酷哇商城网页登录接入 SSO

> 适用范围：酷哇商城网页端登录、回调和本地会话建立。
接入模式：OAuth2 授权码流程。

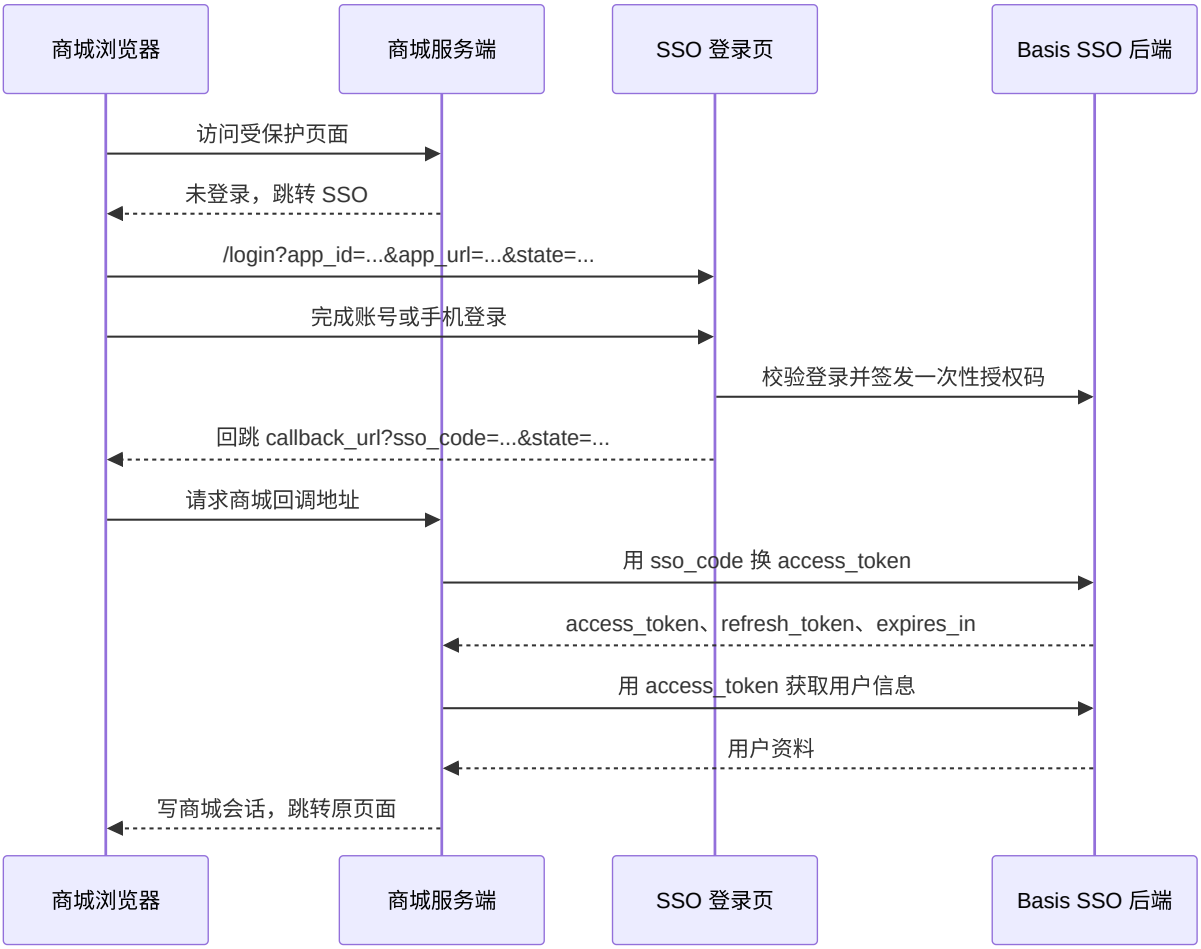
1. 接入前准备

向 SSO/Basis 管理员申请并确认以下配置：

配置	说明
`client_id`	商城专用应用 ID，同时作为登录地址中的 `app_id`
`client_secret`	商城服务端密钥，绝不能下发浏览器
`tenant`	当前联调值为 `cowarobot`
`callback_url`	商城固定回调地址，例如 `https://mall.example.com/sso/callback`
`SSO_URL`	SSO 登录页地址，例如 `https://sso.softtest.cowarobot.com`
`BACKEND_API`	Basis SSO 后端地址，例如 `https://internal.softtest.cowarobot.com`

`callback_url` 必须提前登记，且协议、域名、端口和路径必须完全一致。生产环境使用 HTTPS；`client_secret` 只通过部署平台或密钥管理系统注入商城服务端。

2. 登录流程



商城只需要完成四件事：生成并保存 `state`、构造登录跳转地址、在服务端用 `sso_code` 换令牌、建立商城自己的会话。

3. 实现步骤

3.1 跳转到 SSO

用户未登录时，浏览器跳转到：

```
{SSO_URL}/login?app_id={client_id}&app_url={encodeURIComponent(callback_url)}&state={state}
```

示例：

```
https://sso.softtest.cowarobot.com/login?app_id=
<client_id>&app_url=https%3A%2F%2Fmall.example.com%2Fsso%2Fcallback&state=<random_value>
```

`app_url` 必须来自商城固定配置，不能直接使用用户传入的跳转地址。每次登录都要使用浏览器安全随机数生成新的 `state`，保存到 `sessionStorage` 或服务端会话；`state` 不要包含用户信息、令牌或跳转地址。

3.2 接收回调

登录成功后，SSO 会回跳：

```
https://mall.example.com/sso/callback?sso_code=<one_time_code>&state=<random_value>
```

回调处理器必须先校验 `state`：它必须存在、与本次登录保存的值完全一致，并在通过后立即删除。缺失、不一致或已使用时停止登录并让用户重新发起。校验通过后，立即在**商城服务端**用 `sso_code` 换取令牌；同一个 `sso_code` 只能使用一次，失败后不要重试同一授权码。

3.3 服务端换取令牌

```
POST {BACKEND_API}/api/v1/basis/sso/access_token
Content-Type: application/json

{
  "client_id": "<client_id>",
  "client_secret": "<server_only_secret>",
  "code": "<sso_code>",
  "grant_type": "authorization_code",
  "tenant": "cowarobot"
}
```

成功响应形态：

```
{
  "code": 200,
  "data": {
    "access_token": "<access_token>",
    "refresh_token": "<refresh_token>",
    "expires_in": "7200"
  }
}
```

商城应同时校验 HTTP 状态、业务状态和 `data.access_token`。响应外层 `code: 200` 是业务状态码；回调参数中的 `sso_code` 是授权码，二者不能混用。

3.4 获取用户信息

换取令牌成功后，由**商城服务端**调用：

```
{BACKEND_API}/api/v1/basis/sso/user_info?access_token={access_token}
```

`access_token` 为 SSO 颁发的访问令牌，必须由商城服务端传递，不得下发到浏览器或记录到日志。接口返回示例：

```
interface UserInfoResponse {
  code: number;
  data: {
```

```
    account: string;
    email: string;
    id: number;
    job_title: string;
    key: string;
    phone: string;
    status: string;
    tntkey: string;
    user_name: string;
  };
  msg: string;
}
```

商城应校验业务状态和 `data.key`。`data.key` 是用户绑定的稳定标识；`account`、`user_name`、`email`、`phone`、`job_title`、`status`、`tntkey` 与 `key` 可按商城业务需要同步或展示。不要仅按姓名或手机号创建绑定关系。

3.5 建立商城会话

推荐做法是：令牌只保存在商城服务端会话存储，浏览器只保存随机会话 ID。登录成功后使用 `303` 跳回登录前页面或商城首页。

如果现有架构必须把令牌放入 Cookie，至少设置：

```
HttpOnly; Secure; SameSite=Lax; Path=/
```

Cookie 有效期不应长于 `expires_in`。不要把 `access_token`、`refresh_token` 或 `client_secret` 放入 localStorage、sessionStorage、URL 或浏览器可读 Cookie。

4. 商城需要补齐的能力

项目	接入要求
用户绑定	使用 `user_info` 返回的 `data.key` 绑定会员或员工账号，不能只按姓名、手机号或账号匹配
用户信息	已提供 `/api/v1/basis/sso/user_info`；商城服务端用 SSO 颁发的 `access_token` 获取资料
令牌刷新	确认 refresh token 的请求字段、轮换和失效规则后，由商城服务端实现
退出登录	至少清除商城本地会话；全局退出或令牌吊销需以 Basis 正式接口为准
`state` 防护	商城每次登录生成高随机值并保存；SSO 会原样回传，商城在换令牌前完成一次性校验，用于防御登录 CSRF

5. 联调检查清单

- ☐ 已登记测试和生产 `callback_url`，并取得商城专用 `client_id` / `client_secret`。
- ☐ 未登录访问商城受保护页面时，能正确跳转到 SSO。
- ☐ 账号登录、手机登录都能回跳商城，并同时携带一次性 `sso_code` 与原始 `state`。
- ☐ 回调 `state` 缺失、不一致或重复使用时，商城拒绝换令牌并要求重新登录。
- ☐ 商城服务端能换取 `access_token`，浏览器网络请求和前端代码中没有 `client_secret`。
- ☐ 商城服务端能用 `access_token` 调用 `user_info`，并以 `data.key` 完成用户绑定。
- ☐ 刷新页面后登录态正常；令牌或会话过期后会重新登录，不出现循环跳转。
- ☐ 退出后旧会话不能继续访问商城。
- ☐ 日志中不记录授权码、密钥、访问令牌、刷新令牌或手机号等敏感明文。

6. 常见问题

现象	优先排查
回调缺少 `sso_code`	`callback_url` 是否正确登记，登录跳转地址是否完整编码
换令牌返回 401	`client_id`、`client_secret`、`tenant`、回调地址是否匹配；授权码是否已使用或过期
换令牌成功但商城仍未登录	Cookie 的 `Secure` / `SameSite` / 域名设置，以及商城服务端会话读取逻辑
无法展示用户信息	检查 `access_token` 是否有效，以及 `user_info` 的业务状态和 `data.key` 是否正确返回