/\*\*axios封装

\* 请求拦截、相应拦截、错误统一处理

\*/

import axios from 'axios';

import QS from 'qs';

import store from '../store/index'

import { Toast } from 'vant';

import router from '../router/router.js'

let domain = 'dev';

if(domain == 'dev') { // 环境的切换

axios.defaults.baseURL = 'http://192.168.1.35:8086/';

window.alyun = 'http://jnb-air-dev.oss-cn-shenzhen.aliyuncs.com/'; // 阿里云服务器，头像

window.preDomain = 'http://192.168.1.35:8032/'; //换购跳转

} else if(domain == 'test') {

axios.defaults.baseURL = 'http://192.168.1.46:8086/';

window.alyun = 'http://jnb-air-test.oss-cn-shenzhen.aliyuncs.com/'; // 阿里云服务器

} else if(domain == 'pre') {

axios.defaults.baseURL = 'http://test.jnbsoft.cn/';

window.preDomain = 'http://app.wealth.jnbsoft.cn:58086/';

window.alyun = 'http://jnb-air-pro.oss-cn-shenzhen.aliyuncs.com/'; // 阿里云服务器

}

axios.defaults.timeout = 10000; // 请求超时时间

axios.interceptors.request.use( // 请求拦截器

config => {

let token = localStorage.getItem('access\_token');

if(config.url.split('/')[config.url.split('/').length - 1] == "pic") token = '';

token && (config.headers.Authorization = 'bearer ' + token);

return config;

},

error => {

return Promise.error(error);

})

axios.interceptors.response.use( // 响应拦截器

response => {

if(response.status === 200) {

if(response.data.code == 0) {

return Promise.resolve(response);

} else if(response.data.code == 11 || response.data.code == 12) {

Toast({

message: '登录过期，请重新登录',

duration: 1000,

forbidClick: true

});

localStorage.removeItem('access\_token'); // 清除token

localStorage.removeItem('account'); // account

setTimeout(() => { // 跳转登录页面，并将要浏览的页面fullPath传过去，登录成功后跳转需要访问的页面

router.push({

path: '/login',

});

}, 1000);

} else {

Toast({

message: response.message,

duration: 1000,

forbidClick: true

});

}

} else {

return Promise.reject(response);

}

},

error => { // 服务器状态码不是200的情况

if(error.response.status) {

switch(error.response.status) {

// 401: 未登录

// 未登录则跳转登录页面，并携带当前页面的路径

// 在登录成功后返回当前页面，这一步需要在登录页操作。

case 401:

router.replace({

path: '/login',

query: {

redirect: router.currentRoute.fullPath

}

});

break;

// 403 token过期

// 登录过期对用户进行提示

// 清除本地token和清空vuex中token对象

// 跳转登录页面

case 403:

Toast({

message: '登录过期，请重新登录',

duration: 1000,

forbidClick: true

});

localStorage.removeItem('access\_token'); // 清除token

localStorage.removeItem('account'); // 清除account

setTimeout(() => { // 跳转登录页面，并将要浏览的页面fullPath传过去，登录成功后跳转需要访问的页面

router.replace({

path: '/login',

query: {

redirect: router.currentRoute.fullPath

}

});

}, 1000);

break;

case 404: // 404请求不存在

Toast({

message: '网络请求不存在',

duration: 1500,

forbidClick: true

});

break;

default: // 其他错误，直接抛出错误提示

Toast({

message: error.response.data.message,

duration: 1500,

forbidClick: true

});

}

return Promise.reject(error.response);

}

}

);

/\*\*

\* get方法，对应get请求

\* @param {String} url [请求的url地址]

\* @param {Object} params [请求时携带的参数]

\*/

export function get(url, params) {

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

axios.get(url, {

params: params,

})

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

export function get2(url, params) {

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

axios.get(url + '/' + params.param1, {})

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

/\*\*

\* post方法，对应post请求

\* @param {String} url [请求的url地址]

\* @param {Object} params [请求时携带的参数]

\*/

export function post(url, params) {

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

axios.post(url, QS.stringify(params))

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

export function post2(url, params) {

let urlParams

if(!params.param2) {

urlParams = url + '/' + params.param1;

} else {

urlParams = url + '/' + params.param1 + '/' + params.param2;

}

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

axios.post(urlParams, QS.stringify(params))

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

export function post3(url, params) {

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

axios.post(url, params)

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

/\*\*

\* put方法，对应put请求

\* @param {String} url [请求的url地址]

\* @param {Object} params [请求时携带的参数]

\*/

export function put(url, params) {

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

axios.put(url, params)

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

export function put2(url, params) {

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

axios.put(url, QS.stringify(params))

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

/\*\*

\* delete方法，对应delete请求

\* @param {String} url [请求的url地址]

\* @param {Object} params [请求时携带的参数]

\*/

export function Delete(url, params) {

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

axios.delete(url + '/' + params.param1, QS.stringify(params))

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}