
号外号平台 ACI 接口说明书

2016 年 7 月

目录

1	前言	
2	接口概述	
2.1	通信模型	
2.1.1	业务消息交互	
2.2	消息格式	
2.2.1	消息组成	
2.2.2	消息类型	
2.3	业务交互阶段	
2.3.1	业务交互（ACBSS 接口）	
2.3.1.1	订购请求和响应	
2.3.1.2	退订请求和响应	
2.3.1.3	订购查询请求和响应	
2.3.1.4	真实号码换号请求和响应	
2.3.2	业务交互（ACSA 接口）	
2.3.2.1	在线语音主叫请求和响应	
2.3.2.2	用户定时开关机设置请求和响应	
2.3.2.3	用户开关机查询请求和响应	
2.3.2.4	系统呼叫结束推送（准实时）	
2.3.3	业务交互（AXB 业务 BSS 接口）	
2.3.3.1	订购请求和响应	
2.3.3.2	退订请求和响应	
2.3.3.3	订购查询请求和响应	
2.3.3.4	订购更新请求和响应	
2.3.3.5	外呼 AXB 转接配置设置请求和响应	错误! 未定义书签
2.3.3.6	外呼 AXB 转接配置删除请求和响应	
2.4	返回码定义	
2.5	附:签名算法	
2.5.1	Java	
2.5.2	PHP	

1 前言

本协议描述了和互联网平台之间数据传输的数据格式和规范。

- 约定：

我们把号外号平台和互联网平台之间的接口定义为 **A1 接口**。

号外号平台叫做 **AC 平台**。

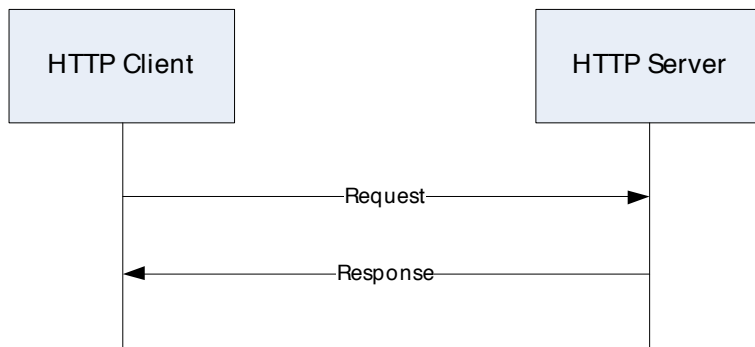
互联网平台我们简称 **IP**

2 接口概述

A1 接口采用 HTTP 协议，发送响应的数据格式采用 JSON 格式。

2.1 通信模型

2.1.1 业务消息交互



根据业务消息的不同，AC 作为服务器端，提供不同的业务地址和端口。

业务类型	地址/端口
互联网 BSS 业务（acbss）	具体待定
互联网 SA 业务（acsa）	具体待定

2.2 消息格式

2.2.1 消息组成

消息由消息头和消息体组成，以 JSON 格式，包含在 HTTP 载荷中。

版本号	消息头
消息ID	
时间戳	
业务类型	
消息类型	
保留字段	
内容	消息体

- 对于消息头的字段，响应方返回的字段与请求方的字段应该保持一致。
- Json 对象的名称，一律以小写表示
- Json 对象的取值为数字的，一律以字符串表示。
- 消息都是以 GBK 编码

消息头包括如下字段：

消息结构	内容	Json 对象名称	说明	举例
消息头	版本号	ver	消息请求方设定的接口协议的版本。 消息响应方原值返回。	“ver”：“1.0”
	消息 ID	msgid	消息请求发送方设定的唯一消息标识。具体由发送方设定。 重发的消息 id 必须保持不变。 消息响应方原值返回。	“msgid”： “4736219853”
	时间戳	ts	以 YYYYMMDDhhmmssNNN 的十进制格式表示，由消息请求发送方在发送消息时设定。其中，NNN 表示 3 位毫秒数。 消息响应方原值返回。	“ts”： “20140402101356320”

	业务类型	service	对于 AC 系统的业务类型定义如下： <table><tr><td>业务类型</td><td>说明</td></tr><tr><td>acbss</td><td>ACBSS 业务</td></tr><tr><td>acsa</td><td>ACSA 业务</td></tr></table> 消息响应方原值返回。	业务类型	说明	acbss	ACBSS 业务	acsa	ACSA 业务	“service”： “acbss”
	业务类型	说明								
	acbss	ACBSS 业务								
	acsa	ACSA 业务								
消息类型	msgtype	消息类型与业务类型相关。见“消息类型与业务类型”。 消息响应方返回对应请求的消息响应类型。	“msgtype”： “subreq”							
签名	sid	对消息头和消息体的数字签名。 参考后面算法	“sid”： “0”							
保留字段	rsvd	扩展保留字段。 消息响应方原值返回。	“rsvd”： “0”							
	应用标识	appkey	可选。第三方应用的 AppKey。 AXB 业务时必须设置。	“appkey”： “kuaidadi”						
消息体	业务消息内容。		内容与业务消息类型相关。见后续业务交互中的消息定义。	采用扁平结构方式，取消嵌套结构。						

2.2.2 消息类型

序号	消息类型	消息说明	产品类别	交互实体
1	Subreq	订购请求	AX	BSS
2	Subrsp	订购响应	AX	
3	Unsubreq	退订请求	AX	
4	Unsubrsp	退订响应	AX	
5	Subqryreq	订购查询	AX	
6	Subqryrsp	订购响应	AX	

7	Chgprtmsreq	真实号码换号请求	AX	
8	Chgprtmsrsp	真实号码换号响应	AX	
9	Axbsubreq	订购请求	AXB	
10	Axbsubrsp	订购响应	AXB	
11	Axbunsubreq	退订请求	AXB	
12	Axbunsubrsp	退订响应	AXB	
13	axbsubqryreq	订购查询	AXB	
14	axbsubqryrsp	订购响应	AXB	
15	axbsubupdreq	AXB 订购更新请求	AXB	
16	axbsubupdrsp	AXB 订购更新请求	AXB	
17	vcallreq	在线语音主叫请求	AX	SA
18	vcallrsp	在线语音主叫响应	AX	
19	smsreq	在线短信主叫请求	AX	
20	smsrsp	在线短信主叫响应	AX	
21	switchtimereq	用户定时开关机设置请求	AX	
22	switchtimersp	用户定时开关机设置响应	AX	
23	switchqryreq	用户开关机查询请求	AX	
24	switchqryrsp	用户开关机查询响应	AX	

2.3 业务交互阶段

2.3.1 业务交互（ACBSS 接口）

互联网 BSS 主要处理的互联网业务请求和响应类型如下：

- 订购
- 退订
- 订购关系查询
- 真实号码换号

IP 发送业务消息到 AC A1 接口。AC BSS 进行内部处理，将结果反馈给 IP 系统。

消息交互如下：



对于一个唯一的订购关系，AC 平台采用如下字段来标识：

- 订购关系标识

AC 系统需要确保“订购关系标识”是唯一的。建议可以考虑使用互联网产品订单编号等。

- 隐私号码

为了保持隐私号码的唯一分配关系。AC 系统需要尽量保证隐私号码只属于一个“订购关系”。

2.3.1.1 订购请求和响应

- 订购请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	Ver		
	消息 ID	Msgid		
	时间戳	Ts		
	业务类型	service	Acbss	
	消息类型	msgtype	subreq	
	签名	Sid		
	应用标识	appkey	第三方应用的 AppKey。	
	保留字段	Rsvd		
消息体	被保护的真实号码	Prtns		
	隐私号码	acms	AC 动态分配或者指定	

	订购时间	Subts	格式为 YYYYMMDDhhmmss。时间采用北京时间，24 小时制。	“subts” : “ 20140728145921”
	订购类型	producttype	现阶段支持的类型： 0: 包年 1: 包季 2: 包月 3: 按次 4: 体验	“producttype” : “0”
	姓名	name	必选	“name” : “ 张三”
	证件类型	cardtype	必选	“cardtype” : “ ”
	身份证号码	cardno	必选	“cardno”: “ ”
	录音控制	callrecording	可选。是否开通录音功能。在本字段不出现的情况下，默认 0（不开通录音功能）。 0: 不开通录音功能 1: 开通录音功能	“callrecording” : “0”

● 订购响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	subrsp	
	签名	sid		
	应用标识	appkey	第三方应用的 AppKey。	
	保留字段	rsvd		

消息体	返回码	result	采用类似 http 的状态码风格。 成功返回码： 200：成功 其他： 客户端（IP 系统）错误： 400：请求报文语法错误 401：认证未通过 403：订购关系不存在 404：绑定数据不一致 其他： 服务端（AC）错误： 501：业务处理超时 502：服务暂时不可用 其他：	
	订购关系标识	subid		
	隐私号码	acms		

2.3.1.2 退订请求和响应

● 退订请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		

	业务类型	service	acbss	
	消息类型	msgtype	unsubreq	
	签名	sid		
	应用标识	appkey	第三方应用的 AppKey。	
	保留字段	rsvd		
消息体	被保护的真实号码	prtms		
	隐私号码	acms		
	退订时间	unsubts	格式为 YYYYMMDDhhmmss。时间采用北京时间，24 小时制。	

● 退订响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	unsubrsp	
	签名	sid		
	应用标识	appkey	第三方应用的 AppKey。	
	保留字段	rsvd		
消息体	返回码	result	参见“返回码定义”	
	订购关系标识	subid		

2.3.1.3 订购查询请求和响应

● 订购查询请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	subqryreq	
	签名	sid		
	保留字段	rsvd		
消息体	订购关系标识	subid	方式一	
	被保护的真实号码	prtms	方式二	
	隐私号码	acms		

● 订购查询响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	subqryrsp	
	签名	sid		
	保留字段	rsvd		
消息体	返回码	result	参见“返回码定义”	
	订购关系标识	subid		
	被保护的真实号码	prtms		

	隐私号码	acms		
	订购时间	subts		
	订购产品类型	producttype		
	姓名	name		
	证件类型	cardtype		
	身份证号码	cardno		“cardno”: “ ”

2.3.1.4 真实号码换号请求和响应

● 真实号码换号请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	chgprtsreq	
	签名	sid		
	保留字段	rsvd		
消息体	订购关系标识	subid	与订购请求的 subid 一致，AC 返回的 sub_id	
	隐私号码	acms		
	当前真实号码	oldprts		
	新的真实号码	newprts		
	换号时间	chgts	格式为 YYYYMMDDhhmmss。时间 采用北京时间,24 小时 制。	

- 真实号码换号响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	chgprtmsrsp	
	签名	sid		
	保留字段	rsvd		
消息体	返回码	result	参见“返回码定义”	
	订购关系标识	subid		

2.3.2 业务交互（ACSA 接口）

互联网 SA 主要处理的互联网业务请求和响应类型如下：

- 在线语音主叫

IP 业务系统发起请求，AC 接收后，进行内部处理，响应处理结果，后续启动业务流程。



2.3.2.1 在线语音主叫请求和响应

- 在线语音主叫请求

消息头	内容	JSON 对象名称	说明	举例
-----	----	-----------	----	----

	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acsa	
	消息类型	msgtype	vcallreq	
	签名	sid		
	保留字段	rsvd		
消息体	隐私号码	acms		
	被叫号码	calledms	被叫号码	

● 在线语音主叫响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acsa	
	消息类型	msgtype	vcallrsp	
	签名	sid		
	保留字段	rsvd		
消息体	返回码	result	参见“返回码定义”	

2.3.2.2 用户定时开关机设置请求和响应

● 用户定时开关机设置请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		

	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acsa	
	消息类型	msgtype	switchtimerreq	
	签名	sid		
	保留字段	rsvd		
消息体	隐私号码	acms		
	定时关机时间点	closetime	<p>可选。24 小时格式表示 hhmm</p> <p>如果 closetime 出现，但是值为空，则表示取消相应设置。即 “closetime” : “ ” 表示取消设置。</p>	<p>“closetime” : “14:30”</p>
	定时开机时间点	opentime	<p>可选。24 小时格式表示 hhmm</p> <p>如果 opentime 出现，但是值为空，则表示取消相应设置。即 “opentime” : “ ” 表示取消设置。</p>	<p>“ opentime ” : “17:30”</p>

● 用户定时开关机设置响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		

	时间戳	ts		
	业务类型	service	acsa	
	消息类型	msgtype	switchtimerrsp	
	签名	sid		
	保留字段	rsvd		
消息体	返回码	result	参见“返回码定义”	

2.3.2.3 用户开关机查询请求和响应

● 用户开关机查询请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acsa	
	消息类型	msgtype	switchqryreq	
	签名	sid		
	保留字段	rsvd		
消息体	隐私号码	acms		

● 用户开关机查询响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acsa	
	消息类型	msgtype	switchqryrsp	

	签名	sid		
	保留字段	rsvd		
消息体	返回码	result	参见“返回码定义”	
	当前开关机状态	switch	0: 开机 1: 关机	“switch” : “1”
	定时关机时间点	closetime	24 小时格式表示 hhmm 如果 closetime 出现，但是值为空，则表示无相应设置。即 “closetime” : “ ” 表示无定时关机设置。	“closetime” : “14:30”
	定时开机时间点	opentime	24 小时格式表示 hhmm 如果 opentime 出现，但是值为空，则表示无相应设置。即 “opentime” : “ ” 表示无定时开机设置。	“ opentime ” : “17:30”

2.3.2.4 系统呼叫结束推送（准实时）

- 推送机制

采用 json over http 协议

- 协议

#协议参数

'v': '1.0', #版本号

'method': 'add_call_release', #方法

'app_key': 'ac', #调用方 appkey

'sign': '0BE1A00DD73A0A3D6037657B7C3C9DF3', #签名

'timestamp': '2016-06-22 14:03:21', #时间戳

#业务参数

'phone_no': '13333300000', #手机号码

'secret_no': '8618788006777', #隐私号码

'call_type': '0', #呼叫类型

'peer_no': '13344400000', #对端号码

'call_id': '0d01000000080000000000000000', #通话唯一标识

'call_time': '2016-06-22 14:03:20', #通话开始时间

'release_time': '2016-06-22 14:03:20', #通话结束时间

'release_dir': '1', #释放方向

'release_cause': '16', #释放原因

'ringing_time': '2016-06-22 14:03:20', #
振铃开始时间

- 推送请求

API 名称	method = add_call_release	
接入方标识	app_key	标识接入方



号外号 ACI 接口协议

呼叫类型	call_type	AC 业务： 0：DTMF 方式通话主叫 1：通话被叫 2：短信发送 3：短信接收 128:PS 方式通话主叫
真实号码	phone_no	真实号码为 A
隐私号码	secret_no	隐私号码为 X
对端号码	peer_no	对端号码为 B 或者其它填充 0
振铃开始时间	ringing_time	格式: yyyy-MM-dd HH:mm:ss
通话开始时间	call_time	格式: yyyy-MM-dd HH:mm:ss
通话结束时间	release_time	格式: yyyy-MM-dd HH:mm:ss
通话标识	call_id	业务参考号
释放方向	release_dir	1 表示主叫， 2 表示被叫， 0 表示平台释放

释放原因	release_cause	000 0001 (1) 未分配的号码 000 0010 (2) 无路由到指定的转接网 000 0011 (3) 无路由到目的地 000 0100 (4) 发送专用信息音 001 0000 (16) 正常的呼叫拆线 001 0001 (17) 用户忙 001 0010 (18) 用户未响应 001 0011 (19) 用户未应答 001 0100 (20) 用户缺席 001 0101 (21) 呼叫拒收 001 0110 (22) 号码改变 001 1011 (27) 目的地不可达 001 1100 (28) 无效的号码格式 (地址不全) 001 1101 (29) 性能拒绝 001 1111 (31) 正常—未指定 类别 010 , 资源不可用类 : 010 0010 (34) 无电路/通路可用 010 1010 (42) 交换设备拥塞 类别 011 , 业务或任选不可用类 : 011 0010 (50) 所请求的性能未预定
------	---------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<p>011 0101 (53) CUG 中限制去呼叫</p> <p>011 0111 (55) CUG 中限制来呼叫</p> <p>011 1001 (57) 承载能力无权</p> <p>011 1010 (58) 承载能力目前不可用</p> <p>类别 100 , 业务或任选未实现类 :</p> <p>100 0001 (65) 承载能力未实现</p> <p>100 0101 (69) 所请求的性能未实现</p> <p>类别 101 , 无效的消息 (例如参数超出范围) 类 :</p> <p>101 0111 (87) 被叫用户不是 CUG 的成员</p> <p>101 1000 (88) 不兼容的目的地</p> <p>101 1010 (90) 不存在的 CUG</p> <p>101 1011 (91) 无效的转接网选择</p> <p>101 1111 (95) 无效的消息 , 未指定</p> <p>类别 110 , 协议错误 (例如未知的消息) 类 : 110</p> <p>0001 (97) 消息类型不存在或未实现</p>
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

		<p>110 0011 (99) 参数不存在或未实现</p> <p>110 0110 (102) 定时器终了时恢复 110</p> <p>0101 (103) 参数不存在或未实现—传递</p> <p>110 1110 (110) 消息带有未被识别的参数—舍弃</p> <p>110 1111 (111) 协议错误，未指定</p> <p>类别 111，互通类：</p> <p>111 1111 (127) 互通，未指定类别 1100、</p> <p>1101，平台拒绝类： 1100 1010 (202) 用户忙，MSRN 获取失败，平台挂机</p> <p>1100 1011(203) 用户去活，平台挂机 1100</p> <p>1100 (204) 用户在平台侧关机，平台挂机</p> <p>1100 1101 (205) 用户未开户，平台挂机</p> <p>1100 1110 (206) 小号不允许呼叫，平台挂机</p> <p>1100 1111 (207) 主号拨打小号，平台挂机</p> <p>1101 0001 (209) 主叫打小号带原始被叫，平台挂机</p>
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

● 推送响应

接收状态	is_success	ture：成功， false：失败
------	------------	-------------------

- 错误码：

错误码	错误信息	建议
400	UnSupport Method:	method 参数有误
403	Erro AppKey	appkey 错误
403	Error Sign	签名错误
202	具体的 Exception	请求接收，处理错误

- 请求示例：

http://172.27.111.69/msg?v=1.0&method=add_call_release×tamp=2014-08-18

%2016:59:11&app_key=abc&sign=BA9854BED1A2986B061E2713F403C752&phone_no=1860

0000000&secret_no=17000000000&call_type=1&peer_no=13900000000&call_id=1919&

ringing_time=2014-08-18%2016:59:10&call_time=2014-08-18%2016:59:11&release_time=2014-08-18%2016:59:21&release_dir=1&release_cause=17

需要说明一下，接口调用中的 appkey 和 secret 都由服务端来分配。

- 响应示例：

成功：{"response":{"is_success":true}} 失败：

{"response":{"is_success":false,"err_code":"400","err_msg":"呼叫类型不能为空！"}}}

2.3.3 业务交互（AXB 业务 BSS 接口）

ACBSS 主要处理的 AXB 业务请求和响应类型如下：

- 订购
- 退订
- 订购关系查询
- 更新

对于一个唯一的订购关系，AXB 业务场景下 BSS 采用如下字段来标识：

- 订购关系标识
AC 系统需要确保“订购关系标识”是唯一的。
- AXB 三元组
AXB 中的 A，X，B 形成唯一三元组。

2.3.3.1 订购请求和响应

- 订购请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	Ver		
	消息 ID	Msgid		
	时间戳	Ts		
	业务类型	service	acbss	
	消息类型	msgtype	axbsubreq	
	签名	Sid		

	保留字段	Rsvd		
	应用标识	appkey	第三方应用的 AppKey。 AXB 业务时必须设置，不能为空。	“appkey” : “GZAXB”
消息 体	真实号码	prtms	对应“产品类别”定义的 AXB 业务（简称 AXB 业务，下同）中的号码 A 注：以真实号码为主叫，AXB 业务中的 A 为主叫，X 为隐私号码，B 为对端号码。以下同。	
	隐私号码	acms	对应 AXB 业务中的号码 X，动态分配	
	其他号码	otherms	AXB 业务时可设置。对应 AXB 业务中的号码 B。 AC 小号场景不支持设置。	“ otherms ” : “ 861300571123 4 ”
	订购时间	subts	格式为 YYYYMMDDhhmmss。时间采用北京时间，24 小时制。	“subts” : “ 201407281459 21 ”
	退订时间	unsubts	格式为 YYYYMMDDhhmmss。时间采用北京时间，24 小时制。这个时间大于订购时间，如果时间到达则执行退订。 可选	“unsubts” : “ 201407281459 21 ”
	自动退订方式	unsubmethod	可选 0: 振铃退订 1: 挂机退订 如果退订时间不为空，则根据	“unsubmethod” : “1”

			退订时间退订的优先级最高。	
订购类型	producttype	现阶段支持的类型： 0：包年 1：包季 2：包月 3：按次 4：体验	“producttype”：“0”	
产品类别	productcat	本字段在 AXB 业务中必须设置。 0： 保留。实际分配给 AX(AC 小号) 11： AXB 12： 语音验证业务 13： AXYB 业务的 XYB	“productcat”：“11”	
录音控制	callrecording	可选。是否开通录音功能。 在本字段不出现的情况下，默认 0（不开通录音功能）。 0： 不开通录音功能 1： 开通录音功能	“callrecording”：“0”	
姓名	Name	必选	“name”：“张三”	
证件类型	cardtype	必选	“cardtype”：“ ”	
身份证号码	cardno	必选	“cardno”：“ ”	

● 订购响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		

	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	axbsubrsp	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。 可以为空，或者不出现。	
消息体	返回码	result	采用类似 http 的状态码风格。 参考 2.3.2	
	订购关系标识	subid		
	隐私号码	acms		

2.3.3.2 退订请求和响应

● 退订请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	Acbss	
	消息类型	msgtype	Axbunsubreq	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。 可以为空，或者不出现。	
消息体	订购关系标识	subid		
	真实号码	prtms	对应 AXB 业务中的号码 A	

	隐私号码	acms	对应 AXB 业务中的号码 X	
	其他号码	otherms	AXB 业务时必须设置。	
			对应 AXB 业务中的号码 B	
	产品类别	productcat	AXB 业务时必须设置。 11: AXB 12: 语音验证业务 13: AXYB 业务的 XYB	
	退订时间	unsubts	格式为 YYYYMMDDhhmmss。时间采用北京时间，24 小时制。	

● 退订响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	axbunsubrsp	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 appKey。 可以为空，或者不出现。	
消息体	返回码	result	参见 2.3.2“返回码定义”	
	订购关系标识	subid		

2.3.3.3 订购查询请求和响应

● 订购查询请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	axbsubqryreq	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。 可以为空，或者不出现。	
消息体	订购关系标识	subid	方式一	
	被保护的真实号码	prtms	方式二 注：otherms 在 AXB 业务时可设置。对应 AXB 业务中的号码 B。	
	隐私号码	acms		
	其他号码	otherms		

● 订购查询响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	

	消息类型	msgtype	axbsubqryrsp	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第 三 方 应 用 的 AppKey。 可以为空，或者不出现。	
消息体	返回码	result	参见 2.3.2 “返回码定义”	
	订购关系标识	subid		
	被保护的真实号码	prtms		
	隐私号码	acms		
	其他号码	otherms		
	订购时间	subts		
	订购产品类型	producttype		
	产品类别	productcat		
	录音控制	callrecording	可选。是否开通录音功能。 在本字段不出现的情况下，默认 0（不开通录音功能）。 0：不开通录音功能 1：开通录音功能	“callrecording”：“0”
	姓名	name		
	证件类型	cardtype		
	身份证号码	cardno		“cardno”：“ ”

2.3.3.4 订购更新请求和响应

- 订购更新请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	Acbss	
	消息类型	msgtype	Axbsubupdreq	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。 AXB 业务时必须设置, 不能为空。	“appkey” : “kuaidadi”
消息体	订购关系标识	subid	订购更新针对的现有订购关系的标识。必须与现有订购关系的内容一致。	
	真实号码	prtms	订购更新针对的现有订购关系的 A 号码。必须与现有订购关系的内容一致。	
	隐私号码	acms	订购更新针对的现有订购关系的 X 号码。必须与现有订购关系的内容一致。	
	其他号码	otherms	变更后的新号码 B。	“ otherms ” : “ 8613005711234 ”
	更新时间	subts	格式为 YYYYMMDDhhmmss。时间采用北京时间, 24 小时制。	“subts” : “ 20140728145921 ”

	产品类别	productcat	<p>本字段在 AXB 业务中必须设置为 11。AX 业务不支持。</p> <p>0： 保留。实际分配给 AX (AC 小号)</p> <p>11： AXB</p>	“productcat” : “11”
--	------	------------	------------------------------------------------------------------------------------	---------------------

● 订购更新响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	Acbss	
	消息类型	msgtype	axbsubupdrsp	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。 可以为空，或者不出现。	
消息体	返回码	result	采用类似 http 的状态码风格。 参考返回码定义。	
	订购关系标识	subid		

2.3.3.5 外呼 AXB 转接配置设置请求和响应

● 外呼 AXB 转接配置设置请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		

	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	outaxbtransfersetreq	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。 可以为空，或者不出现。	
消息体	隐私号码	acms	X 号码	
	转接号码	transfers	转接号码	
	转接号码设置 放音编码	transfervoicecode	设置转接号码的放音编码	
	来显控制	transfercalldisplay	转接号码的来显控制，默认 0 0:显示 X 号码 1:显示真实号码	

● 外呼 AXB 转接配置设置响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	Acbs	
	消息类型	msgtype	outaxbtransfersetrsp	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。	

			可以为空，或者不出现。	
消息体	返回码	result	参见 2.3.2 “返回码定义”	

2.3.3.6 外呼 AXB 转接配置删除请求和响应

● 外呼 AXB 转接配置删除请求

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	outaxbtransferdelreq	
	签名	sid		
	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。 可以为空，或者不出现。	
消息体	隐私号码	acms	X 号码	

● 外呼 AXB 转接配置删除响应

消息头	内容	JSON 对象名称	说明	举例
	版本号	ver		
	消息 ID	msgid		
	时间戳	ts		
	业务类型	service	acbss	
	消息类型	msgtype	Outaxbtransferdelr sp	
	签名	sid		

	保留字段	rsvd		
	应用标识	appkey	第三方应用的 AppKey。 可以为空，或者不出现。	
消息体	返回码	result	参见 2.3.2“返回码定义”	

2.4 返回码定义

IP 系统应该检查响应消息中的“返回码”。如果结果不是成功，则 IP 系统需要针对不同返回码进行处理。

2XX 成功类返回码

返回码	描述	定义
200	成功	请求报文成功被处理
其他	待定义	

4XX 客户端错误类返回码

返回码	描述	定义
400	请求报文语法错误	请求报文所携带的字段，不符合接口定义。比如 producttype 的值不在定义范围内。AC 系统需要检查字段的有效取值。
401	认证未通过	请求的来源或者请求的签名未通过校验。
402	隐私号码已经使用	请求中要求绑定的隐私号码已经被分配给其他用户。
403	订购关系不存在	AC 系统提交的请求中提供的“订购关系标识”，在互联网系统中不存在。
404	绑定数据不一致	请求中提供的“订购关系标识”在 AC 系统中查询获得的“真实号码”和“隐私号码”绑定关系，与请求中提供的“真实号码”与“隐私号码”不一致。

429	请求过多	AC 系统在一定时间内提交了过多的请求。AC 系统在访问某些特定服务接口时，可能会出现该错误。
其他	待定义	

5XX 服务端错误类返回码

返回码	描述	定义
501	业务处理超时	互联网系统内部处理引起的超时。AC 系统可以尝试重新提交请求。
502	服务暂时不可用	互联网系统当前不可用，可能是因为系统负载过重，或者暂时停机。
503	服务接口被禁止	互联网系统已经关闭相关服务。后续对于该服务的请求是被禁止的。
其他	待定义	

2.5 附:签名算法

2.5.1 Java

```
public static String signature(Map<String, String> params, String secret,
    String signName) {
    String result = null;
    if (params == null)
        return result;

    // 1. 把字典按Key的字母顺序排序
    Map<String, String> treeMap = new TreeMap<String, String>();
    treeMap.putAll(params);

    // 2. remove sign parameter
    treeMap.remove(signName);

    // 3 把参数串起来
    Iterator<String> iter = treeMap.keySet().iterator();
    StringBuffer origin = new StringBuffer(secret);
    while (iter.hasNext()) {
        String name = (String) iter.next();
        origin.append(name).append(params.get(name));
    }

    // 4. 加密: MD5 + 大写&十六进制
    try {
        MessageDigest md = MessageDigest.getInstance("MD5");
        result = byte2hex(md.digest(origin.toString().getBytes("utf-8")));
    } catch (Exception e) {
        throw new java.lang.RuntimeException("sign error !");
    }

    return result;
}

private static String byte2hex(byte[] b) {
    StringBuffer hs = new StringBuffer();
    String stmp = "";
    for (int n = 0; n < b.length; n++) {
        stmp = (java.lang.Integer.toHexString(b[n] & 0xFF));
        if (stmp.length() == 1)
            hs.append("0").append(stmp);
        else
            hs.append(stmp);
    }
    return hs.toString().toUpperCase();
}
```

2.5.2 PHP

```
public static function sign($params)
{
    unset($params['sid']);
    unset($params['sign']);
    //按照 key 进行排序
```

```
ksort($params);

$signParams = [];
foreach ($params as $key => $val) {
    $signParams[] = $key . $val;
}
$signParams = self::APP_SECRET . implode('', $signParams);

$sign = strtoupper(md5($signParams));
return $sign;
}
```