说明 本项目中采用的登陆控制是spring oauth2,位于

```
com.github.wxiaoqi.security.auth.module.oauth.
              oauth2的接口简单说明
关于spring
```

864765462@qq.com 本项目采用密码认证方式 864765462@qq.com

• 注册准入客户端,并分配客户端权限

```
限password, refresh_token (表示该客户端具有自定义提交密码认证和刷新token的权限)。
    比如现在的vue前端,我们创建了一个vue 的客户端,其中我们分配了该客户端的权
     密码认证方式(PostMan举例)
```

864765462@qq.com 64765462 Type 864765 Clear 04765462 ". QQQ.com 864765462@qq.com Basic Auth The authorization header will be generated and added as a custom header Save helper data to request

```
864765462@qq.com
Password Password
Response COM
配置客户端id和secret(本质就是BasicAuth),会在请求头部形成如下信息

● 登录接口
                                                        364765462@qq.com
64765467
        POST
                  http://localhost:8765/api/oauth/token
6476546
```

864765462@dq.com ,64765462@qq.com Params Authorization • Pre-request Script Headers (3) 864165465@dd. Headers (2) 6476546 KEY 864765462@qq DESCRIPTION VALUE 64765461 application/x-www-form-urlencoded Content-Type 记得配置对应租户ID X-Tenant-Auth ac88ceb386aa4231b09bf472cb937c24

```
- Alue COM
                                                                                            Description
                                                                                  820 ddicom
           ▼ Temporary Headers (1) ①
6476546
                                   364765462@qq.com
                                                                         364765462@qq.co
              Authorization
647654620
                                                                         864765462@qq.com
                   http://localhost:8765/api/oauth/token
        配置表单提交参数
04765462
                                                                         864765462@,00.0
                                                                            DESCRIPTION
               Authorization •
64765467 none
               form-data x-www-form-urlencoded raw binary GraphQL BETA
                                          VALUE
                                  password
Value
username

password

v
                                                                            Description description
gra.
Key
                                                                            Status: 200 OK Time: 3.97s Size: 1.75 KB Save
        Body Cookies Headers (9) Test Results
        获取有效的token
```

864765462@qq.com 364765462@QQ 5462@qq.com add.com POST Y http://localhost:8765/api/auth/oauth/token Pre-request Script

647654626 462@dd.com 765462@qq.c form-data Bulk Edit x-www-form-urlencoded raw Key CO Value Description password grant_type username admin Status: 200 OK Time: 3214 ms Headers (14) □ Q Preview "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9 .eyJzdWIiOiJhZG1pbiIsInVzZXJfbmFtZSI6ImFkbWluIiwic2NvcGUiOlsicmVhZCJdLCJleHBpcmUiOjE1MjIzMDI5Njg5NTUsImV4cCI6MTUyMjMxMzc2OCwiZGVwYX J0Ijoicm9vdCIsInVzZXJOYW1lIjoiTXIuQUciLCJ1c2VySWQi0iIxIiwiYXV0qG9yaXRpZXMi0lsiUk9MRV9VU0VSIl0sImp0aSI6IjU3MGQxZjM1LWU3OTAtNDk3MC1iN 2VhLTZjZTdmNGFmZGZmMSIsInRlbmFudCI6ImFjODhjZWIzODZhYTQyMzFiMDliZjQ3MmNi0TM3YzI0IiwiY2xpZW50X2lkIjoidnVlIn0 .dKXw7jYatJiXpYbRhQkyJBLgLK7kWER_\$WB5pFk4B5Ni9fY0Idnz6jMGQA9 -UisYDLECJkhh8CngogiF0GeQgBnoXlTeQhXD2nBf0RqgiQZSETSq_WDPkbjdy95BIZHFsSK9_WHnD54JDnkwFj3IQNvcHWiSmrwSoGk27NznEyI", 'refresh_token": "eyJhbGci0iJSUzI1NiIsInR5cCI6IkpXVCJ9 .eyJzdWIiOiJhZG1pbiIsInVzZXJfbmFtZSI6ImFkbWluIiwidXNlck5hbWUiOiJNci5BRyIsInVzZXJJZCI6IjEiLCJhdXRob3JpdGllcyI6WyJST0xFX1VTRVIiXSwiY2 xpZW50X2lkIjoidnVlIiwic2NvcGUiOlsicmVhZCJdLCJleHBpcmUiOjE1MjIzMDI5Njg5NTUsImF0aSI6IjU3MGQxZjM1LWU3OTAtNDk3MC1iN2VhLTZjZTdmNGFmZGZmM

SIsImV4cCI6MTUyNDcyNDg0NiwiZGVwYXJ0Ijoicm9vdCIsImp0aSI6Ijdl0DVkYmQ3LTQ3MWUtNGVlZS05YzQ4LWU3ZTMwM2MzMDZhYSIsInRlbmFudCI6ImFj0DhjZWIz

-n3BuFcmw4Cl5CTXscHnxqgoFkJ2X0lE6jYpD9zmSagw7nofjvVa8_Spu3V0VBFcymU2weYYdHcNIxzHjIiW0MlgfJvCLJKgaYX83QB015Uz8PuHJtiBVd0G7SW9SEzJe4N

ODZhYTQyMzFiMDliZjQ3MmNiOTM3YzI0In0.D9DXBlRjt88X_CYs26XsknNgBbOcQzHIaj9

j3ewA", 'expires_in": 13708, "scope": "read", 'sub": "admin",

"depart": "root", "userName": "Mr.ÁG", "userId": "1",

▶ build

■ config

Js prod.env.js

Js sit.env.js

Js login.js

'expire": "2018-03-29 13:56:08",

凭借token访问后端信息

▶ 测试demo服务 http://localhost:8766/sampleDataTest/all Pre-request Script Params Authorization Headers (9) Settings ▼ Headers (2) DESCRIPTION Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWliOiJhZ.. Authorization X-Tenant-Auth ac88ceb386aa4231b09bf472cb937c24 Temporary Headers (7) Status: 200 OK Time: 3.27s Body Headers (9) Test Results Visualize **BETA** 前端自定义登录(框架 https://panjiachen.github.io/vue-element-admin-site/zh/guide/#contribution

配置路径api 开发环境路径转发,config/index.js ,生产环境我们直接配置nginx 转发

assetsPublicPath: '/'

pathRewrite: { \'^/api': '/'

autoOpenBrowser: true, errorOverlay: true, notifyOnErrors: false,

loginByUsername(username, userInfo.password).then(response => {

assetsSubDirectory: 'static',

target: 'http://localhost:9777',

host: 'localhost', // can be overwritten by process.env.HOST

port: 9527, // can be overwritten by process.env.PORT, if port is in use, a free one will b

poll: false, // https://webpack.js.org/configuration/dev-server/#devserver-watchoptions-

dev: {

// Paths

proxyTable: { '/api':{

10

13

16 ⊟

19

♥ src/views/login/index vue 登录界面,定位登录方法handleLogin

this.\$store.dispatch('LoginByUsername', this.loginForm)

const username = userInfo.username.trim()

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

if (data.access token === '') {

message: '账户或密码错误!',

return Promise.reject('error')

• src/store/modules/user.js 代码中的LoginByUsername

LoginByUsername({ commit }, userInfo) {

const data = response

type: 'warning'

Message({

} else {

resolve()

}).catch(error => {

reject(error)

// respone拦截器

11

864765462@0

```
const token = 'Bearer' + data.access token;
setToken(token)
commit('SET TOKEN', token);
```

```
{前端工程}/src/utils/request.js 利用axios 对前端访问后端的接口自动加上token,同时
  根据后端不同状态进行登出等控制
// request拦截器
service.interceptors.request.use(config => {
  // Do something before request is sent
if (store.getters.token) {
    config.headers.Authorization = getToken(); // 让每个请求携带token--
 ['Authorization']为自定义key 请根据实际情况自行修改
  } else {
    config.headers.Authorization = 'Basic dnVlOnZ1ZQ=='; // 增加客户端认证
  config.headers['X-Tenant-Auth'] = getTenant(); // 增加租户ID
  config.url = config.url.replace('{tenant}', getTenant())
return config
 }, error => {
// Do something with request error
  console.log(error) // for debug
Promise.reject(error)
```

service.interceptors.response.use(response => { const res = response.data if (response.status !== 200 && res.status !== 200) { Message({ message: res.message, type: 'error', duration: 5 * 1000 }) } else { return response.data error => { console.log('err' + error) // for debug const response = error.response if (response === undefined) { Message({ message: '服务请求超时!' type: 'error', duration: 5 * 1000 }) return Promise.reject(error) const info = response.data
if (""") if (response.status === 401 || info.status === 40101) { MessageBox.confirm('你已被登出,可以取消继续留在该页面,或者重新登录', '确 定登出', { confirmButtonText: '重新登录', cancelButtonText: '取消', type: 'warning' }).then(() => { store.dispatch('FedLogOut').then(() => { location.reload() // 为了重新实例化vue-router对象 避免bug }) if (response.status === 400 || info.status === 40001) { Notification.warning({ title: '用户密码错误! ', message: info.message, type: 'error', duration: 2 * 1000 return Promise.reject('error')
} if (response.status === 403) { Notification.warning({ ˈtitle: '禁止', message: info.message, type: 'error', duration: 2 * 1000

return Promise.reject('error')

if (info.status === 30101) {

title: '失败',

type: 'error',

type: 'error',

duration: 5 * 1000

message: info.message,

return Promise.reject(error)

● 过滤某些路劲不做拦截,修改application.yml

● 认证拦截代码,ResourceServerConfigurer

public class ResourceServerConfigurer {

private String startWith;

public SecurityWebFilterChain

http.authorizeExchange()

http.csrf().disable();

return http.build();

@Value("\${gate.ignore.startWith}")

springSecurityFilterChain(ServerHttpSecurity http) {

http.oauth2ResourceServer().jwt();

● 如果后端某些路径不需要做登录拦截,修改application.yml

ignoreResources: #配置哪些路径不做登录控制

● 用户信息初始化代码片段,UserAuthRestInterceptor

throw new NonLoginException("登录过期");

BaseContextHandler.setName(user.getName());

BaseContextHandler.setUserID(user.getId());

BaseContextHandler.setUsername(user.getUsername());

BaseContextHandler.setDepartID(user.getDepartId());

BaseContextHandler.setTenantID(user.getTenantId());

Object principal = SecurityContextHolder.getContext()

.getAuthentication()

User user = userBiz.getUserByUsername(String.valueOf(principal));

.getPrincipal();

.anyExchange().authenticated();

/actuator/**,/api/oauth/token,/api/admin/tenant/all,/static/**,/api/admi

n/wf/editor/stencilset,/api/code/generator/build,/api/admin/generator/bu

.pathMatchers(startWith.split(",")).permitAll()

duration: 5 * 1000

type: 'error',

return Promise.reject(error)

})

Message({

Message({

})

Gateway网关

ignore:

startWith:

@EnableWebFluxSecurity

@Bean

自定义服务

security:

- /v2/api-docs

- /static/**

- /favicon.ico

/webjars/**

- /actuator/**

if(principal==null){

如何在Service中获取当前用户

- /error

- /swagger-resources/**

- /admin/generator/build

- /admin/groupType/all

- /admin/tenant/all

后端项目

gate:

ild

duration: 2 * 1000

Notification.warning({

message: info.message,

return Promise.reject('error')

message: '后端服务异常,请联系管理员!',

864765462@qq.com

if (response.status === 504) {

- /swagger-ui.html - /admin/wf/editor/stencilset - /.well-known/jwks.json - /api/log/** - /api/permissions

String username = BaseContextHandler.getUsername(); String userName = BaseContextHandler.getName(); String userId = BaseContextHandler.getUserID(); String departId = BaseContextHandler.getDepartID(); String tenantId = BaseContextHandler.getTenantID();

```
● 服务之间token和传递,cloud-common中的OkHttpTokenInterceptor
        @Component
        @Log
        public class OkHttpTokenInterceptor implements Interceptor {
            @Override
            public Response intercept(Chain chain) throws IOException {
                Request newRequest = chain.request();
                if (StringUtils.isNotBlank(BaseContextHandler.getToken())) {
                    newRequest = chain.request()
                           .newBuilder()
                           .header(RequestHeaderConstants.TENANT,
        BaseContextHandler.getTenantID())
                           .header(RequestHeaderConstants.AUTHORIZATION,
        BaseContextHandler.getToken())
                           .build();
return response;
                                                         8647654620
                           86/165/167
                Response response = chain.proceed(newRequest);
                           86476546200q.com
                                                         8647654620qq.com
34765462039.com
                           864765462@qq.com
                                                         864765462009.com
                           864765462@qq.com
                                                         864765462@qq.com
64765462009
64.765462@qq.com
                           864765462@qq.com
                                                         864765462@qq.com
```