwC4sriVmFFQ/videos>

Change logs

- 1.0 .0 (03 March 22)
 - Initial release

Source and Credits

I've used the following images, icons or other files as listed.

- [Yii framework](#)
- [Quasar dev framework](#)
- [Vuejs 3](#)
- [Pinia vuejs](#)
- [Axios](#)
- [google font](#)
- [freepik.com](#)
- [Capacitor](#)

Finally

Once again, thank you so much for purchasing this application. As I said at the beginning, I'd be glad to help you if you have any questions relating to this theme. No guarantees, but I'll do my best to assist. If you have a more general question relating to the themes on codecanyon, you might consider visiting the forums and asking your question in the "Item Discussion" section.