hilongjw / **vue-progressbar**

A lightweight progress bar for vue http://hilongjw.github.io/vue-progres...

94 commits **2** branches **0** releases **17** contributors MIT

Branch: **master** New pull request Find File Clone or download

hilongjw Merge pull request #81 from chiaweilee/master ... Latest commit a803468 on Dec 3, 2018

dist	add: build v0.7.5	last year
src	refactor: no component require	last year
.babelrc	add: rollup	2 years ago
.gitignore	test	2 years ago
LICENSE	first commit	3 years ago
README.md	🚀 docs: fix default value of 'autoFinish'	last year
package.json	add: build v0.7.5	last year
rollup.config.js	Switched to es6 export and added module name to rollup config	last year

README.md

vue-progressbar

Table of Contents

Demo

Demo

Requirements

- Vue.js `1.x` or `2.x`

Installation

```
# npm
$ npm install vue-progressbar

#yarn
$ yarn add vue-progressbar
```

Usage

main.js

```
import Vue from 'vue'
import VueProgressBar from 'vue-progressbar'
import App from './App'

const options = {
  color: '#bffaf3',
  failedColor: '#874b4b',
  thickness: '5px',
  transition: {
    speed: '0.2s',
    opacity: '0.6s',
    termination: 300
  },
  autoRevert: true,
  location: 'left',
  inverse: false
}

Vue.use(VueProgressBar, options)

new Vue({
  ...App
}).$mount('#app')
```

Constructor Options

key	description	default	options
`color`	color of the progress bar	`'rgb(143, 255, 199)'`	`RGB` `HEX` `HSL` `HSV` `VEC`
`failedColor`	color of the progress bar upon load fail	`'red'`	`RGB` , `HEX` , `HSL` , `HSV` , `VEC`
`thickness`	thickness of the progress bar	`'2px'`	`px` , `em` , `pt` , `%` , `vh` , `vw`
`transition`	transition speed/opacity/termination of the progress bar	`{speed: '0.2s', opacity: '0.6s', termination: 300}`	`speed` , `opacity` , `termination`

key	description	default	options
`autoRevert`	will temporary color changes automatically revert upon completion or fail	`true`	`true` , `false`
`location`	change the location of the progress bar	`top`	`left` , `right` , `top` , `bottom`
`position`	change the position of the progress bar	`fixed`	`relative` , `absolute` , `fixed`
`inverse`	inverse the direction of the progress bar	`false`	`true` , `false`
`autoFinish`	allow the progress bar to finish automatically when it is close to 100%	`true`	`true` , `false`

Implementation

App.vue

```
<template>
    <div id="app">
        <!-- for example router view -->
        <router-view></router-view>
        <!-- set progressbar -->
        <vue-progress-bar></vue-progress-bar>
    </div>
</template>

<script>
export default {
  mounted () {
    //  [App.vue specific] When App.vue is finish loading finish the progress bar
    this.$Progress.finish()
  },
  created () {
    //  [App.vue specific] When App.vue is first loaded start the progress bar
    this.$Progress.start()
    //  hook the progress bar to start before we move router-view
    this.$router.beforeEach((to, from, next) => {
      //  does the page we want to go to have a meta.progress object
      if (to.meta.progress !== undefined) {
        let meta = to.meta.progress
        // parse meta tags
        this.$Progress.parseMeta(meta)
      }
      //  start the progress bar
      this.$Progress.start()
      //  continue to next page
      next()
    })
    //  hook the progress bar to finish after we've finished moving router-view
    this.$router.afterEach((to, from) => {
      //  finish the progress bar
      this.$Progress.finish()
    })
  }
}
</script>
```

vue-router

```
export default [
  {
    path: '/achievement',
    name: 'achievement',
    component: './components/Achievement.vue',
    meta: {
      progress: {
        func: [
          {call: 'color', modifier: 'temp', argument: '#ffb000'},
          {call: 'fail', modifier: 'temp', argument: '#6e0000'},
          {call: 'location', modifier: 'temp', argument: 'top'},
          {call: 'transition', modifier: 'temp', argument: {speed: '1.5s', opacity: '0.6s', termination: 400}}
        ]
      }
    }
  }
]
```

vue-router meta options

call	modifier	argument	example
color	`set`, `temp`	`string`	`{call: 'color', modifier: 'temp', argument: '#ffb000'}`
fail	`set`, `temp`	`string`	`{call: 'fail', modifier: 'temp', argument: '#ffb000'}`
location	`set`, `temp`	`string`	`{call: 'location', modifier: 'temp', argument: 'top'}`
transition	`set`, `temp`	`object`	`{call: 'transition', modifier: 'temp', argument: {speed: '0.6s', opacity: '0.6s', termination: 400}}`

Methods

function	description	parameters	example
start	start the progress bar loading	`N/A`	`this.$Progress.start()`
finish	finish the progress bar loading	`N/A`	`this.$Progress.finish()`
fail	cause the progress bar to end and fail	`N/A`	`this.$Progress.fail()`
increase	increase the progress bar by a certain %	`number: integer`	`this.$Progress.increase(number)`
decrease	decrease the progress bar by a certain %	`number: integer`	`this.$Progress.decrease(number)`
set	set the progress bar %	`number: integer`	`this.$Progress.set(number)`
setFailColor	cause the fail color to permanently change	`color: string`	`this.$Progress.setFailColor(color)`
setColor	cause the progress color to permanently change	`color: string`	`this.$Progress.setColor(color)`
setLocation	cause the progress bar location to permanently change	`location: string`	`this.$Progress.setLocation(location)`

function	description	parameters	example
setTransition	cause the progress bar transition speed/opacity/termination to permanently change	`transition: object`	`this.$Progress.setTransition(transition)`
tempFailColor	cause the fail color to change (temporarily)	`color: string`	`this.$Progress.tempFailColor(color)`
tempColor	cause the progress color to change (temporarily)	`color: string`	`this.$Progress.tempColor(color)`
tempLocation	cause the progress bar location to change (temporarily)	`location: string`	`this.$Progress.tempLocation(location)`
tempTransition	cause the progress bar location to change (temporarily)	`transition: object`	`this.$Progress.tempTransition(transition)`
revertColor	cause the temporarily set progress color to revert back to it's previous color	`N/A`	`this.$Progress.revertColor()`
revertFailColor	cause the temporarily set fail color to revert back to it's previous color	`N/A`	`this.$Progress.revertFailColor()`
revertTransition	cause the temporarily set transition to revert back to it's previous state	`N/A`	`this.$Progress.revertTransition()`
revert	cause the temporarily set progress and/or fail color to their previous colors	`N/A`	`this.$Progress.revert()`
parseMeta	parses progress meta data	`meta: object`	`this.$Progress.parseMeta(meta)`

Examples

Loading Data (vue-resource)

```
<script>
export default {
  methods: {
    test () {
      this.$Progress.start()
      this.$http.jsonp('http://api.rottentomatoes.com/api/public/v1.0/lists/movies/in_theaters.json?apikey=7waqfqbpr
      .then((response) => {
          this.$Progress.finish()
      }, (response) => {
          this.$Progress.fail()
      })
    }
  }
}
</script>
```

Accessing the progress bar externally through the vue instance (e.g. axios interceptors)

main.js

```
// main.js from Usage section

Vue.use(VueProgressBar, options)

export default new Vue({ // export the Vue instance
  ...App
}).$mount('#app')
```

api-axios.js

```
import axios from 'axios';
import app from '../main'; // import the instance

const instance = axios.create({
    baseURL: '/api'
});

instance.interceptors.request.use(config => {
    app.$Progress.start(); // for every request start the progress
    return config;
});

instance.interceptors.response.use(response => {
    app.$Progress.finish(); // finish when a response is received
    return response;
});

export default instance; // export axios instance to be imported in your app
```

License