

用户中心集成方案

关于本文档

文档名称	用户中心集成方案				
说 明	本文档用于各个渠道系统与用户中心集成方案设计和开发				
适用对象	各个渠道系统相关管理人员、设计人员、开发人员、测试人员				
文 件 名	用户中心集成方案设计文档.doc				
修订历史					
版本	章节	类型	日期	作者	说明
1.0.0	全部	创建	2015-11-04	施向伟	
2.0.0	全部	更新	2016-01-20	施向伟	
2.0.1	全部	更新	2016-02-02	施向伟	
2.1.0	全部	更新	2016-04-07	施向伟	1. 绑卡、实名认证接口交由核心系统提供 2. 增加邮箱、证件号码登录 3. 完善部分接口信息
2.1.1	部分	更新	2016-04-14	施向伟	1. 注册接口增加userRegDevice参数 2. 增加绑定手机号唯一性校验接口
2.1.1	部分	更新	2016-05-12	施向伟	1. 增加财富经理、推荐人与用户绑定
2.2.0	部分	更新	2016-08-15	施向伟	1. 增加登录锁定 2. 增加Token延时
2.2.1	部分	更新	2016-08-25	施向伟	1. 增加登录锁定处理逻辑描述 2. 增加错误代码表

目 录

1. 前言.....	1
1.1. 文档编写目的.....	1
1.2. 内容摘要.....	1
2. 总体设计.....	2
3. SSO 单点登录流程图（以 i 投为例）	3
3.1. i 投第一次访问	3
3.2. i 投第二次访问	4
3.3. 其它系统第一次访问.....	4
3.4. 单点退出.....	5
4. SSO 单点登录集成接口	5
4.1. SSO 认证 Service	5
4.2. SSO 重定向 URL	10
5. 用户中心对接流程图.....	12
6. 用户中心集成接口.....	13
6.1. 用户注册接口.....	13
6.2. 修改密码接口.....	15
6.2.1. 根据用户 Id 修改密码接口：	15
6.2.2. 根据绑定手机号修改密码接口：	16
6.3. 获取用户资料接口.....	18
6.4. 用户安全绑定唯一性校验接口.....	22
6.4.1. 用户登录名唯一性校验接口：	22
6.4.2. 绑定手机号唯一性校验接口：	23
6.4.3. 绑定邮箱唯一性校验接口：	24
6.4.4. 绑定证件号码唯一性校验接口：	25
6.5. 用户手机号绑定接口.....	26
6.6. 用户邮箱绑定接口.....	28
6.7. 用户财富经理(客户经理)绑定接口	29

7. 错误代码表.....	30
---------------	----

1. 前言

1.1. 文档编写目的

解决渠道系统集成到单点登录系统的问题、集成统一用户中心。

1.2. 内容摘要

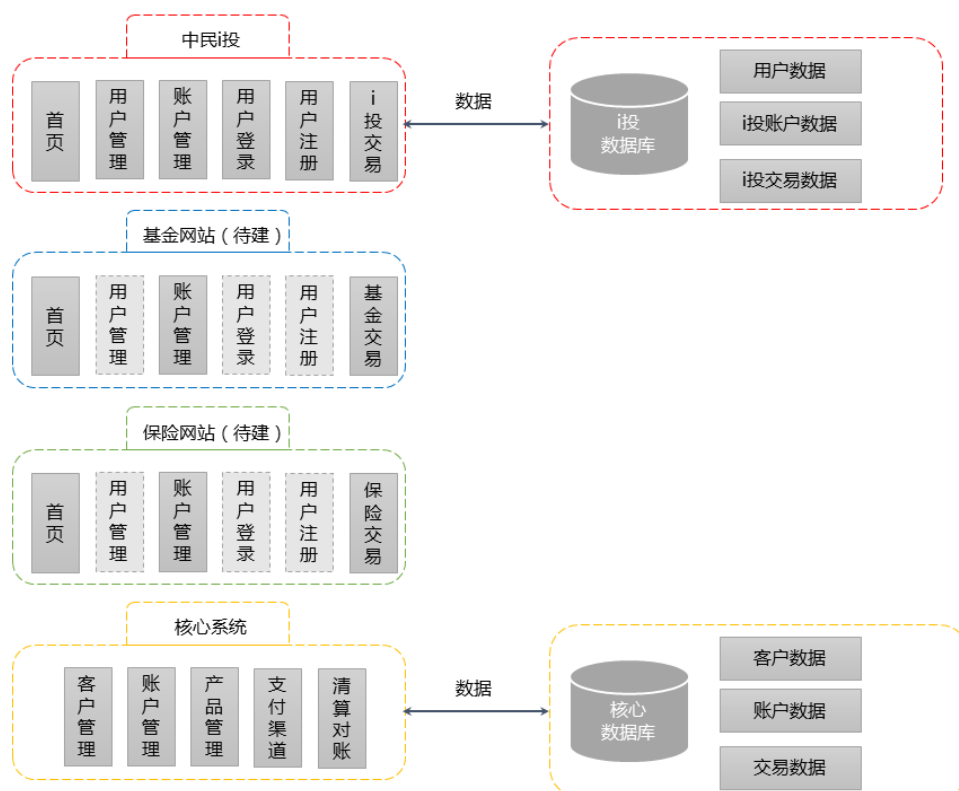
本文详细说明了客户端的集成方案的设计和接口，包括集成中部分细节的处理。

2. 总体设计

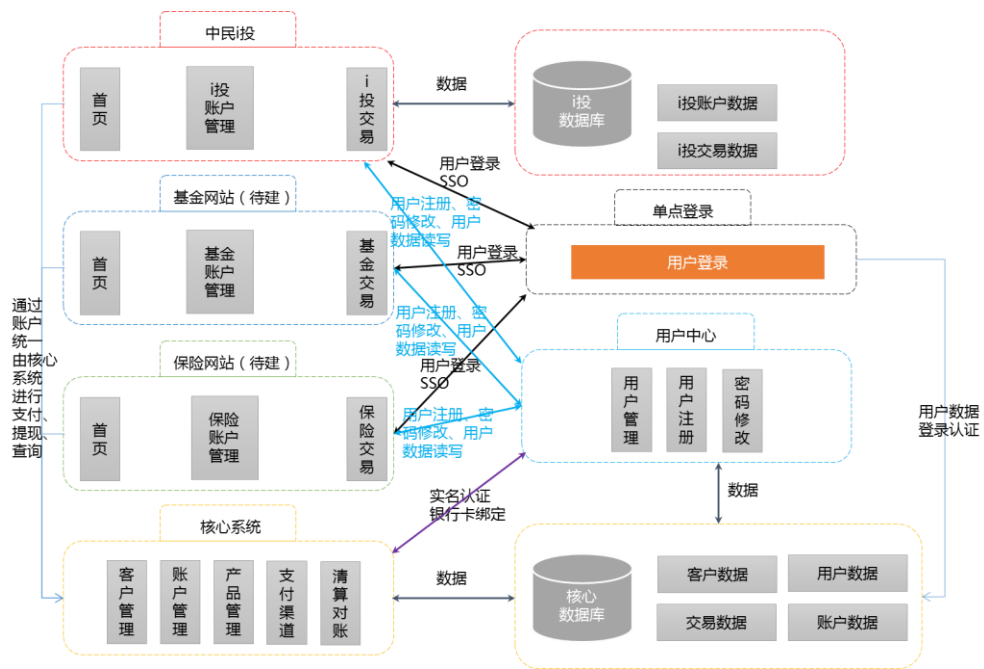
建立 SSO 单点登录认证中心，所有站点都通过该认证中心验证用户名密码，用户数据源还是来自于用户中心，登录成功后 SSO 返回用户中心用户 uid，渠道系统可根据 uid 调用用户中心接口获取该用户最新的用户数据。

为保持页面风格一致性，用户注册、密码修改、用户信息获取和修改通过渠道系统自己的前端入口，由渠道系统后端调用用户中心对应接口，写入和获取相应数据。

目前系统结构图，如下图

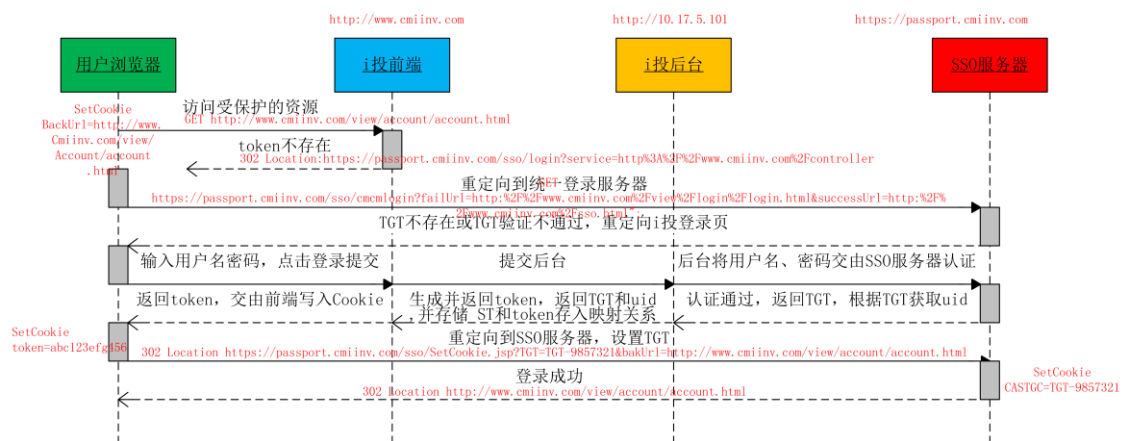


统一用户和认证之后，见下图：

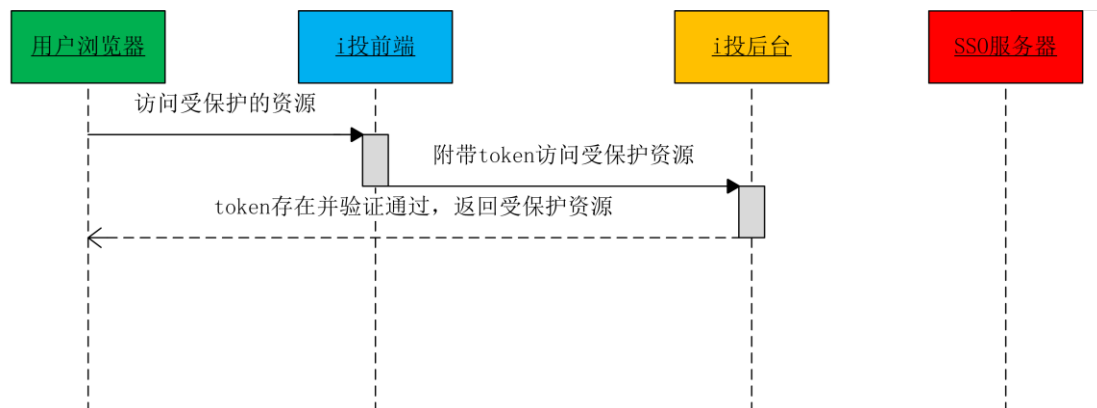


3. SSO 单点登录流程图（以 i 投为例）

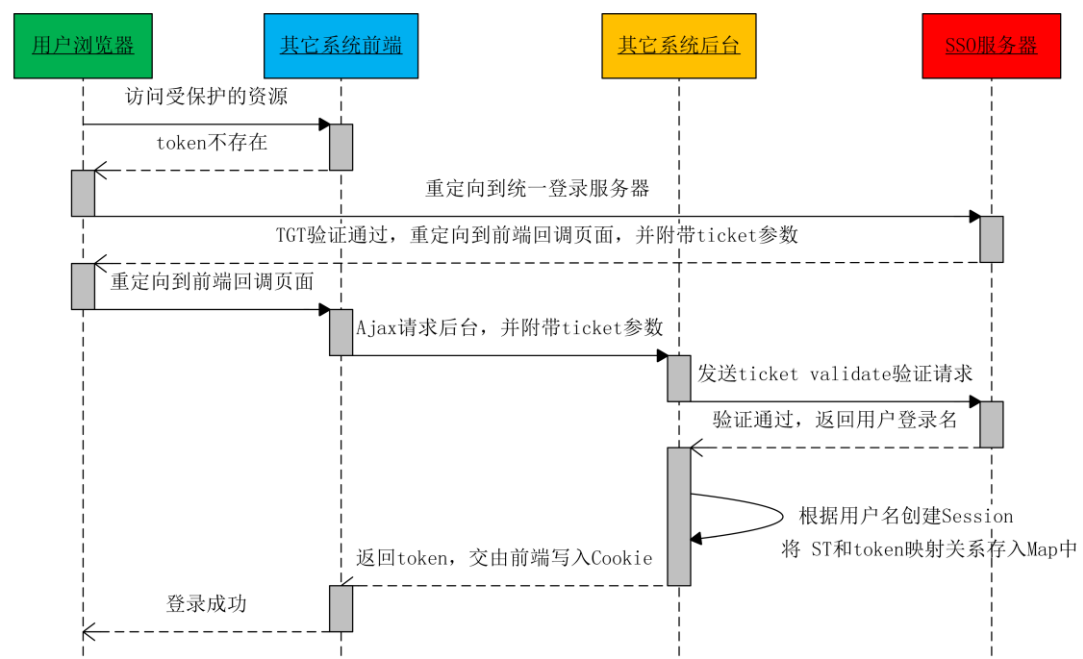
3.1.i 投第一次访问



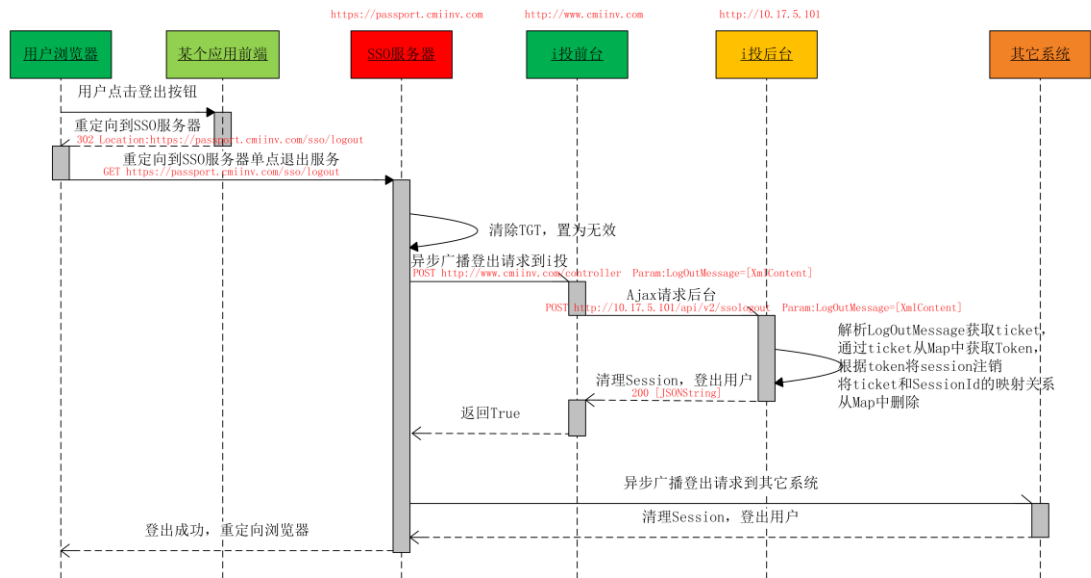
3.2. i 投第二次访问



3.3. 其它系统第一次访问



3.4. 单点退出



4. SSO 单点登录集成接口

4.1. SSO 认证 Service

1. SSO 用户名(手机号、邮箱、证件号码)和密码认证服务接口:

功能: SSO 用户名(手机号、邮箱、证件号码)和密码认证

说明: 用户锁定机制:

- 1). 用户输错密码达到 5 次将锁定, 第 6 次输入此用户登录 (无论密码是非正确), 将提示此用户已被锁定相关信息;
- 2). 用户的锁定时间为 30 分钟, 锁定时间过后将自动解锁 (用户恢复登录), 错误计数将自动从新开始计数;
- 3). 错误计数以天为单位累计, 第二天 (0: 00) 自动重新开始计数;
- 4). 5 次密码输入过程中, 任何一次输入密码正确, 错误计数也将重新开始计数;
- 5). 用户可通过重置密码进行人工解锁, 解锁后用户恢复登录, 错误计数将重新开始;

HTTP 请求方式: POST

请求 URL: http://10.17.5.86/cas/v1/tickets

请求参数:

参数代码	参数内容	是否必须	备注
identifyType	登录类型	否（建议必传）	username: 使用用户名登录 mobile: 使用手机号登录 email: 使用邮箱登录 01: 身份证; 02: 军官证; 03: 护照; 04: 回乡证; 05: 台胞证; 06: 警官证; 07: 士兵证; 99:其他证件;
username	用户名(手机号、邮箱、证件号码)	是	
password	密码(明文)	是	

返回结果:

成功返回格式为 HTML，格式见示例

参数	参数内容	是否必须	备注
response	String	是	包含 TGT

失败返回格式为 JSON，格式见示例

参数	参数内容	是否必须	备注
code	String	是	错误代码

failedTryTimes	Integer	是	登录失败次数
message	String	是	错误信息

例如：

Response 正确结果：

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"><html><head><title>201
The request has been fulfilled and resulted in a new resource being
created</title></head><body><h1>TGT Created</h1><form
action="http://casserver/cas/v1/tickets/TGT-1-db0B4YsJctLCJFYyzUueITSd
X0ycWf1VDF5dM6rn5MCxpwkKuj-cas01.example.org"
method="POST">Service:<input type="text" name="service"
value=""><br><input type="submit" value="Submit"></form></body></html>
```

Response 错误如下：

```
{"Resp": { "code": "0018",
"failedTryTimes":1,message="error.authentication.credentials.bad" }}
或者：
{"Resp": { "code": "0024", "failedTryTimes":5,message="user locked" }}
```

2. SSO 令牌生成服务接口：

功能：SSO 令牌生成服务接口

HTTP 请求方式：POST

请求 URL： http://10.17.5.86/cas/v1/tickets/{0}

请求参数：

参数代码	参数内容	是否必须	备注
{0}	TGT	是	
service	请求域 URL	是	

返回结果：

返回格式为 **HTML**，格式见示例

参数	参数内容	是否必须	备注
response	String	是	ST

例如：

Request 请求：

http://casserver/cas/v1/tickets/TGT-1-db0B4YsJctLCJFYyZUueITSdX0ycWf1V
DF5dM6rn5MCxpwkKuj-cas01.example.org

POST 参数：

service=http%3A%2F%2Fwww.cmiinv.com%2F index.html

Response 正确结果：

ST-1-xUVicQx3eaLyuFd0HqTB-cas01.example.org

3. SSO ST 认证服务接口：

功能：SSO ST 认证

HTTP 请求方式：GET

请求URL： http://10.17.5.86/cas/proxyValidate?ticket
ticket=ticket 加密字符串 &service=encodeURIComponent(i 投回调页面 url)

请求参数：

参数代码	参数内容	是否必须	备注
ticket	ticket 加密字符串	是	
service	请求域 URL	是	url 必须是经过 encodeURIComponent 处理过的，比如冒号、反斜杠都是经过转义之后的值，

返回结果：

返回格式为 XML，格式见示例

参数	参数内容	是否必须	备注
user	String	是	返回登录用户名

例如:

Request 请求:

http://10.17.5.86/cas/proxyValidate?ticket=ST-1-xUVicQx3eaLyuFd0HqTB-cas01.example.org&service=http%3A%2F%2Fwww.cmiinv.com%2F index.html

Response 正确结果:

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
  <cas:authenticationSuccess>
    <cas:user>shixiangwei</cas:user>
    <cas:attributes>
      <cas:uid>A053540F-374C-4334-84A2-D07B54FFE44D</cas:uid>
    </cas:attributes>
  </cas:authenticationSuccess>
</cas:serviceResponse>
```

Response 错误结果:

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
  <cas:authenticationFailure code='INVALID_TICKET'>
    未能够识别出目标 &#039;ST-16&#039;;票根
  </cas:authenticationFailure>
</cas:serviceResponse>
```

4. SSO TGT 延时服务接口:

功能: TGT 默认有效时间为 10 分钟, 如需延长失效时间, 调用本接口

HTTP 请求方式: GET

请求 URL: http://10.17.5.88:8080/cas/v1/ticketsKeepAlive

请求参数:

参数代码	参数内容	是否必须	备注
timeout	失效时长	是	Integer 类型, 单位为秒, 默认 10 分

			钟
--	--	--	---

返回结果:

返回格式为 JSON，格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000，返回信息 Success， 如果是失败，返回错误信息

例如:

Request 请求:

<http://10.17.5.88:8080/cas/v1/ticketsKeepAlive>

或者

<http://10.17.5.88:8080/cas/v1/ticketsKeepAlive?timeout=600>

Cookie:

TGT=TGT-174-czqg7kLMyZ2JN15VlUKN3U0cYT4DhqiRdDyuP61iQ50KbwSrI9-sso.cmi

inv.com; ST=ST-208-pfaak3h1wB3cfV6SihEn-sso.cmiinv.com;

Response 正确结果:

```
{
  "Resp": {
    "code": "0000",
    "message": "Success"
  }
}
```

4.2. SSO 重定向 URL

1. SSO 登录重定向 URL:

功能：SSO 登录前验证 TGT 是否存在或有效，有效则返回 `successUrl`，无效或不存在则返回 `failUrl`

请求 URL：`http://10.17.5.86/cas/cmlogin?failUrl={0} &successUrl={1};`

请求参数：

参数代码	参数内容	是否必须	备注
<code>failUrl</code>	TGT 无效或不存在则返回 <code>failUrl</code>	是	
<code>successUrl</code>	TGT 有效则返回 <code>successUrl</code>	是	

2. SSO 登录成功写入 TGT：

功能：SSO 登录成功后，将 TGT 写入到 SSO 域 Cookie 中

请求 URL：`http://10.17.5.86/cas/SetCookie.jsp?TGT={0}&bakUrl={1}`

请求参数：

参数代码	参数内容	是否必须	备注
TGT	待写入 Cookie 的 TGT 参数	是	
<code>bakUrl</code>	写入后返回 <code>bakUrl</code>	是	

3.SSO 退出重定向 URL：

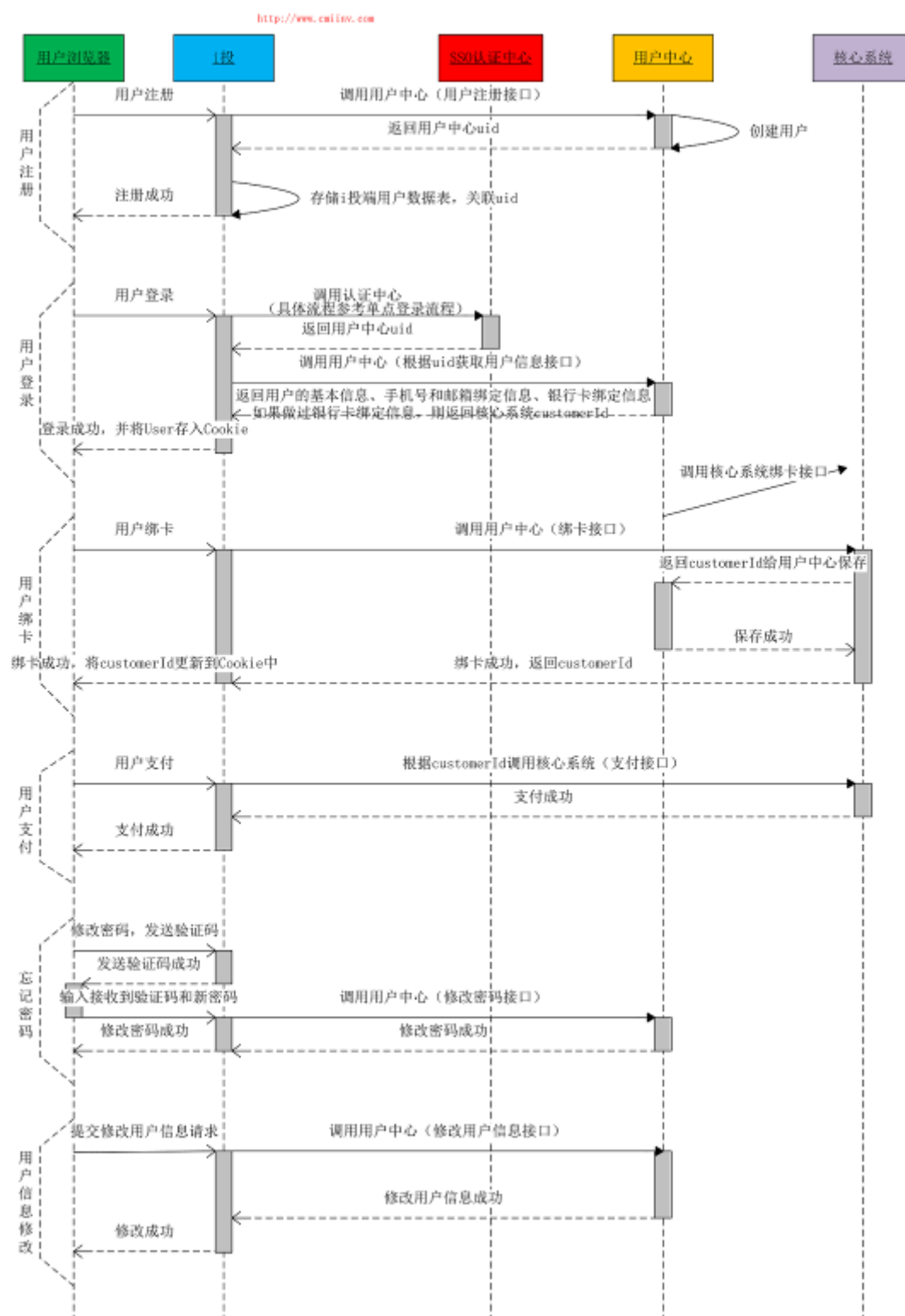
功能：SSO 退出时，将 TGT 从 SSO 域 Cookie 中删除

请求 URL：`http://10.17.5.86/cas/logout?service={0};`

请求参数：

参数代码	参数内容	是否必须	备注
<code>service</code>	删除后返回 url	是	

5. 用户中心对接流程图



6. 用户中心集成接口

6.1. 用户注册接口

1. 用户注册接口：

功能：i 投注册前调用该接口，将用户存入用户中心

HTTP 请求方式：POST

请求 URL：http://10.17.5.62:8880/user/pw/regist

请求参数：

参数代码	参数内容	参数类型	是否必须	备注
loginName	用户登录名	varchar(50)	是	用户名不能为纯数字、邮箱
password	密码(明文)	varchar	是	
mobile	手机号	varchar(11)	否	考虑到个别系统情况，注意：非必输
userRegDevice	注册使用的设备类型	varchar	否	WEB WEB 网页 IOS IOS ANDROID ANDROID WAP WAP MOBILE 移动端 OTHER 其他
salesNo	财富经理员工号	varchar	否	
referrerType	推荐人代码类型	varchar	否	username: 使用用户名登录 mobile: 使用手机号登录

				email: 使用邮箱登录 01: 身份证; 02: 军官证; 03: 护照; 04: 回乡证; 05: 台胞证; 06: 警官证; 07: 士兵证; 99:其他证件;
referrerCode	推荐人代 码	varchar	否	

返回结果:

返回格式为 JSON, 格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000, 返回信息 Success, 如果是失败, 返回错误信息
uid	用户 id	String	是	32 位字符串

例如:

Request 请求:

```
{
  "UserRegistPwReq":{
    "loginName": "shixiangwei",
    "password": "123456",
    "mobile": "18612345678",
```

```
"salesNo": "9000000160",  
"referrerType": "mobile",  
"referrerCode": "18612345678",
```

```
}  
}
```

Response 正确结果:

```
{  
  "Resp": {  
    "code": "0000",  
    "message": "Success",  
    "uid": "a84fb6611a104edbb1b44c480ae708e9"  
  }  
}
```

6.2. 修改密码接口

6.2.1. 根据用户 Id 修改密码接口:

功能: 根据用户 Id 修改密码

HTTP 请求方式: POST

请求 URL: http://10.17.5.62:8880/user/pw/reset

请求参数:

参数代码	参数内容	是否必须	备注
uid	用户 Id	是	
password	新密码(明文)	是	

返回结果:

返回格式为 JSON, 格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000，返回信息 Success， 如果是失败，返回错误信息

例如：

Request 请求：

```
{
  "UserResetPwReq":{
    "uid": "a84fb6611a104edbb1b44c480ae708e9",
    "password": "123456"
  }
}
```

Response 正确结果：

```
{
  "Resp": {
    "code": "0000",
    "message": "Success"
  }
}
```

6.2.2. 根据绑定手机号修改密码接口：

功能：根据绑定手机号修改密码

HTTP 请求方式：POST

请求 URL： http://10.17.5.62:8880/user/pw/reset/mobile

请求参数：

参数代码	参数内容	是否必须	备注
mobileNo	手机号	是	
password	新密码(明文)	是	

返回结果:

返回格式为 JSON，格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000，返回信息 Success， 如果是失败，返回错误信息

例如:

Request 请求:

```
{
  "MobileResetPwReq":{
    "mobileNo": "18611324445",
    "password": "123456"
  }
}
```

Response 正确结果:

```
{
  "Resp": {
    "code": "0000",
    "message": "Success"
  }
}
```

6.3. 获取用户资料接口

1. 获取用户资料接口:

功能: i 投获取用户中心最新用户资料信息

HTTP 请求方式: POST

请求 URL: http://10.17.5.62:8880/user/info/getUser/iTou

请求参数:

参数代码	参数内容	是否必须	备注
uid	用户 id	用户 Id、用户名、手机、邮箱、证件任选一种即可	
loginName	用户名	同上	
mobileNo	手机号	同上	
email	邮箱	同上	
identityType	证件类型	同上	
identityNo	证件号码	同上	

返回结果:

返回格式为 JSON, 格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000，返回信息 Success， 如果是失败，返回错误信息
userBasic	用户基本信息	UserBasic	是	
userSecurityBind	用户安全绑定 信息	UserSecurityBind	否	
userSales	用户绑定财富 经理信息	UserSales	否	
userReferrer	注册推荐人信 息	UserReferrer	否	

userBasic 对象:

字段名称	字段描述	数据类型	备注
userId	用户 ID	VARCHAR	
loginCode	用户登录名	VARCHAR	
userName	用户姓名	VARCHAR	
nickName	用户昵称	VARCHAR	
userType	用户类型	VARCHAR	
userRegSource	用户注册来源	VARCHAR	
createTime	用户注册时间	VARCHAR	
custId	核心客户 ID	VARCHAR	

userSecurityBind 对象:

字段名称	字段描述	数据类型	备注
mobileNo	绑定手机号	VARCHAR	
mobileNoUpdateTime	手机号绑定时间	VARCHAR	

email	绑定 Email	VARCHAR	
emailUpdateTime	Email 绑定时间	VARCHAR	
idCardType	绑定证件类型	VARCHAR	
idCardNo	绑定证件号码	VARCHAR	
idCardNoUpdateTime	证件绑定时间	VARCHAR	

userSales 对象:

字段名称	字段描述	数据类型	备注
salesNo	财富经理员工号	VARCHAR	注：财富经理姓名、手机号码等详细需要调用核心 SALES0004 接口
createTime	绑定时间	VARCHAR	

userReferrer 对象:

字段名称	字段描述	数据类型	备注
referrerId	推荐人 uid	VARCHAR	
createTime	绑定时间	VARCHAR	

例如:

Request 请求:

```
{
  "UserInfoReq":{
    "uid": "4e3cfd612f5e43529e57bf38e87da12e"
  }
}
```

或者

```
{
  "UserInfoReq":{
    "mobileNo": "18612345678"
  }
}
```



```
}  
}
```

Response 正确结果:

```
{  
  "Resp": {  
    "code": "0000",  
    "message": "Success",  
    "userBasic": {  
      "nickName": "王五",  
      "userType": "0",  
      "userRegSource": "0000"  
    },  
    "userSecurityBind": {  
      "mobileNo": "18612345678",  
      "mobileNoUpdateTime": "2016-03-16 12:51:43",  
      "email": "shixiangwei@mail.com",  
      "emailUpdateTime": "2016-03-16 12:52:30"  
    },  
    "userSales": {  
      "salesNo": "9000000160",  
      "createTime": "2016-03-16 12:51:43"  
    },  
    "userReferrer": {  
      "referrerId": "573cfd612f5e73529e57bf38e12ef680",  
      "createTime": "2016-03-16 12:51:43"  
    },  
  }  
}
```

6.4. 用户安全绑定唯一性校验接口

6.4.1. 用户登录名唯一性校验接口：

功能：用户登录名唯一性校验

HTTP 请求方式：POST

请求 URL：http://10.17.5.62:8880/check/loginName

请求参数：

参数代码	参数内容	是否必须	备注
type	key 类型	可为空	
key	用户登录名	是	

返回结果：

返回格式为 JSON，格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000，返回信息 Success， 如果是失败，返回错误信息

例如：

Request 请求：

```
{
  "CheckReq":{
    "key": "shixiangwei"
  }
}
```

Response 正确结果：

```
{
```

```

"Resp": {
  "code": "0000",
  "message": "Success"
}
}

```

6.4.2. 绑定手机号唯一性校验接口：

功能：绑定手机号唯一性校验

HTTP 请求方式：POST

请求 URL：http://10.17.5.62:8880/check/mobile

请求参数：

参数代码	参数内容	是否必须	备注
type	key 类型	可为空	
key	手机号码	是	

返回结果：

返回格式为 JSON，格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000，返回信息 Success， 如果是失败，返回错误信息

例如：

Request 请求：

```

{
  "CheckReq":{
    "key": "18612345678"
  }
}

```

```
}
```

Response 正确结果:

```
{
  "Resp": {
    "code": "0000",
    "message": "Success"
  }
}
```

6.4.3. 绑定邮箱唯一性校验接口:

功能: 绑定邮箱唯一性校验

HTTP 请求方式: POST

请求 URL: http://10.17.5.62:8880/check/email

请求参数:

参数代码	参数内容	是否必须	备注
type	key 类型	可为空	
key	邮箱	是	

返回结果:

返回格式为 JSON, 格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000, 返回信息 Success, 如果是失败, 返回错误信息

例如:

Request 请求:

```
{
```

```
"CheckReq":{  
    "key": "shixiangwei@gmail.com"  
}  
}
```

Response 正确结果:

```
{  
    "Resp": {  
        "code": "0000",  
        "message": "Success"  
    }  
}
```

6.4.4. 绑定证件号码唯一性校验接口:

功能: 绑定证件号码唯一性校验

HTTP 请求方式: POST

请求 URL: http://10.17.5.62:8880/check/identityNo

请求参数:

参数代码	参数内容	是否必须	备注
type	证件类型	是	
key	证件号码	是	

返回结果:

返回格式为 **JSON**, 格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000，返回信息 Success， 如果是失败，返回错误信息

例如：

Request 请求：

```
{
  "CheckReq": {
    "type": "1",
    "key": "110001197901010016"
  }
}
```

Response 正确结果：

```
{
  "Resp": {
    "code": "0000",
    "message": "Success"
  }
}
```

6.5. 用户手机号绑定接口

1. 用户手机号绑定接口：

功能：用户手机号绑定接口

HTTP 请求方式: POST

请求 URL: http://10.17.5.62:8880/certify/mobile

请求参数:

参数代码	参数内容	是否必须	备注
uid	用户 id	是	
mobile	手机号	是	

返回结果:

返回格式为 JSON, 格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000, 返回信息 Success, 如果是失败, 返回错误信息

例如:

Request 请求:

```
{
  "CertifyMobileReq":{
    "uid": "4e3cfd612f5e43529e57bf38e87da12e",
    "mobile": "18612345678"
  }
}
```

Response 正确结果:

```
{
  "Resp": {
    "code": "0000",
    "message": "Success"
  }
}
```

}

6.6. 用户邮箱绑定接口

1. 用户邮箱绑定接口:

功能: 用户邮箱绑定接口

HTTP 请求方式: POST

请求 URL: http://10.17.5.62:8880/certify/email

请求参数:

参数代码	参数内容	是否必须	备注
uid	用户 id	是	
email	email 地址	是	

返回结果:

返回格式为 JSON, 格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000, 返回信息 Success, 如果是失败, 返回错误信息

例如:

Request 请求:

```
{
  "CertifyEmailReq":{
    "uid": "4e3cfd612f5e43529e57bf38e87da12e",
    "email": "shixiangwei@gmail.com"
  }
}
```

Response 正确结果:


```

{
  "Resp": {
    "code": "0000",
    "message": "Success"
  }
}

```

6.7. 用户财富经理(客户经理)绑定接口

1. 用户财富经理绑定接口：

功能：用户财富经理绑定接口

HTTP 请求方式：POST

请求 URL：http://10.17.5.62:8880/user/sales/modify

请求参数：

参数代码	参数内容	是否必须	备注
uid	用户 id	是	
salesNo	财富经理员工号	否	当该值为空，为取消绑定财富经理 当该值为非空，为绑定或修改财富经理

返回结果：

返回格式为 JSON，格式见示例

参数	参数描述	参数类型	是否必须	备注
code	返回标识	String	是	处理成功--0000 请求不合法--0001 数据未找到-- 0002 系统内部错误--9999
message	状态信息	String	是	如果是 0000，返回信息 Success， 如果是失败，返回错误信息

例如：

Request 请求：

```

{
    "UserSalesReq":{
        "uid": "4e3cfd612f5e43529e57bf38e87da12e",
        "salesNo": "9000000160"
    }
}

```

Response 正确结果:

```

{
    "Resp": {
        "code": "0000",
        "message": "Success"
    }
}

```

7. 错误代码表

错误代码	英文标识	中文标识
0001	BadRequest	请求不合法
0002	ParamError	参数错误
0012	MobileNoError	手机号码不正确
0013	MobileDuplicate	手机号码重复
0014	LoginNameDuplicate	登录名重复
0015	IdentityNoDuplicate	证件号重复
0016	EmailDuplicate	邮箱重复
0017	UserNotFound	用户没找到
0018	UserPasswordWrong	用户密码不正确
0019	LoginNameError	登录名错误
0020	IdentityNoError	证件号错误
0021	EmailError	邮箱错误
0022	CredentialsBad	TGT 无效错误
0023	SendSmsError	短信发送失败
0024	UserLockedError	用户被锁定
9999	SystemError	系统内部错误
1000	Other	其它错误