建一个hello调用例子

FE

1.

Src\vuex 建一个mutation-types.js

export const GET\_MYHELLO = 'GET\_MYHELLO'

export const GET\_MYHELLO\_FAILURE = 'GET\_MYHELLO\_FAILURE'

export const CLEAR\_MYHELLO = 'CLEAR\_MYHELLO'

2.

src\vuex\modules 建一个myHello.js

import {GET\_MYHELLO, GET\_MYHELLO\_FAILURE, CLEAR\_MYHELLO} from '../mutation-types'

const state = {

hello: 'ffff'

}

const mutations = {

[GET\_MYHELLO] (state, data) {

state.hello = data.hello

},

[GET\_MYHELLO\_FAILURE] (state) {

return state

},

[CLEAR\_MYHELLO] (state) {

state.hello = ''

}

}

export default {

state,

mutations

}

3. src\components 建一个Home.vue

<template>

<div class="list-wrapper">

<p>

myhello={{myHello.hello}}

</p>

<!-- <ul class="list-container" id="list-ul">

<li v-for="item in items">

<a v-link="{ name: 'showproduct', params: {id: item.objectId}}">

<h3><p class="list-title">{{item.title}}</p></h3>

<p class="list-updated">{{item.description}}</p>

</li>

</ul>-->

</div>

</template>

<script type="text/babel">

//import {myHello,contentList} from '../vuex/getters'

//import {getContentList, getHello} from '../vuex/actions'

import {myHello } from '../vuex/getters'

import { getHello} from '../vuex/actions'

export default {

vuex: {

getters: {

//items: contentList,

myHello: myHello

},

actions: {

//getList: getContentList,

getMyHello: getHello

}

},

created () {

// console.log("myHello")

this.getMyHello()

//this.getList()

}

}

</script>

<style>

</style>

4. src\vuex建一个store.js(作用暂时不明)

import Vue from 'vue'

import Vuex from 'vuex'

//import contentList from './modules/contentList'

import myHello from './modules/myHello'

Vue.use(Vuex)

Vue.config.debug = true

const debug = process.env.NODE\_ENV !== 'production'

export default new Vuex.Store({

strict: debug,

modules: {

//contentList,

myHello

}

})

5. src\vuex建一个getters.js

export const myHello = ({ myHello }) => myHello

6. src\vuex建一个actions.js

import \* as types from './mutation-types'

import { API\_ROOT } from '../config'

export const getHello = function ({ dispatch }) {

this.$http.get(API\_ROOT + 'api/content-hello').then(response => {

console.log(response)

dispatch(types.GET\_MYHELLO, JSON.parse(response.body))

}, error => {

dispatch(types.GET\_MYHELLO\_FAILURE, error)

})

}

7. src建一个config.js(不知道为什么总会自动变成'production',暂时都写成本机器)

export const API\_ROOT = (process.env.NODE\_ENV === 'production')

? 'http://localhost:3000/'//'http://xxxx.leanapp.cn/'

: 'http://localhost:3000/'

8. src建一个App.vue(可以直接用home标签,或者router-view配跳转)

<template>

<div id="myapp">

<!--<home></home>-->

<div id="mydiv">

<router-view class="main-wrapper" transition="fade" transition-mode="out-in"></router-view>

</div>

</div>

</template>

<script>

import Home from './components/Home'

import Store from './vuex/store'

export default {

components: {

Home

},

store: Store

}

</script>

<style>

</style>

9. src建一个main.js (针对上面的router-view配跳转,id为app的div在index.html里这样用,如果在app.vue里还有些写法区别,具体百度)

import Vue from 'vue'

import App from './App.vue'

import VueRouter from 'vue-router'

import VueResource from 'vue-resource'

//import Home from "./components/home.vue";

/\* eslint-disable no-new \*/

/\*

new Vue({

el: 'body',

components: { App }

})\*/

/\*

new Vue({

el: '#app',

template: '<App/>',

components: { App }

})\*/

// 注册两个插件

Vue.use(VueResource)

Vue.use(VueRouter)

Vue.http.options.emulateJSON = true

/\*

const routes = [

{ path:'/home', component: App }

]

\*/

const router = new VueRouter({

history: true,

hashbang: false

//routes

})

/\*

new Vue({

el: 'body',

router ,

components: { App }

})\*/

// 路由map

router.map({

'/home': {

component: function (resolve) {

require(['./components/Home'], resolve)

}

}

})

router.redirect({

'/index': '/home'

})

router.start(App, '#app')

10. fe下的index.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>fe</title>

</head>

<body>

<div id="app">

<app></app>

</div>

<!-- built files will be auto injected -->

</body>

</html>

11. fe下的package.json

{

"name": "fe",

"version": "1.0.0",

"description": "A Vue.js project",

"author": "zhangsiyuan <1021808625@qq.com>",

"private": true,

"scripts": {

"dev": "node build/dev-server.js",

"build": "node build/build.js",

"test": "",

"lint": "eslint --ext .js,.vue src test/unit/specs test/e2e/specs"

},

"dependencies": {

"vue": "^1.0.21",

"babel-runtime": "^6.0.0",

"vue-router": "^0.7.13",

"vue-strap": "^1.0.11",

"vuex": "^0.8.2"

},

"devDependencies": {

"babel-core": "^6.0.0",

"babel-loader": "^6.0.0",

"babel-plugin-transform-runtime": "^6.0.0",

"babel-preset-es2015": "^6.0.0",

"babel-preset-stage-2": "^6.0.0",

"connect-history-api-fallback": "^1.1.0",

"css-loader": "^0.23.0",

"eslint": "^2.10.2",

"eslint-friendly-formatter": "^2.0.5",

"eslint-loader": "^1.3.0",

"eslint-plugin-html": "^1.3.0",

"eslint-config-standard": "^5.1.0",

"eslint-plugin-promise": "^1.0.8",

"eslint-plugin-standard": "^1.3.2",

"eventsource-polyfill": "^0.9.6",

"express": "^4.13.3",

"extract-text-webpack-plugin": "^1.0.1",

"file-loader": "^0.8.4",

"function-bind": "^1.0.2",

"html-webpack-plugin": "^2.8.1",

"http-proxy-middleware": "^0.12.0",

"json-loader": "^0.5.4",

"ora": "^0.2.0",

"shelljs": "^0.6.0",

"url-loader": "^0.5.7",

"vue-hot-reload-api": "^1.2.0",

"vue-html-loader": "^1.0.0",

"vue-loader": "^8.3.0",

"vue-resource": "^0.9.3",

"vue-style-loader": "^1.0.0",

"webpack": "^1.12.2",

"webpack-dev-middleware": "^1.4.0",

"webpack-hot-middleware": "^2.6.0",

"webpack-merge": "^0.8.3"

}

}

Sever端 配一个hello的http接口

1. server-modules 建hello.js

'use strict';

const tool = require('./tool');

let pub = {};

pub.hello = (req, res) => {

tool.l('It works.');

res.send({

hello: 'It works.'

});

};

module.exports = pub;

1. server-modules 建api-router.js

'use strict';

const router = require('express').Router();

// 添加一个模块

const hello = require('./hello');

// 一个 API 路由下的 hello 接口，访问 /api/hello

router.get('/content-hello', hello.hello)

// 测试 async/await 支持

const f = () => {

return new Promise((resolve, reject) => {

setTimeout(() => {

resolve(123);

}, 2000);

});

};

const testAsync = async () => {

const t = await f();

console.log(t);

};

testAsync();

module.exports = router;

1. server-modules 的app.js 添加

const apiRouter = require('./api-router');

app.use('/api', apiRouter);