

# ARC 平台应用 App 接口文档

## 更新记录

Author: 覃建策

Email: qinjiance@pwr.com

时间	更新说明
2015/7/17	初始版本
2015/12/15	增加 app 用户接口 （客户端接口 4 - 8）
2015/12/30	增加服务端登陆接口 12
2016/8/17	增加服务端接口【SSO ticket 登录】、【获取 173 用户 ID】
2016/12/8	增加【用户名密码或神盾登录（无需验证密保）】接口
2016/12/22	修改【用户名密码或神盾登录】【用户名密码或神盾登录（无需验证密保）】接口
2017/3/2	增加【获取二维码登录授权码】【二维码登录】接口
2017/3/21	AppUser 对象增加 accountProtected 字段

目录

目录

ARC 平台应用 App 接口文档..... 1

    更新记录..... 2

    目录 ..... 3

    完美用户登录 App 流程 ..... 4

        1. 时序图（客户端登录接口） ..... 4

    客户端接口列表（使用 clientKey） ..... 5

        2. 用户名密码或神盾登录..... 5

        3. 验证密保卡..... 7

        4. 验证手机令牌..... 8

        5. 验证用户名是否被使用..... 8

        6. 用户名密码或神盾登录（无需验证密保） ..... 10

        7. 获取注册验证码..... 11

        8. 获取重置密码 验证码..... 11

        9. App 注册 ..... 12

        10. App 密码重置 ..... 13

        11. 获取登录密码加密公钥..... 15

        12. 获取二维码登录授权码..... 16

        13. 二维码登录..... 17

    服务端接口列表（使用 serverKey） ..... 18

        14. 用户登出..... 18

        15. 获取应用 app token..... 19

        16. 获取用户基本信息..... 20

        17. 用户名密码或神盾登录（无需验证密保） ..... 21

        18. SSO ticket 登录，无需二次验证密保，直接获取 token ..... 22

        19. 获取 173 用户 ID ..... 23

    附录: ..... 24

        1) 接口说明..... 24

        2) 签名，参数说明..... 24

        3) 常见 code ..... 25

        4) AppToken 对象..... 26

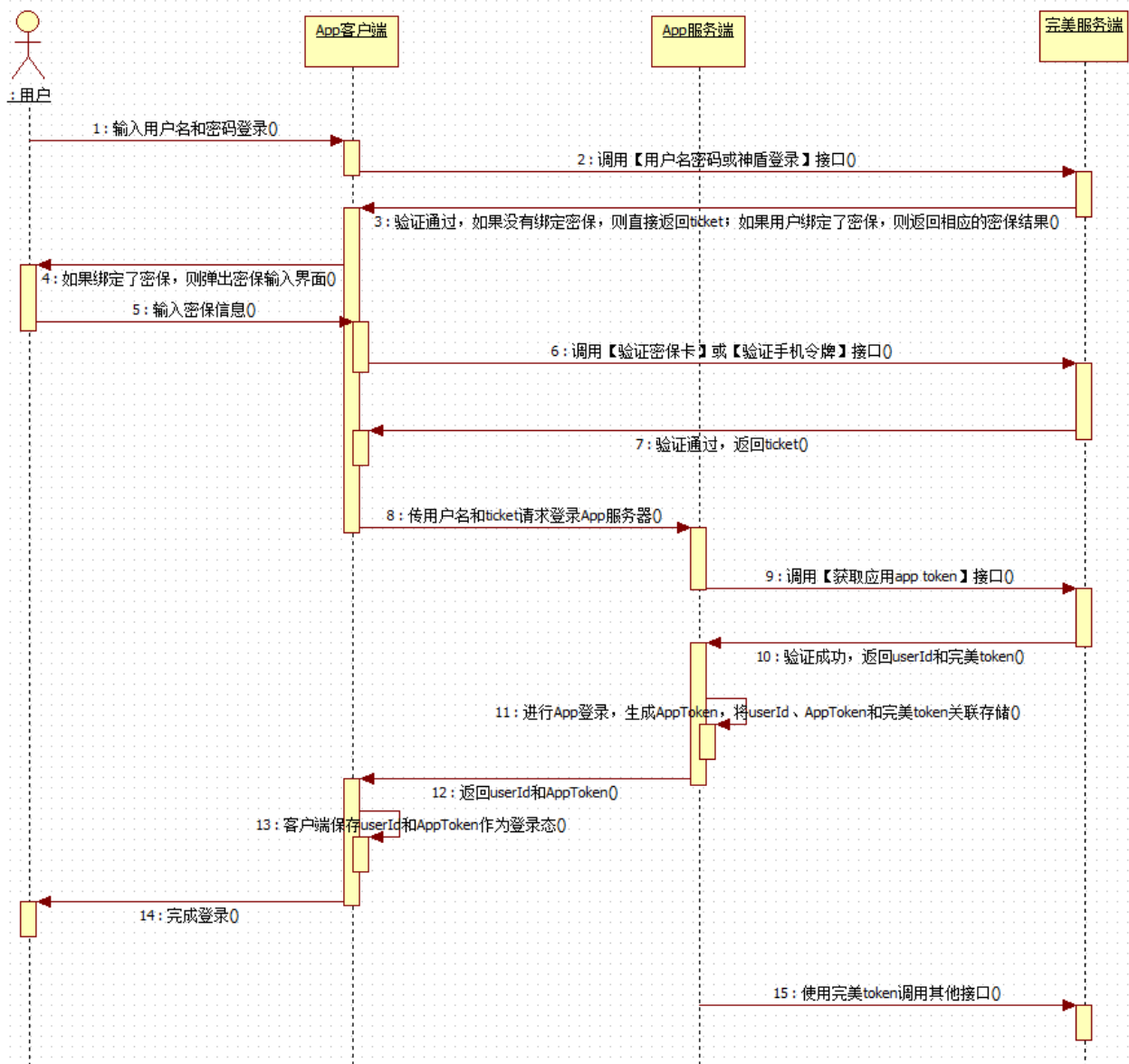
        5) AppUser 对象 ..... 27

        6) Coordinates 对象 ..... 28

        7) QrcodeLoginAuthCode 对象..... 29

完美用户登录 App 流程

1. 时序图（客户端登录接口）



详细的接口参数见接口列表

## 客户端接口列表（使用 clientKey）

### 2. 用户名密码或神盾登录

#### 说明：

对于移动 APP，神盾用户无法登陆

#### 请求地址：

https://openapi.wanmei.com/appApi/appPasswdLogin

#### 请求方式：

post

#### 请求参数：

参数名	必选	类型及范围	说明	参与签名
appId	是	Long	appId	是
username	是	String	用户名	是
password	是	string	1, 普通密码加密：使用【获取登录密码加密公钥】接口返回的公钥进行 RSA 加密后的密文，公钥和加密结果都使用 base64 编码。 2, 神盾密码加密：使用原算法得到 macKey 和 macVal 后，使用公钥对拼接的字符串 macKey+" "+macVal 进行加密	是
timestamp	是	Long	unix 时间戳，毫秒	是
sign	是	String	签名	否

#### 返回数据：

参数名	类型及范围	说明
code	Int	0: 成功，无密保，result 中取得 ticket 1: 需要验证密保卡，result 中取得密保坐标 <a href="#">Coordinates 对象</a> 数组，严格按照数组顺序来提交用户填写的值，调用验证密保卡接口取得 ticket 2: 需要验证手机令牌，调用验证手机令牌接口取得 ticket  <a href="#">错误码：</a> 10001 20001 20002 20005 30014 90001
message	String	消息提示

result	String	ticket
--------	--------	--------

### 3. 验证密保卡

请求地址:

https://openapi.wanmei.com/appApi/appMatrixLogin

请求方式:

post

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	string	appId	是
username	是	string	用户名（邮箱或手机号）	是
code1	是	string	第 1 个密保值（顺序为获取密保坐标的顺序）	是
code2	是	string	第 2 个密保值	是
code3	是	string	第 3 个密保值	是
timestamp	是	long	unix 时间戳，毫秒	是
sign	是	string	签名	否

返回数据:

参数名	类型及范围	说明
code	int	0: 成功  <a href="#">错误码:</a> 10001 20001 20003 30001 90001
message	string	消息提示
result	String	ticket

#### 4. 验证手机令牌

请求地址:

https://openapi.wanmei.com/appApi/appPhoneLogin

请求方式:

post

请求参数:

参数名	必选	类型及范围	说明	参与签名
appld	是	string	appld	是
username	是	string	用户名（邮箱或手机号）	是
phoneToken	是	string	手机令牌值	是
timestamp	是	long	unix 时间戳，毫秒	是
sign	是	string	签名	否

返回数据:

参数名	类型及范围	说明
code	int	0: 成功  <a href="#">错误码:</a> 10001 20001 20004 30001 90001
message	string	消息提示
result	string	ticket

#### 5. 验证用户名是否被使用

请求地址:

https://openapi.wanmei.com/appApi/existUsername

请求方式:

post/get

请求参数:

参数名	必选	类型及范围	说明	参与签名
appld	是	string	appld	是
username	是	string	用户名（邮箱或手机号）	是



timestamp	是	long	unix 时间戳，毫秒	是
sign	是	string	签名	否

返回数据：

参数名	类型及范围	说明
code	int	0: 没有被使用 1: 已经被使用  <a href="#">错误码：</a> 10001 20001 30014 90001
message	string	消息提示
result	string	

## 6. 用户名密码或神盾登录（无需验证密保）

说明：

对于移动 APP，神盾用户无法登陆

请求地址：

https://openapi.wanmei.com/appApi/appPasswdLoginNoSecure

请求方式：

post

请求参数：

参数名	必选	类型及范围	说明	参与签名
appId	是	Long	appId	是
username	是	String	用户名	是
password	是	string	1, 普通密码加密：使用【获取登录密码加密公钥】接口返回的公钥进行 RSA 加密后的密文，公钥和加密结果都使用 base64 编码。 2, 神盾密码加密：使用原算法得到 macKey 和 macVal 后，使用公钥对拼接的字符串 macKey+" " + macVal 进行加密	是
timestamp	是	Long	unix 时间戳，毫秒	是
sign	是	String	签名	否

返回数据：

参数名	类型及范围	说明
code	Int	0: 成功，无密保，result 中取得 ticket 1: 已绑定密保卡，result 中取得 ticket 2: 已绑定手机令牌，result 中取得 ticket  <a href="#">错误码：</a> 10001 20001 20002 20005 30014 90001
message	String	消息提示
result	String	ticket

## 7. 获取注册验证码

请求地址:

https://openapi.wanmei.com/appApi/sendRegCaptcha

请求方式:

post/get

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	string	appId	是
username	是	string	用户名（邮箱或手机号）	是
timestamp	是	long	unix 时间戳，毫秒	是
sign	是	string	签名	否

返回数据:

参数名	类型及范围	说明
code	int	0: 发送失败 1: 发送成功  <a href="#">错误码:</a> 10001 20001 30014 90001
message	string	消息提示
result	string	

## 8. 获取重置密码 验证码

请求地址:

https://openapi.wanmei.com/appApi/sendCMPCaptcha

请求方式:

post/get

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	string	appId	是
username	是	string	用户名（邮箱或手机号）	是
timestamp	是	long	unix 时间戳，毫秒	是

sign	是	string	签名	否
------	---	--------	----	---

返回数据：

参数名	类型及范围	说明
code	int	0: 发送失败 1: 发送成功  <a href="#">错误码：</a> 10001 20001 30014 90001
message	string	消息提示
result	string	

## 9. App 注册

请求地址：

https://openapi.wanmei.com/appApi/register

请求方式：

post

请求参数：

参数名	必选	类型及范围	说明	参与签名
appId	是	string	appId	是
username	是	string	用户名（邮箱或手机号）	是
password	是	string	密码	是
captcha	是	string	验证码	是
timestamp	是	long	unix 时间戳，毫秒	是
sign	是	string	签名	否

返回数据：

参数名	类型及范围	说明
-----	-------	----

code	int	0: 注册成功  <u>错误码:</u> 10001 参数验证错误 错误信息在 result 20001 30014 90001
message	string	消息提示
result	string	具体错误信息 是键值对 比如: password: 6 位数字或字母 captcha: 验证码错误

## 10. App 密码重置

请求地址:

<https://openapi.wanmei.com/appApi/resetPwd>

请求方式:

post

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	string	appId	是
username	是	string	用户名（邮箱或手机号）	是
password	是	string	密码	是
captcha	是	string	验证码	是
timestamp	是	long	unix 时间戳，毫秒	是
sign	是	string	签名	否

返回数据:

参数名	类型及范围	说明
code	int	0: 密码修改成功

		<a href="#">错误码:</a> 10001 参数验证错误 错误信息在 result 20001 30014 90001
message	string	消息提示
result	string	具体错误信息 是键值对 比如: password: 6 位数字或字母 captcha: 验证码错误

11. 获取登录密码加密公钥

说明

公钥几乎不会变化，可以每次打开客户端只取一次

请求地址：

https://openapi.wanmei.com/getLoginPasswdEncPubKey

请求方式：

post

get

请求参数：

参数名	必选	类型及范围	说明	参与签名
appId	是	string	appId	是
timestamp	是	long	unix 时间戳，毫秒	是
sign	是	string	签名	否

返回数据：

参数名	类型及范围	说明
code	int	0: 成功  <a href="#">错误码：</a> 10001 20001 30001 90001
message	string	消息提示
result	string	RSA 公钥字符串

## 12. 获取二维码登录授权码

说明:

- 1, 用户打开二维码登陆界面
- 2, 调用【获取二维码登录授权码】接口
- 3, 每隔 3-5 秒钟调用一次【二维码登录】接口, 根据返回值做不同的处理:
  - a 返回 0: 登陆成功, 取得 result 里的 ticket, 不再调用【二维码登录】接口。
  - b 返回 20008: 登陆二维码已失效, 显示一个提示: 主动点击图片来刷新二维码。用户点击后, 回到步骤 2
  - c 返回其他错误: 继续步骤 3

请求地址:

<https://openapi.wanmei.com/appApi/qrcodeLoginAuthCode>

请求方式:

post

get

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	string	appId	是
productId	是	long	登录产品 ID	是
timestamp	是	long	unix 时间戳, 毫秒	是
sign	是	string	签名	否

返回数据:

参数名	类型及范围	说明
code	int	0: 成功  <u>错误码:</u> 10001 20001 30001 90001
message	string	消息提示
result	json	<a href="#">QrcodeLoginAuthCode 对象</a>



### 13. 二维码登录

说明:

- 1, 用户打开二维码登陆界面
- 2, 调用【24.获取二维码登录授权码】接口
- 3, 每隔 3-5 秒钟调用一次【二维码登录】接口, 根据返回值做不同的处理:
  - a 返回 0: 登陆成功, 取得 result 里的 ticket, 不再调用【二维码登录】接口。
  - b 返回 20008: 登陆二维码已失效, 显示一个提示: 主动点击图片来刷新二维码。用户点击后, 回到步骤 2
  - c 返回其他错误: 继续步骤 3

请求地址:

<https://openapi.wanmei.com/appApi/appQrcodeLogin>

请求方式:

post

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	string	appId	是
authCode	是	string	授权码	是
productId	是	long	登录产品 ID	是
authSign	是	string	授权签名	是
timestamp	是	long	unix 时间戳, 毫秒	是
sign	是	string	签名	否

返回数据:

参数名	类型及范围	说明
code	int	0: 成功  <a href="#">错误码:</a> 10001 20001 20008 30001 90001
message	string	消息提示
result	json	ticket

服务端接口列表（使用 serverKey）

14. 用户登出

请求地址：

https://openapi.wanmei.com/appApi/appLogout

请求方式：

post

get

请求参数：

参数名	必选	类型及范围	说明	参与签名
appId	是	Long	appId	是
userId	是	Long	用户 ID	是
token	是	String	app token	是
timestamp	是	Long	unix 时间戳，毫秒	是
sign	是	String	签名	否

返回数据：

参数名	类型及范围	说明
code	Int	0: 成功  <a href="#">错误码：</a> 10001 20001 30015 30016 90001
message	String	消息提示

15. 获取应用 app token

请求地址:

https://openapi.wanmei.com/appApi/getAppToken

请求方式:

post

get

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	Long	appId	是
username	是	String	用户名	是
ticket	是	String	ticket	是
timestamp	是	Long	unix 时间戳, 毫秒	是
sign	是	String	签名	否

返回数据:

参数名	类型及范围	说明
code	Int	0: 成功  <a href="#">错误码:</a> 10001 20001 30006 30015 90001
message	String	消息提示
result	json	<a href="#">AppToken 对象</a>

16. 获取用户基本信息

请求地址:

https://openapi.wanmei.com/appApi/getUserInfo

请求方式:

post

get

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	Long	appId	是
userId	是	Long	用户 ID	是
token	是	String	app token	是
timestamp	是	Long	unix 时间戳, 毫秒	是
sign	是	String	签名	否

返回数据:

参数名	类型及范围	说明
code	Int	0: 成功  <a href="#">错误码:</a> 10001 20001 30015 30016 90001
message	String	消息提示
result	json	<a href="#">AppUser 对象</a>

17. 用户名密码或神盾登录（无需验证密保）

说明：

对于移动 APP，神盾用户无法登陆

请求地址：

https://openapi.wanmei.com/appApi/appPasswdServerLogin

请求方式：

post

请求参数：

参数名	必选	类型及范围	说明	参与签名
appId	是	Long	appId	是
username	是	String	用户名	是
password	是	string	1, 普通密码加密：使用【获取登录密码加密公钥】接口返回的公钥进行 RSA 加密后的密文，公钥和加密结果都使用 base64 编码。 2, 神盾密码加密：使用原算法得到 macKey 和 macVal 后，使用公钥对拼接的字符串 macKey+" "+macVal 进行加密	是
timestamp	是	Long	unix 时间戳，毫秒	是
sign	是	String	签名	否

返回数据：

参数名	类型及范围	说明
code	Int	0: 成功，无密保，result 中取得 appToken 1: 用户绑定了密保卡，result 中取得 appToken 2: 用户绑定了手机令牌，result 中取得 appToken  <a href="#">错误码：</a> 10001 20001 20002 20005 30014 90001
message	String	消息提示
result	Json	<a href="#">AppToken 对象</a>

18. SSO ticket 登录，无需二次验证密保，直接获取 token

说明：

请求地址：

https://openapi.wanmei.com/appApi/appSSOTicketServerLogin

请求方式：

post

请求参数：

参数名	必选	类型及范围	说明	参与签名
appId	是	Long	appId	是
username	是	String	用户名	是
ssoTicket	是	String	sso ticket	是
timestamp	是	Long	unix 时间戳，毫秒	是
sign	是	String	签名	否

返回数据：

参数名	类型及范围	说明
code	Int	0: 成功，无密保，result 中取得 appToken  <a href="#">错误码：</a> 10001 20001 20002 30014 90001
message	String	消息提示
result	Json	<a href="#">AppToken 对象</a>

19. 获取 173 用户 ID

请求地址:

https://openapi.wanmei.com/appApi/get173UserIdByUserId

请求方式:

post

get

请求参数:

参数名	必选	类型及范围	说明	参与签名
appId	是	Long	appId	是
userId	是	Long	用户 ID	是
token	是	String	app token	是
timestamp	是	Long	unix 时间戳, 毫秒	是
sign	是	String	签名	否

返回数据:

参数名	类型及范围	说明
code	Int	0: 成功 1: 173 用户 ID 不存在  <a href="#">错误码:</a> 10001 20001 30015 30016 90001
message	String	消息提示
result	Long	173 用户 ID

## 20.

### 附录:

#### 1) 接口说明

- 1、接口中有关字符串转字节流的所有操作，字符集约定都使用 UTF-8 编码
- 2、接口大类分为客户端接口和服务端接口，使用不同的 key 进行签名

#### 2) 签名，参数说明

##### 1、签名:

- (1) 接口参数列表中如果无参与签名的参数则不需要签名；非必选参数如果出现则必须参与签名
- (2) 所有需要签名的接口中都有时间戳 `timestamp`，单位毫秒，每次请求都需要重新生成当前时间作为时间戳，不可重复使用
- (3) Get 请求时签名 `sign` 需要进行 url encode
- (4) 一个应用 `appId` 对应一对签名密钥，一个客户端 `clientKey`，一个服务端 `serverKey`，签名时根据调用的接口类型来选择不同的 key
- (5) 签名算法：
  - A. 将所有参与签名的参数按照参数名的字母顺序升序排列，如：  
`appId: 1, userId: 2`
  - B. 将排序后的参数拼接成 `key1=value1&key2=value2&.....` 的字符串 `srcStr`，如：  
`appId=1&userId=2`
  - C. 客户端或服务端 key 为 32 位字符串，取它的特定位数作为最终加密密钥（`encKey`），特定位数见例子：例子中的下标从 0 开始，如 `substring(1, 4)` 表示取下标 1、2、3 的字符：  
`key.substring(1, 4) + key.substring(7, 13) + key.substring(15, 17) + key.substring(26, 31)`
  - D. 将 `srcStr` 进行 SHA256 加 salt 哈希，再用 base64 编码成字符串（`sign`），salt 值为 `encKey`，即 `SHA256(encKey+srcStr)`
  - E. `sign` 即为签名



### 3) 常见 code

服务器接口返回的 Code 根据值的范围分为如下几大类：

- 1, < 10000, 处理成功, 根据具体接口中特殊定义
- 2, > 10000, 参数错误
- 3, > 20000, 业务错误
- 4, > 30000, 访问权限错误
- 5, > 90000, 系统错误

错误码 (> 10000) 列表：

code	message	说明
10001	参数错误	通用错误码
20001	业务处理失败	通用错误码
20002	用户名或密码错误	
20003	密保卡验证错误	
20004	手机令牌验证错误	
20005	用户被封禁	
30006	ticket 无效	
30014	应用客户端签名无效	
30015	应用服务端签名无效	
30016	应用 token 无效	
90001	系统内部错误	通用错误码

#### 4) AppToken 对象

参数名	类型	说明
userId	long	用户 userId，唯一标识
token	string	用户客户端 token
expireIn	long	有效期，秒

## 5) AppUser 对象

参数名	类型	说明
userId	long	用户 userId，唯一标识
username	string	用户名
email	string	邮箱
cellphone	string	手机号
nickname	string	昵称
isIndulge	Int	0: 未注册防沉迷信息且已纳入防沉迷 1: 已纳入防沉迷 (<18 岁) 2: 未纳入防沉迷 (>=18 岁)
accountProtected	int	0: 无密保措施 1: 已绑定密保卡 2: 已绑定手机令牌 3: 已绑定神盾
registerTime	long	注册时间 (unix 时间戳，毫秒)

## 6) Coordinates 对象

参数名	类型	说明
x	string	密保卡横坐标
y	string	密保卡纵坐标

## 7) QrcodeLoginAuthCode 对象

参数名	类型	说明
authCode	string	授权码
productId	long	产品 ID
sign	string	授权签名
qrcodeUrl	string	二维码图片 Url