

OA登录接入指引

oa登录接入流程

1、统一登陆接入方式有以下两种

① 独立登录态接入方案：

业务站点维护自己的Session与Cookie，业务站点内所有请求由自己校验登录态。

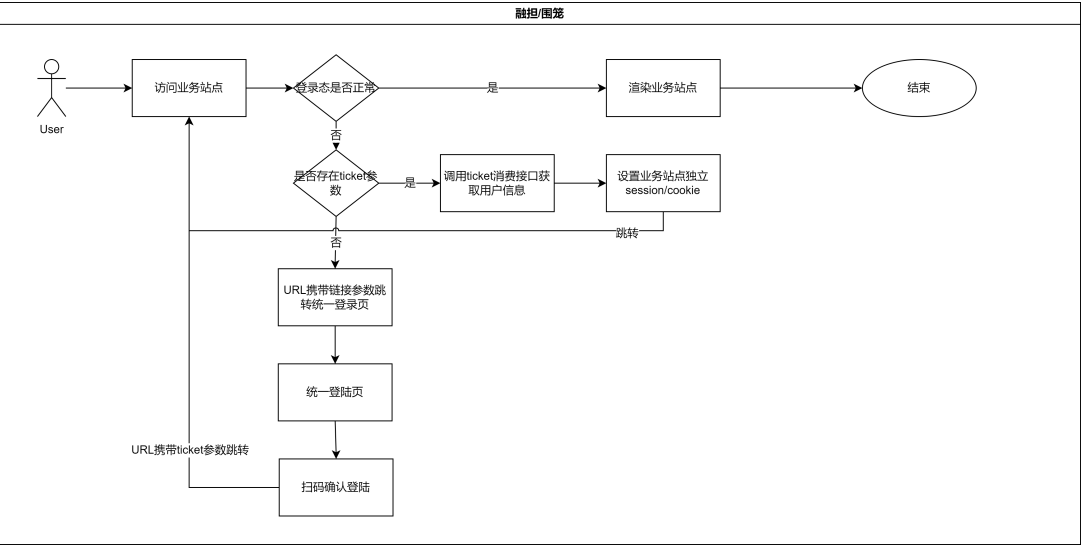
② 共享Cookie方案：

业务站点不维护登录态，共享统一登陆站点cookie，站点内所有请求通过网关（OA风控网关）校验登录态，要求网关API的域名与统一登陆同域，调用网关接口时浏览器可携带统一登陆的登录态。

两种方案，业务可根据实际情况，选择改造方案，减少改动量。

2、统一登陆交互流程

① 独立登录态接入方案



业务站点逻辑：

登录态判断

业务站点根据站点cookie判断登录态是否正常

ticket判断

登录态异常，判断URL上是否有统一登陆ticket

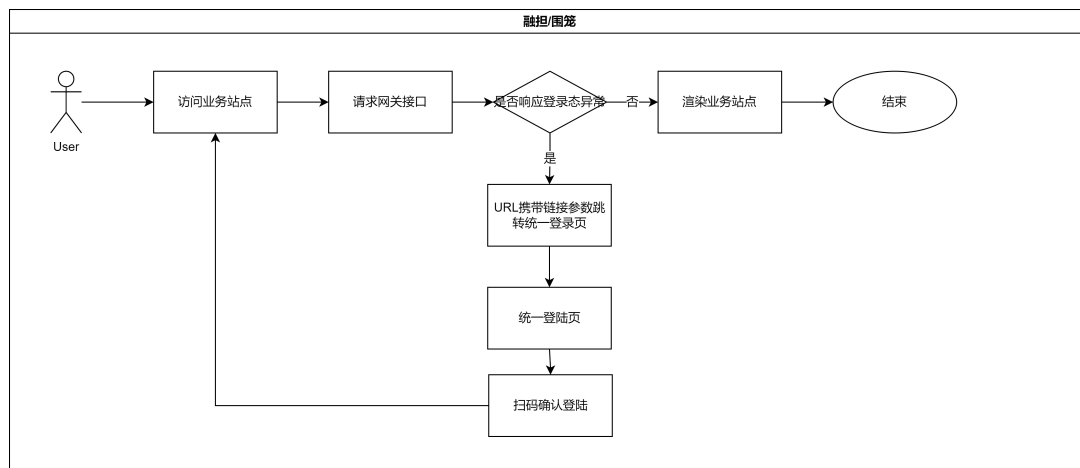
ticket消费及登录态设置

-> 有ticket则调用站点接口，传递ticket到站点后端，后端调用统一登陆ticket消费接口换取登陆用户信息，设置站点的session和cookie。

-> 无ticket则将站点域名urlencode后拼接到统一登陆链接后作为get参数跳转。如：

<https://oa.fenqile.com/passport?url=https%3A%2F%2Frd.oa.fenqile.com%2F>

② 共享Cookie方案



业务站点逻辑：

登录态判断

调用网关SESSION接口，判断响应retcode是否为登陆态异常

跳转统一登陆

-> 将站点域名urlencode后拼接到统一登陆链接后作为get参数跳转。如：

<https://oa.fenqile.com/passport?url=https%3A%2F%2Frd.oa.fenqile.com%2F>

③ 应用接入

通过「[开放平台管理系统](#)」申请应用接入，申请成功后通过 `appkey` 和 `appid` 进行接口签名认证。



开发者需要 **注意**，`appkey` 和 `appid` 为系统自动生成，不同环境申请的应用不能互通。目前公司研发环境分为 `开发环境` 和 `生产环境`（预发布、灰度、线上、OA）。

④ 生成签名

我们通过请求签名的方式来确保请求来源的真实性，签名生成的 `token` 会被放置于请求参数中一起发送。

签名规则

签名由 `timestamp`、`rand`、`appid` 三个固定参数加密得到，开放平台统一约定，使用 `appkey` 作为密钥进行 AES 加密算法。

参数名称	参数类型	示例	说明
timestamp	number	1529055300	当前时间戳
rand	number	92731603500	随机数
appid	number	13037645	应用ID

签名示例

开发者需要 **注意**，拼接字符串时，`timestamp`、`rand`、`appid` 三个参数的顺序。

```
1 encrypt("timestamp=$timestamp&rand=$rand&appid=$appid", $appkey)
```

JAVA

```
1
2 /**
3  * 加密方法
4  *
5  * @param data 要加密的数据
6  * @param key 加密key
7  * @return 加密的结果
8  * @throws Exception
9  */
10 public static String encrypt(String data, String key) throws Exception {
11     Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding");//"算法/模式/补
    码方式"
12     int blockSize = cipher.getBlockSize();
13
14     byte[] dataBytes = data.getBytes();
15     int plaintextLength = dataBytes.length;
16     if (plaintextLength % blockSize != 0) {
17         plaintextLength = plaintextLength + (blockSize - (plaintextLength
    % blockSize));
18     }
19
20     byte[] plaintext = new byte[plaintextLength];
21     System.arraycopy(dataBytes, 0, plaintext, 0, dataBytes.length);
22
23     SecretKeySpec keyspec = new SecretKeySpec(key.getBytes(), "AES");
24     IvParameterSpec ivspec = new IvParameterSpec(key.getBytes());
25
26     cipher.init(Cipher.ENCRYPT_MODE, keyspec, ivspec);
27     byte[] encrypted = cipher.doFinal(plaintext);
28
29     return new Base64().encodeToString(encrypted);
30 }
```

PHP

```
1 // PHP-签名算法
2 function encrypt($body, $appkey)
3 {
4     return base64_encode(mcrypt_encrypt(MCRYPT_RIJNDAEL_128, $appkey, $bod
    y, MCRYPT_MODE_CBC, $appkey));
5 }
```

涉及的改造点

① 独立登录态接入方案

跳转统一登陆站点的URL和消费ticket的URL需要替换，乐信科技、乐信融担、盟天融担、朴道围笼对应的URL都不一样，需要抽到配置中心之类的管理。

② 共享Cookie方案

跳转统一登陆站点的URL需要替换，乐信科技、乐信融担、盟天融担、朴道围笼对应的URL都不一样，需要抽到配置中心之类的管理。

调用网关接口的域名需要替换，乐信融担、盟天融担、朴道围笼对应的域名都不一样，需要抽到配置中心之类的管理。