

与服务端协议

跨域

跨域使用服务代理转发方案解决，所以在部署跟服务启动时注意配置好相应的服务地址；

Token

需要Token的请求，在请求头添加Authorization属性，Token短时间到期后台会给一个新Token，长时间到期则会登录失效（这部分在请求拦截器中已处理）

参数

- 1、请求统一使用JSON格式传参；
- 2、参数返回格式

```
1 {  
2   code: "SUCCESS", //业务编码  
3   message: "请求描述信息或错误信息",  
4   data: "String || Array || Object || *", //业务数据  
5 }
```

- 3、code编码说明

```
1 SUCCESS    //成功  
2 TOKEN_EXPIRE //登录失效 出现这个编码就需要重新登录了  
3  
4 // 以上是比较关注的code 其他业务错误请参考 后台服务的编码说明  
5
```

请求拦截器

参照了Axios框架，对uni.request进行了简单封装msbUniRequest.js 具体使用请查看其使用文档

 [msbUniRequest.js使用说明](#)

业务中的请求处理规范

1. 所有的请求必须统一放到common/api文件夹进行统一管理
 2. 所有请求方法命名以（Api+请求类型+具体方） 法命名
 3. 必须使用ToAsyncAwait进行结果包装； [ToAsyncAwait使用说明](#)
- 不允许使用try catch 和 then 处理返回结果

示例

```
1  xxapi.js
2  /**
3   * 获取订单
4   * @param {*} params
5   */
6
7  export const ApiGetOrderList = (params) =>
8      ToAsyncAwait(msbRequest.get('/app/oreder', params));
9
10
11  xx.vue
12  exprot default {
13  //....
14  methods:{
15      async getOrderList(){
16          const {error, result} = await ApiGetOrderList();
17          if(error){
18              //错误处理
19              console.log(error);
20              return false;
21          }
22          //正常业务
23          this.orderList = result;
24      }
25
26  }
27  //...
28
29
```

