

Silence

Silence is fully responsive single page dashboard template built with Bootstrap 3+, Vue.js and Vue Router. The theme is fully responsive, easy to customize, and the source code is clean. This template comes with a collection of easy to use components and widgets.

Build Setup

```
# install dependencies
npm install

# serve with hot reload at http://silence.test:8080/
npm run dev

# build for production with minification
npm run build

# build for production and view the bundle analyzer report
npm run build --report

# run unit tests
npm run unit
```

For a detailed explanation on how things work, check out the [guide](#) and [docs for vue-loader](#).

Built files are meant to be served over an HTTP server. Opening `.../dist/index.html` over `file://` won't work.

Features

- Single Page Application built with Vue.js and Vue Router
- Modular, component-based router configuration
- View transition effects powered by Vue.js' transition system
- 7 color schemes
- 31 components
- 12 pages
- SCSS source files (compiled CSS included)
- Javascript sources written using ES6 (compiled JS included)
- Built with Bootstrap 3.x
- NPM package manager
- Webpack module bundler

- Hot reload
- Easy to customize
- Clean source code
- Flexible modular code structure, which allows you easily remove unused plugins and components

Project Structure

```

.
├─ build/                # webpack config files
│   └─ ...
├─ config/
│   ├── index.js         # main project config
│   └─ ...
├─ src/
│   ├── main.js          # app entry file
│   ├── App.vue          # main app component
│   ├── pages/           # pages
│   │   └─ ...
│   ├── layout/          # layout ui components
│   │   └─ ...
│   ├── components/      # ui components
│   │   └─ ...
│   ├── assets/          # module assets (processed by webpack)
│   │   ├── scss
│   │   └─ ...
│   ├── router/          # routes
│   │   └─ ...
├─ static/               # pure static assets (directly copied)
├─ test/
│   └─ unit/             # unit tests
│       ├── specs/       # test spec files
│       ├── eslintrc     # config file for eslint only for unit tests
│       ├── jest.conf.js # Config file when using Jest for unit tests
│       └─ setup.js      # file that runs before Jest runs your tests
├─ .babelrc              # babel config
├─ .editorconfig         # settings for your editor
├─ .eslintrc.js          # eslint config
├─ .eslintignore         # eslint ignore rules
├─ .gitignore            # sensible defaults for gitignore
├─ .postcssrc.js         # postcss config
├─ index.html            # index.html template
├─ package.json          # build scripts and dependencies
└─ README.md

```

Documentation

Panels

Panel with heading

```
<panel :header="true"> Body... </panel>  
or  
<panel>Body...</panel>
```

Usage

```
<panel type="panel-green" :header=true>  
  <h3 slot="header">Header</h3>Body...  
</panel>
```

Header is set to true by default.

Panel with footer

```
<panel :footer=true> Body... </panel>
```

Usage:

```
<panel type="panel-green" :header=false :footer=true>  
  Body...  
  <h5 slot="footer">Footer</h5>  
</panel>
```

Panel Types

- panel-primary
- panel-green
- panel-red
- panel-yellow
- panel-orange
- panel-teal
- panel-purple
- panel-stone

- panel-moss
- panel-mist
- panel-sky
- panel-pink
- panel-blue-sky
- panel-strawberry
- panel-blueberry
- panel-gray
- panel-brown
- panel-white

Run tests

```
npm run unit Panel
```

Datatables

[DataTables](#) is a plug-in for the jQuery Javascript library. It is a highly flexible tool, based upon the foundations of progressive enhancement, and will add advanced interaction controls to any HTML table.

HTML (DOM) sourced data

```
<datatable :table="true"><table>...</table></datatable>
```

Javascript sourced data

```
<datatable :table-data="[{'title: 'Title'}, {'title: 'Your'},...]"  
           :columns="[['1', '2'],...],...]">  
</datatable>
```

Options

```
<datatable :options="{...}"></datatable>
```

Run tests

```
npm run unit Datatable
```

Charts.js

[Chart.js](#) is simple yet flexible JavaScript charting for designers & developers.

Chart types

<http://www.chartjs.org/docs/latest/charts/>

Usage

```
<chart type="pie" :height="75"  
      :labels='["Red", "Blue", "Yellow"]'  
      :datasets="{data: [300, 50, 100]}">  
</chart>
```

Options

```
<chart :options="{...}"></chart>
```

Run tests

```
npm run unit Chart
```

Chartist

Chartist - Simple responsive charts.

```
<chartist chart-height="height"  
          type="type"  
          :data="data"  
          :options="options">  
</chartist>
```

Usage

```
<chartist chart-height="300px"  
          type="bar"  
          :data="{labels: [...], series: [[...]]}"  
          :options="{...}">  
</chartist>
```

Run tests

```
npm run unit Chartist
```

Peity

Peity - progressive svg pie, donut, bar and line mini charts.

```
<inline-chart type="[pie|donut|bar|line]" :options="{...}">
  1/2
</inline-chart>
```

Usage

```
<inline-chart type="donut" :options="{radius:14,fill:['#fa7252','#fff4dd']}">
  4.2/5
</inline-chart>
```

Run tests

```
npm run unit InlineChart
```

PhotoSwipe

PhotoSwipe - Responsive JavaScript Image Gallery

```
<photo-swipe :items="[ {... }, ... ]" :options="{...}"></photo-swipe>
```

Usage

```
<photo-swipe :items="[
  {
    src: 'http/...',
    w: '960px',
    h: '960px',
    title: 'Title',
    author: 'Author'
  }
]"
:options="{...}">
</photo-swipe>
```

Run tests

```
npm run unit PhotoSwipe
```

Calendar

FullCalendar - A JavaScript event calendar. Customizable and open source.

```
<calendar :events="events"
:draggable="true"
:external-events="values.externalEvents"
:delete-events="true"
:add-events="true"
@eventAdded="addEvent"
@externalEventAdded="addExternalEvent"
></calendar>
```

Properties & Events

- events - An array of Event Objects that will be displayed on the calendar.
- externalEvents - Array of events that can be dropped onto the calendar.
- draggable - Determines if jQuery UI draggables can be dropped onto the calendar.
- aspectRatio - Determines the width-to-height aspect ratio of the calendar.
- deleteEvents - Determines if events can be deleted.

- options - Options Object.
- addEvents - If true, add event form will be displayed.
- @externalEventAdded - emitted when event is added.
- @eventAdded - emitted when event is dropped onto calendar.
- onDrop & onDropExternal - Called when a valid jQuery UI draggable has been dropped onto the calendar.
- onResize - Called when event is resized.
- onDropExternal - Called when external event has been dropped onto the calendar.

Run tests

```
npm run unit Calendar
```

Select2

Select2 - The jQuery replacement for select boxes

```
<v-select2 :data="{...}"></v-select2>
```

With placeholder

```
<v-select2 :data="{...}" placeholder="Custom Placeholder"></v-select2>
```

Disabled Search

```
<v-select2 :data="{...}" :search="false"></v-select2>
```

Usage

```
<v-select2 :data="{laravel: 'Laravel', gulp: 'Gulp', sass: 'Sass'}"
  placeholder="Custom Placeholder"
  v-model="data">
</v-select2>
```


Run tests

```
npm run unit Select2
```

Color Picker

Bootstrap Colorpicker is a simple and customizable colorpicker component.

Component supports v-model.

With placeholder

```
<colorpicker placeholder="Custom Placeholder"></colorpicker>
```

With icon

```
<colorpicker icon="true"></colorpicker>
```

Usage

```
<colorpicker :icon="true" v-model="color " placeholder="Enter Color">
</colorpicker>
```

Run tests

```
npm run unit Colorpicker
```

Date/Time Picker

Component supports v-model.

Enabling Icon

```
<datepicker :icon="true"></datepicker>
```

Custom Format

```
<datepicker format="DD-MM-YYYY"></datepicker>
```

Custom Icon

```
<datepicker :icon="true"  
            custom-icon='<i class="fa fa-clock-o" aria-hidden="true"></i>'  
</datepicker>
```

Usage

```
<datepicker :icon="true"  
            format="LT"  
            custom-icon='<i class="fa fa-clock-o" aria-hidden="true"></i>'  
</datepicker>
```

Run tests

```
npm run unit Datepicker
```

Tabs

Usage

```
Usage  
<tabs>  
  <tab title='Tab 1' :selected="true"><h4>Content of Tab 1</h4>  
  ...  
</tab>  
  <tab title='Tab 2'><h4>Content of Tab 2</h4>  
  ...  
</tab>  
  <tab title='Tab 3'><h4>Content of Tab 3</h4>  
  ...  
</tab>  
</tabs>
```

Select active tab with :selected="true".

Verical Tabs

```
<tabs :vertical="true" :right="[true|false]"></tabs>
```

Run tests

```
npm run unit Tabs  
npm run unit Tab
```

Modal

Modals are streamlined, but flexible, dialog prompts with the minimum required functionality and smart defaults.

```
<modal id="modal-id" :show='showModal' @modalClosed="showModal = false">  
</modal>
```

Properties & Events

- show

If true, the modal dialog will be displayed. Pass in whatever state in your application controls this.

- @modalClosed

This event is emitted when the close button of the modal dialog is clicked or is clicked outside of the modal. Don't forget to change the state which controls the show prop.

Small Modal

```
<modal id="modal-id"  
  :small="true"  
  :show='showModal'  
  @modalClosed="showModal = false">  
</modal>
```

Large Modal

```
<modal id="modal-id"  
  :large="true"  
  :show='showModal'  
  @modalClosed="showModal = false">  
</modal>
```

Usage

```
<button type="button" class="btn" @click="showModal = true">  
  LAUNCH MODAL  
</button>  
<modal id="small_modal"  
  :small=true :show="showModal"  
  @modalClosed="showModal = false"  
  >  
  <h3 slot="title">Modal Title</h3>  
  <p>...</p>  
  <div slot="footer">  
    <button type="button" class="btn" @click="modal2 = false">CLOSE</button>  
    <button type="button" class="btn">SAVE</button>  
  </div>  
</modal>
```

Run tests

```
npm run unit Modal
```

Trumbowyg

Trumbowyg - A lightweight WYSIWYG editor

Component supports v-model.

Usage

```
<trumbowyg v-model="text"></trumbowyg>
```

Overriding server path for image uploads

```
<trumbowyg upload-path="http/..."></trumbowyg>
```

Run tests

```
npm run unit Trumbowyg
```

Codemirror

CodeMirror is a versatile text editor implemented in JavaScript for the browser.

Component supports v-model.

```
<codemirror mode="mode" theme="theme" v-model="code"></codemirror>
```

Themes

- material
- dracula
- eclipse
- cobalt
- monokai
- yeti
- zenburn
- twilight

Usage

```
<codemirror mode="php" theme="material" v-model="code"></codemirror>
```

Widget

```
<widget class-name="class" :shadow="Boolean" icon='icon' title="Title">  
  body...  
</widget>
```

Class Names

- widget-green
- widget-blue
- widget-red
- widget-teal
- widget-yellow
- widget-orange
- widget-purple
- widget-mist
- widget-brown
- widget-stone
- widget-moss
- widget-sky
- widget-pink
- widget-blue-sky
- widget-blueberry
- widget-strawberry

With Shadow

```
<widget :shadow="true"></widget>
```

Usage

```
<widget class-name="widget-green"  
  icon='<i class="fa fa-calendar" aria-hidden="true"></i>'  
  title="Events">  
  2  
</widget>
```

Run tests

```
npm run unit Widget
```

Accordion

Usage

```
<accordion id="accordion-id">
  <accordion-item title="Title 1" :active="true"></accordion-item>
  <accordion-item title="Title 2">....</accordion-item>
  <accordion-item title="Title 3">...</accordion-item>
</accordion>
```

Select active panel with :active="true".

Run tests

```
npm run unit Accordion
npm run unit AccordionItem
```

Carousel

A slideshow component for cycling through elements, like a carousel.

Usage

```
<carousel id="carousel-id">
  <carousel-item img-src="/image1.jpg" caption="Caption 1"></carousel-item>
  <carousel-item img-src="/image2.jpg" caption="Caption 2"></carousel-item>
  <carousel-item img-src="/image3.jpg" caption="Caption 3"></carousel-item>
</carousel>
```

Run tests

```
npm run unit Carousel
npm run unit CarouselItem
```

Jumbotron

A lightweight, flexible component that can optionally extend the entire viewport to showcase key content on your site.

```
<jumbotron title="Title" img="img_path">
  body...
  <div slot="buttons"> ...</p>
</jumbotron>
```

Without image

```
<jumbotron title="Title">
  body...
  <div slot="buttons"> ...</p>
</jumbotron>
```

Custom Class

```
<jumbotron title="Title" class-name="custom-class-name">
  body...
  <div slot="buttons"> ...</p>
</jumbotron>
```

Usage

```
<jumbotron title="Hello World" img="/static/imgs/pexels-photo-235615.jpeg">
  <p>...</p>
  <p slot="buttons">
    <button class="btn btn-outlined-orange rounded">Learn More</button>
  </p>
</jumbotron>
```

Run tests

```
npm run unit Jumbotron
```


Credits

[Bootstrap](#)

[Vue.js](#)

[Vue Router](#)

[jQuery](#)

[animate.css](#)

[Social media vector created by Freepik.](#)

[Free stock photos by Pexels](#)

[Chart.js](#)

[Chartist](#)

[Peity](#)

[FullCalendar](#)

[Select2](#)

[eonasdan-bootstrap-datetimepicker](#)

[bootstrap-colorpicker](#)

[PhotoSwipe](#)

[malihu-custom-scrollbar-plugin](#)

[Trumbowyg](#)

[CodeMirror](#)

[Lodash](#)

[Datatables](#)

[epic-spinners](#)

[js-cookie](#)

[jquery-ui](#)

[vue-highlightjs](#)

Fonts

[Lato](#)

[FontAwesome](#)