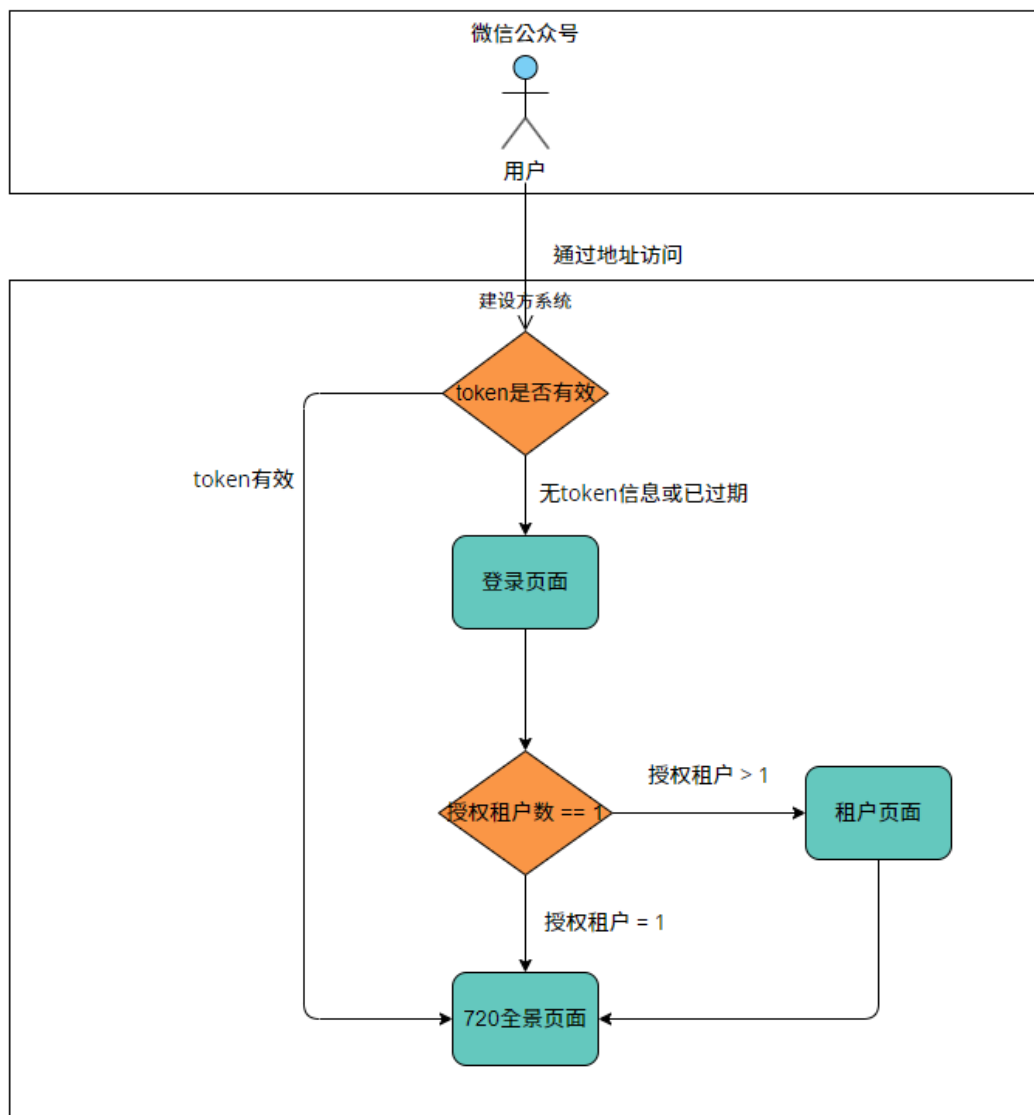


微信公众号720全景开发技术文档

0、背景

为了满足用户更便捷查看720全景数据，现增加微信公众号的入口让用户快速进入720全景模块，并支持手机端用户账号登录时的权限识别与控制。

1、页面设计逻辑



2、用户数据存储

- 采用localStorage存储用户的信息。
- 存储内容：
 - userToken: 用户token信息;
 - exptime: userToken的过期时间，用于判断userToken是否过期;
 - username: 用户名账号;
 - isOnlyOneTenant: 当前账号是否只授权了一个租户;
 - tenantId: 用户上一次操作中选择租户id;

- 清除用户数据：在微信中通过退出微信号来清除公众号中存储的数据，重新登录微信即可恢复到初始状态。

3、全局前置路由守卫

重要参数：**isDirectlyLoad** - 用于区分已有的app端h5页面和微信公众号端的h5页面，app端走原生的接口，微信公众号不依赖原生接口，使用路由跳转页面；此参数需要一直放在url上。

```
vue.prototype.$moment = moment;
// 初始化一些配置
startupParameterConfig(getUrlVar, api);
const isDirectlyLoad = getUrlVar('isDirectlyLoad');
// 对于打过来的路由进行解析
const initVue = async () => {
  router.beforeEach((to, from, next) => {
    const token = getUrlVar('token');
    const tenantId = getUrlVar('tenantId');
    // 清除旧的用户Info并设置新的token和tenantId
    sessionStorage.removeItem('userInfo');
    sessionStorage.setItem('token', token);
    sessionStorage.setItem('tenantId', tenantId);
    const userToken = getUrlVar('userToken');
    const theme = getUrlVar('theme');
    store.commit('THEME', theme);
    sessionStorage.setItem('userToken', userToken);
    // h5单页面
    if (!isDirectlyLoad) {
      next();
    } else {
      // 直接加载的路由方式走单独的逻辑
      initDirectlyLoadRoute(to, from, next);
    }
  });
  // 判断是否是直接加载
  if (!isDirectlyLoad) {
    const userInfo = await api.getCurrentUserInfo();
    window['userInfo'] = userInfo || {};
    sessionStorage.setItem('userInfo', JSON.stringify(userInfo));
    window['entryUserInfo'] = userInfo || {};
  }
}
```

```

/**
 * 直接加载路由方式的前置路由设置
 */
export const initDirectlyLoadRoute = (to, from, next) => {
  const userToken = localStorage.getItem('userToken'); // 获取token信息
  const exptime = localStorage.getItem('exptime'); // 获取token到期时间
  const tenantId = localStorage.getItem('tenantId'); // 获取tenantId信息
  const date = new Date().getTime().toString().slice(0, 10);
  const isValid = Number(date) < Number(exptime); // 判断token是否过期
  if (
    userToken !== '' &&
    userToken !== 'null' &&
    userToken !== 'undefined' &&
    isValid
  ) {
    //userToken存在且有效
    //设置新的token和tenantId
    localStorage.setItem('userToken', userToken);
    localStorage.setItem('tenantId', tenantId);
    if (!to.meta.hasEntry) {
      // 判断路由上是否已经登录过,避免进入死循环
      to.meta.hasEntry = true;
      let newTo = { ...to };
      newTo.query.userToken = userToken;
      if (
        (tenantId == '' || tenantId == 'null' || tenantId == 'undefined') &&
        to.name !== 'PanoramicImageProgress720_TenantList'
      ) {
        // 没有存储租户信息,直接跳到登录页面
        next({
          name: 'PanoramicImageProgress720_PanoramicImageProgress720Main',
          query: { isDirectlyLoad: 'true' },
        });
        return;
      } else {
        // 有租户信息
        to.meta.hasEntry = true;
        newTo.query.tenantId = tenantId;
        newTo.query.userToken = userToken;
      }
    }
  }
}

```

```

1  if (!to.meta.hasEntry) {
2      // 判断路由上是否已经登录过,避免进入死循环
3      to.meta.hasEntry = true;
4      let newTo = { ...to };
5      newTo.query.userToken = userToken;
6      if (
7          (tenantId == '' || tenantId == 'null' || tenantId == 'undefined') &&
8          to.name !== 'PanoramicImageProgress720_TenantList'
9      ) {
10         // 没有存储租户信息,直接跳到登录页面
11         next({
12             name: 'PanoramicImageProgress720_PanoramicImageProgress720Main',
13             query: { isDirectlyLoad: 'true' },
14         });
15         return;
16     } else {
17         // 有租户信息
18         to.meta.hasEntry = true;
19         newTo.query.tenantId = tenantId;
20         newTo.query.userToken = userToken;
21         next(newTo);
22         return;
23     }
24 } else {
25     next();
26 }
27 } else {
28     //userToken无效
29     if (to.name == 'PanoramicImageProgress720_PanoramicImageProgress720Main') {
30         next();
31     } else {
32         // 跳到登录页面
33         next({
34             name: 'PanoramicImageProgress720_PanoramicImageProgress720Main',
35             query: {
36                 isDirectlyLoad: 'true',
37             },
38         });
39     }
40 }
41 }
42 };

```

4、请求头的修改-apiConfig.js

兼容app端和微信公众号的逻辑。

```

1  axios.defaults.withCredentials = true;
2  // 封装请求拦截
3  axios.interceptors.request.use(
4      (config) => {
5          const token = getUrlVar('token'); // 获取token
6          const userToken =
7              getUrlVar('userToken') || localStorage.getItem('userToken'); // 获取userToken
8          if (!config.url.indexOf('myhuaweicloud') > 0) {
9              config.headers['Content-Type'] = 'application/json;charset=UTF-8';
10             // 处理get方法 请求头中没有Content-Type
11             if (config.method === 'get') {
12                 config.data = { unused: 0 };
13             }
14         }
15     }
16 );

```

5、微信公众号访问地址

本地调试:

http://localhost:8080/index#/PanoramicImageProgress720_PanoramicImageProgress720?isDirectlyLoad=true

微信公众号跳转地址:

http://bimdev.glodon.com/static/gcjg_quality_field_h5/index.html#/PanoramicImageProgress720_PanoramicImageProgress720?isDirectlyLoad=true

https://jsf.glodon.com/static/gcjg_quality_field_h5/index.html#/PanoramicImageProgress720_PanoramicImageProgress720?isDirectlyLoad=true