## FX-Mobile开发规范说明

### 框架源码路径：

git clone <https://github.com/fx-mobile/fx-mobile.git>

### 创建新页面：

在src/page下添加新页面\*.vue

语法参考：vue风格指南：<https://cn.vuejs.org/v2/style-guide/>

### **如何配置路由：**

1. 在src/router/modules下添加新页面的路由

/\*\*

\* 模块一路由信息

\*/

const company = r => require.ensure([], () => r(require('@/pages/company/')))

const module1 = r => require.ensure([], () => r(require('@/pages/company/module1.vue')))

export default [{

path: 'company', component: company, children: [

{ path: 'module1', component: module1, meta: { title: '栏目一',

keepAlive: true, requiresAuth: true, }, desc: '栏目一' },

]

}]

1. 在src/router/index.js文件配置路由表，格式如下

import module1 from './modules/module-1'

{

path: '/entry',

component: entry,

children: [

...base,

...module1,

]

}

### **如何创建子组件:**

在src/components/下创建业务子组件，组法格式参考vue@2

### **页面如何跳转：**

this.$router.push({

path: "/login",

query: { redirect\_url: this.$route.fullPath }

});

### **发起http请求：**

1. 在src/api/modules/下新增\*.js服务端接口请求配置文件，在src/api/modules/index.js中引入该模块。格式如下：

export default {

test: {

uri: config.cdnUrl +'/v1/user/ping,

method: 'get',

cache: true

},

getUserInfo: { uri: '/v1/user/getUserInfo' },

getNsrInfo: { uri : '/v1/user/getNsrInfo'}

}

2、发起接口请求，传参

import { customAsync } from "@/api/async";

let that = this;

customAsync({

that: this,

method: "getUserInfo",

paramObj: this.form,

callback: function(res) {

getUserInfo(that, null, false).then(res => {

that.$router.push("/login/success");

});

}

});

### **vuex的使用方式：**

1、在src/store/modules下新增\*.js状态管理文件，格式如下：

2、在src/store/index.js中将新增的状态管理文件引入。

export default {

state: {

badgeNum: sessionStorage.getItem('message\_badge\_num') || 0

},

mutations: {

SET\_BADGE\_NUM: (state, num) => {

sessionStorage.setItem('message\_badge\_num', num);

state.badgeNum = num

}

},

actions: {

SetBadgeNum({ commit }, num) {

commit('SET\_BADGE\_NUM', num)

}

}

}

使用方式：

this.$store.dispatch("SetBadgeNum", 0);

### **开发配置文件说明：**

在config/index.js中配置dev结点中配置开发模式下服务连接地址。

dev: {

env: require('./dev.env'),

port: 8062,

host: utils.getLocalIP(),

autoOpenBrowser: true,

assetsSubDirectory: 'static',

assetsPublicPath: '',

proxyTable: {

'/v1': {

target: 'http://172.16.20.122:8089',

changeOrigin: true

},

},

poll: false,

cssSourceMap: false

}

### **命令行说明：**

```

# 开发模式

npm run dev

# 打包 开发环境

npm run build:dev

# 打包 测试环境

npm run build:test

# 打包 预生产环境

npm run build:pre

# 打包 生产环境

npm run build:prod

```

### **Mock服务：**

Mock服务提供，我们使用的是[https://www.easy-mock.com](https://www.easy-mock.com/)（各项目可以自己注册，或者搭建自己的Mock服务）

数据准备：



配置：

在config/index.js中的dev节点配置如下：

proxyTable: {

'/v1': {

target: 'https://www.easy-mock.com/mock/58ff650c739ac1685205a1a7/foresee',

changeOrigin: true

},

},