

京东开放 API 接入与规范文档

文档版本 V2.6
发布日期 2017-01
更新时间 2018-12-03

版本记录

序号	版本	更改说明	日期	人员
1	0.1	初始版本	2017-01-17	柳刘
2	1.1	授权及签名更新	2017-01-20	李长春
3	1.2	增加设备接入	2017-02-14	雒根雄
4	2.1 beta	重新整合文档	2017-04-19	柳刘
5	2.1	标注授权流程及请求格式	2017-04-27	李长春
6	2.2	修正信息错误	2017-06-02	李长春
7	2.3	完善内测阶段申请注册流程	2017-06-09	李长春
8	2.4	更新注册流程和关键说明	2017-11-27	尚进、毕博
8	2.5	加入 3.1 接入安全要求，更新描述 整合主控被控一体化授权	2017-11-27	李长春、柳刘
9	2.6	access_token 使用说明，格式调整	2017-12-12	柳刘、李长春
10	2.8	新增更新场景和卡片相关接口，补充 消息推送相关接口	2018-09-29	王洪岳
11	2.9	增加消息推送类型，包括设备重命名、设备绑定/分享、设备解绑/解除分享/子设备解绑、场景新增、场景删除和场景变更。增加按照用户订阅消息。在消息结构中增加用户 id。	2018-12-03	王洪岳

版权所有 ©京东智能



JD Alpha

目 录

1	概述	5
1.1.	开放架构	5
1.2.	术语	5
2	接入流程	6
2.1	总体流程介绍	6
2.2	注册京东账号	6
2.3	申请成为企业开发者	7
2.4	应用管理	9
3	接入和调用规范	13
3.1.	接入安全要求	13
3.2.	接口调用规范	13
3.3.	错误信息	16
4	用户授权	16
4.1.	基于 Server 的授权	17
4.2.	二维码授权	19
	获取二维码	20
	获取二维码授权结果	21
4.3.	主控被控一体化授权	22
	设备访问授权接口	22
	刷新授权接口	23
	解除授权接口	24
4.4.	解除授权	24
4.5.	access_token 使用和刷新	25
	access_token 使用	25
	token 失效及处理策略	25
	刷新授权接口（适用于 Server 和二维码授权）	25
4.6.	授权业务错误码	26
5	设备访问服务 OpenAPI	27
5.1	设备访问域名与 API 列表	27
5.2	设备访问 API 说明及示例	27
	获取用户基本信息	27
	获取设备列表	28
	控制设备	28
	获得设备信息	29
	获得批量设备信息	30
	获取卡片列表	30
	获取场景列表	33
	执行场景	35
	获取场景详情	36
	获取用户详情	36
6	数据推送服务 OpenAPI	37
6.1	概述	37
6.2	数据推送订阅	38
	订阅数据	38
	取消订阅	39
6.3	数据推送规范	40
	回调第三方的数据返回格式	40
	回调第三方返回的常见错误及说明	41

6.4	数据推送接口.....	41
	设备数据快照推送.....	41
	场景消息推送	41
7	设备访问服务 SDK	46

JD Alpha

1 概述

京东开放平台将京东的 IoT 能力、Skill 接入能力、赋能或开放给第三方应用、APP、云或主控设备等。

目前，本文档规范适用于 IoT 开放和京东赋能授权两大类 API，最新能力以对接项目为准。

1.1. 开放架构

如图 1 所示：

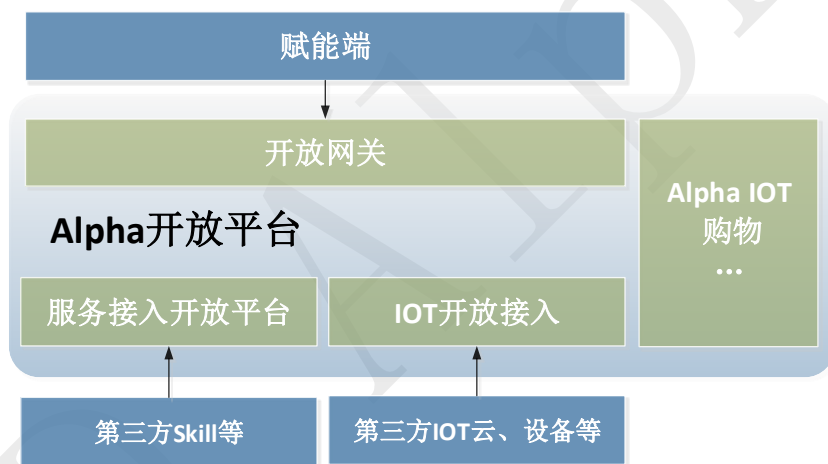


图 1 京东开放平台示意图

1.2. 术语

赋能端：泛指所有接入京东，使用京东服务的应用/系统/设备/云等

2 接入流程

2.1 总体流程介绍

总体流程包括：

- 注册京东账号
- 申请成为京东开放平台的企业开发者
- 创建应用
- 提交应用审核，审核通过后分配测试联调的 AppKey（可多次审核）
- 开发联调
- 应用发布，确认正式上线，分配正式线上的 AppKey

特别说明：开放服务目前处于内测阶段，开发者申请成功后请联系对接产品经理，开通内测权限。

其它说明：目前，开放服务暂时无法提供开发联调环境，因此开放服务开发者申请的测试环境使用的 AppKey 每天最多可以有 2000 次的接口访问量，并且在开发者服务正式上线后，会定期回收对应的 AppKey。

2.2 注册京东账号

接入开放服务要先成为京东开放平台的开发者，成为开发者首先需要京东账号，如果已经有相应的京东账号，请跳过本节。

注册入口可以通过京东主站 www.jd.com 或者京东商城 App，图 2 为网站注册界面。

图 2 京东新用户注册

2.3 申请成为企业开发者

特别说明：开放服务目前处于内测阶段，开发者申请成功后请联系对接产品经理，开通内测权限

1、尚未注册开发者

如果您尚未注册成为京东开放平台的开发者，访问 <http://smartdev.jd.com>，登陆已注册的京东账号，点击顶部的『成为开发者』-『企业开发者』进行申请。

图 3 为开发者申请入口，图 4 为开发者类型选择页面



图 3 开发者申请入口



图 4 开发者类型选择

2、已注册个人开发者

如果您已经成为了京东开放平台的个人开发者，可进入『用户中心』，点击『升级成为企业开发者』进行账号升级。

图 5 为用户中心入口，图 6 为升级企业开发者入口



图 5 用户中心入口



图 6 升级企业开发者入口

2.4 应用管理

1、应用入口

访问 <http://devsmart.jd.com/dev/index> 点击左侧控制菜单的『应用中心』-『应用管理』，进入应用管理页面。

图 7 为应用中心入口

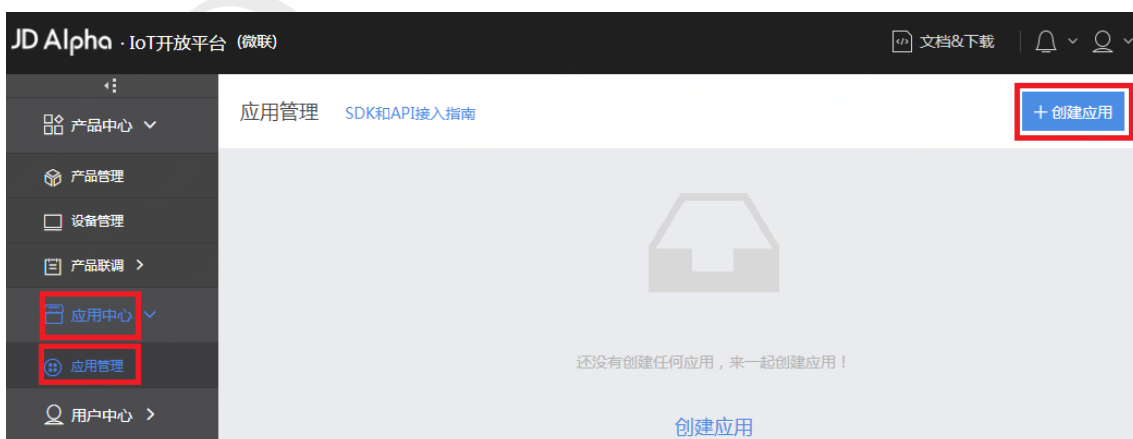


图 7 应用中心入口

2、创建应用

点击『创建应用』，进入应用的相关信息录入页面，填写应用信息。

图 8 为应用信息录入

图 8 应用信息录入

3、应用列表

创建成功的应用会出现在应用列表中，可以重新编辑应用、提交审核或删除，应用信息确认无误后，即可点击『提交审核』，提交应用审核。

图 9 为应用编辑状态的效果



图 9 应用编辑状态的效果

4、应用审核

应用提交审核后，我们会收到审核通知，尽快会对您的应用做出审核。

图 10 为应用审核状态效果



图 10 应用审核状态效果

5、开发调试

应用审核通过后，会以邮件或者站内信的方式通知您，请及时登录京东开放平台的『应用中心』-『应用管理』，点击『查看』，获取分配的测试环境的 AppKey 等信息进行开发调试

特别说明：开发调试过程中，如果应用的信息需要修改，可以点击『编辑』进行调整，信息调整后需要重新提交审核

图 11 为应用审核成功状态的效果



图 11 应用审核成功状态的效果

6、应用发布

开发调试通过后，请点击『发布』，将已经联调通过的应用发布到线上正式环境，并获得正式上线上的 AppKey 等信息

特别说明：应用发布后不能更改，请谨慎操作

图 12 为应用发布状态的效果



图 12 应用发布状态的效果

3 接入和调用规范

3.1. 接入安全要求

安全作为用户使用京东 IOT 和各类服务的第一屏障，直接影响用户的使用安全，直接影响用户对平台、系统、服务和设备的信任和使用抉择。为此，京东要求严格保障用户数据、隐私等所有信息的安全，要求对接京东的相关系统、服务及设备实现安全不得低于本规范要求。

本规范只从技术上明确安全技术要求，包括但不限于所描述的场景，实际处理以本规范及法务和商务中的最高标准执行。

3.1.1 针对不安全接入的措施

- 1) 限制某 AppKey 接入的请求访问。从技术上，京东平台（云端）会针对各类攻击进行防御和限制，因此有可能对存在可疑安全问题的赋能端进行限制访问的操作，如限制某 Appkey 对应的所有应用无法访问京东任何服务等。
- 2) 重新换发 AppSecret。针对可疑赋能端的请求，京东有权要求其及时更换 AppSecret，以保证正常的访问接入，屏蔽和发现非法接入。

3.1.2 数据保存和访问安全：

- 1) AppSecret 作为赋能端的全局私密信息，只能在一般人无法触及的服务器，或者云端或者保密设备上保持和访问，比如明确只有某些（某个）运维管理员或系统开发维护人员才有权访问的系统/设备的安全位置。**不能保存也不能发送**到最终用户使用的任何普通人可以接触的终端上，以防止黑客或恶意人员破解或者攻击。
- 2) 用户的 Access Token 作为用户的身份密码，除非明确说明的场景，不能长久保存在赋能端终端上，赋能端的用户终端需要使用时，应当通过安全的方式从赋能端的云端服务等接口获取。

3.1.3 访问通道安全

- 1) 对于用户的 Access Token 等信息，从京东获取的相关信息，除非明确说明，在赋能端的服务和终端之间，都应该通过安全的数据访问通道进行传输。推荐所有的请求使用 HTTPS 进行通信。
- 2) 除非得到京东和用户的授权，否则赋能端无权向其他方开放相关数据接口，以保证用户隐私和数据安全。

3.2. 接口调用规范

访问京东开放平台的开放服务，只需要通过发送 HTTPS 请求到开放网关，即能获取到所需的数据。

调用 API 时，必须传入系统级参数和对应的应用级参数。

接口调用请求方法：POST

Content-Type: application/x-www-form-urlencoded; charset=utf-8

1) 系统级参数

当请求或者数据错误的时候，开放平台会返回响应的错误信息，错误信息带有错误码，错误码分为两类：平台错误码和业务错误码。

名称	类型	是否必须	描述
method	String	是	API 接口名称
access_token	String	否	分配给用户的令牌，通过授权获取，方法详见授权说明
app_key	String	是	应用的 app_key
sign	String	是	见详细说明
timestamp	String	是	时间戳，格式为 yyyy-MM-dd HH:mm:ss，例如：2011-06-16 13:23:30。京东 API 服务端允许客户端请求时间误差为 6 分钟
v	String	是	API 协议版本，可选值:2.0.

注：系统级参数直接组装在 URL 参数中，调用 API 时，将所有的参数转换成 UTF-8 编码，然后进行拼装

2) 应用级参数（API 应用级参数）

名称	类型	是否必须	示例值	描述
fields	String	否		设置需要返回的字段列表，字段之间用","分隔。

如下为应用级参数拼装示例

名称	描述	示例	
360buy_param_json	String	标准 json 类型, 而且是按照字母先后顺序排序的	{ "end_date": "2012-05-16 17:03:56", "optional_fields": "vender_id", "page": "1", "page_size": "20", "start_date": "2012-05-14 17:03:56" }

注：应用级参数作为请求的 Body 传送

3) 参数签名算法。

调用 API 时需要对请求参数进行签名，京东开放平台的开放网关会验证请求参数是否合法。签名规则如下

- 1) 所有请求参数按照字母先后顺序排列

例如：将 access_token, app_key, method, timestamp, v 排序为
access_token, app_key, method, timestamp, v

- 2) 把所有参数名和参数值进行拼装

例如：access_tokenxxxapp_keyxxxmethodxxxxxxtimestampxxxxxxvxx

- 3) 把 appSecret 夹在字符串的两端

例如: appSecret+XXXX+appSecret

4) 使用 MD5 进行加密, 再转化成大写

5) 示例 (以下示例只体现逻辑):

假设调用服务方法:

```
360buy.order.search, app_key=yourappkey、appSecret=yourappSecret、
access_token=yourtoken
```

a. 输入参数

```
access_token=yourtoken
app_key=yourappkey
method=360buy.order.search
timestamp=2011-07-20 11:10:04
v=2.0
360buy_param_json=
{
  "end_date":"2012-05-16 17:03:56",
  "optional_fields":"vender_id",
  "page":"1",
  "page_size":"20",
  "start_date":"2012-05-14 17:03:56"
}
```

b. 把参数按照字母顺序排列

```
360buy_param_json=
{
  "end_date":"2012-05-16 17:03:56",
  "optional_fields":"vender_id",
  "page":"1",
  "page_size":"20",
  "start_date":"2012-05-14 17:03:56"
}
access_token=yourtoken
app_key=yourappkey
method=360buy.order.search
timestamp=2011-07-20 11:10:04
v=2.0
```

c. 连接参数名与参数值, 并在首尾加上 appSecret

授权接口: (示例)

```
yourappSecret360buy_param_json{"end_date":null,"optional_fields":null
,"page":"1","page_size":"200","start_date":null}access_tokenyourtoken
app_keyyourappkeymethod360buy.order.searchtimestamp2012-06-
2116:28:02v2.0yourappSecret
```

注:

- 以上字符串不允许有跨行, 粘贴时请特别注意, 除日期和时间中间的空格之外, 不允许在其它位置出现空格。
- 时间戳 timestamp 最好就填写当前时间的前几分钟 (必须在 6 分钟之内)
- 签名时嵌套 JSON 需要进行格式转义
- 签名是 key 和 value 直接拼接, 无等号
- MD5 签名时要按照 UTF-8 格式进行

d. MD5 加密后转成大写: EDBE87FB9640EDD681FEB79D858AAC0

4) 最终授权 API 调用 URL 示例 (仅为示例):

```
https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&app_key=123456780233FA31AD94AA59CFA65305&method=360buy.order.search&timestamp=2013-05-30%2000:00:00&v=2.0&sign=E981702AF260F37FCCD7D60FD19AAEA7

360buy_param_json={"end_date":"2013-12-01 00:00:00","order_state":"WAIT_SELLER_STOCK_OUT,WAIT_GOODS_RECEIVE_CONFIRM","page":"1","page_size":"20","start_date":"2013-05-01 00:00:00"}
```

注：以上示例中使用域名仅供参考，请根据对应接入规范使用。比如通用服务域名为 smartopen.jd.com，设备管理集成域名则为 devopen.jd.com

3.3. 错误信息

当请求或者数据错误的时候，开放平台会返回响应的错误信息，错误信息带有错误码，错误码分为两类：平台错误码和业务错误码。

- 平台错误返回的格式示例如下：

```
{
  "error_response": {
    "code": -3,
    "zh_desc": "服务器系统错误",
    "en_desc": "ERROR",
    "errorMsg": "参数错误!"
  }
}
```

- 业务错误码的格式示例如下：

```
{
  "jingdong_smart_api_voice_interaction_responce": {
    "code": "0",
    "result": "{\"code\":56,\"errorMsg\":\"参数错误!\"}"
  }
}
```

4 用户授权

京东开放平台支持多种授权方式，并支持解除授权。

4.1. 基于 Server 的授权

● 授权说明

在接入京东平台时，需要获取一些与用户紧密相关的数据，为保证数据的安全性和隐私性，需要取得用户的同意，引导用户授权。采用国际通用的 OAuth2.0 标准协议。如果要了解更多关于 OAuth2.0 的技术说明，请参考官方网站 <http://oauth.net/2/>。

授权方式当前仅支持集成 SDK 的方式进行授权。

(1) Authorization Code

流程要求开发者(包括企业、厂商、ISV)有 web 服务器，能够保持应用本身的密钥以及状态，可以通过 https 直接访问京东的授权服务器。

(2) Refresh Token

通过 Authorization Code 授权流程，获取到 Access token 以及 Refresh token。Access token 有一定的时效，在 Access token 的有效内可以通过 Refresh token 来延长 Access token 的时效。如果超过授权时效，需要引导用户重新授权。

正式环境授权地址：<https://smartopen.jd.com/oauth/authorize?>（需要拼接参数，无法直接访问）

授权有效期说明：如果应用 appkey 状态为“在线测试”，access token 有效期只有 24 小时，“上线运行”状态的应用 access token 有效期为 30 天，当有效期失效后，应当引导用户重新授权。

● 获取 Authorization Code 参数说明

此流程要求开发者有 web 服务器，能够保持应用本身的密钥以及状态，可以通过 https 直接访问京东开放平台的授权服务器。

参数名称	参数选项	描述
response_type	必须	此流程下，该值固定为 code
client_id	必须	即创建应用时的 Appkey（从控制台->管理应用中获取）
redirect_uri	必须	即应用的回调地址，必须与 APP 注册时填写的值一致。
State	可选	状态参数，由 ISV 自定义，颁发授权后会原封不动返回
Scope	可选	权限参数，API 组名串。多个组名时，用", "分隔， 目前支持参数值：read
View	可选	移动端授权，该值固定为 wap；非移动端授权，无需传值
Signature	必须	如果为 ios，不用传；Android 传 apk 的签名
Identity	必须	如果为 ios，传入 bundle id；如果为 Android，传入 package name
Timestamp	必须	格式为：yyyy-MM-dd HH:mm:ss

● 获取访问令牌 Access token 参数说明

参数名称	参数选项	描述
grant_type	必须	授权类型，此流程下，该值固定为 authorization_code
code	必须	授权请求返回的授权码
redirect_uri	必须	即应用的回调地址，必须与 APP 注册时填写的值一致。默认为 https://smartopen.jd.com/oauth/code
client_id	必须	即创建应用时的 Appkey（从控制台->管理应用中获取）
client_secret	必须	即创建应用时的 Appsecret（从控制台->管理应用中获取）
state	可选	状态参数，由 ISV 自定义，颁发授权后会原封不动返回

● 基本流程

步骤1 引导使用应用的用户登录到如下地址获取 code，请求 url 如下：

```
https://smartopen.jd.com/oauth/authorize?response_type=code&client_id=YOUR_CLIENT_ID&redirect_uri=YOUR_REGISTERED_REDIRECT_URI&state=YOUR_CUSTOM_CODE&signature=YOUR_SIGNATURE&identity=YOUR_IDENTITY&timestamp=YOUR_TIMESTAMP
```

步骤2 用户登录授权

如果用户没有登录京东账号，则跳转到京东登录页面：



登录用户将会被引导至授权页面：

需要以下授权

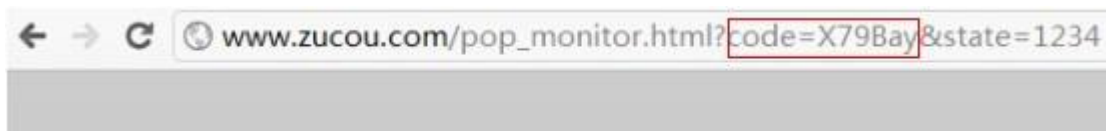
- 允许访问你的头像、昵称等
- 允许访问你的所有设备，包括控制及状态
- 允许新加入的设备自动授权

确认授权

取消

步骤3 获取授权码 code，如果用户同意授权，页面跳转至应用的回调地址，同时返回授权码 code

以及 state 参数。



如果用户取消授权，则页面跳转至应用的回调地址，同时返回如下错误信息：error=access_denied

← → ↻ www.zucou.com/pop_monitor.html?error=access_denied&error_description=User%20denied%2

步骤4 获取访问令牌 access_token

用上一步获取的 code 码和注册应用时分配的 AppSecret，由服务器端通过 Http Get 方式向京东开放平台换取 Token（访问令牌，即 Access token），会以 json 文本的形式返回数据。

最后生成的 http 请求 url 如下：

```
https://smartopen.jd.com/oauth/token?grant_type=authorization_code&client_id=YOUR_CLIENT_ID&redirect_uri=YOUR_REGISTERED_REDIRECT_URI&code=GET_CODE&state=YOUR_CUSTOM_CODE&client_secret=YOUR_APP_SECRET
```

注：code 只允许使用一次，如获取 token 失败，请重新授权获取 code，再换取

access_token

步骤5 从 http 返回结果中，得到 AccessToken 和 Refresh_token。

```
{
  "access_token": "adsfasfsafasfaba-961f-3ffb444ee094",
  "code": 0,
  "expires_in": 315535999,
  "refresh_token": "e4cab69f-5ca2-4ada-9199-9778e6581c99",
  "scope": "read",
  "time": "1390909039384",
  "token_type": "bearer",
  "uid": "",
  "user_nick": "hello_kitty",
  "avatar": "https://i.jd.com/commons/img/no-img_mid_.jpg"
}
```

注释：

```
uid: 授权用户对应的京东 ID
user_nick: 授权用户对应的京东昵称
expires_in: 失效时间（从当前时间算起，单位：秒）
time: 授权的时间点（UNIX 时间戳，单位：秒）
token_type: token 类型（暂无意义）
avatar: 用户头像
```

4.2. 二维码授权

● 二维码授权原理

基于开放接口获取特定二维码信息，由京东账户登录的应用将二维码信息与登录用户授权信息进行关联，从而分配唯一的授权信息包括 access_token 等信息。

● 交互流程

第三方应用通过接口获取二维码→展示二维码并使用小京鱼 App 扫描二维码（如果微信等第三方应用扫描二维码，会自动跳到 App 下载页面）→第三方应用 10s 后每 2s 拿二维码信息通过接口轮询授权结果。（二维码 **120s** 内有效）

获取二维码

功能：获取授权临时二维码

方法名：jingdong.smart.api.auth.getqrcode

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.auth.getqrcode
```

HTTP POST BODY:

```
360buy_param_json={ "device_id": "3deffdd", "user_identity": "ryan", "device_name": "xx 手表", "os_type": "android", "os_ver": "19", "os_ver_name": "4.4 Yun OS", "device_type": "phone", "app_name": "xx 京东", "app_package_name": "com.weilian.samsung", "app_version": "13", "app_channel": "wandoujia" }
```

请求参数解释：

device_id: 可以唯一作 appKey 上用户的标识
user_identity: 用户的登录 ID. 可以为空
device_name: 设备名称. 可以为空. 例如: 三星手表.
os_type: 操作系统类型. 可以为空. 例如: android, ios
os_ver: 操作系统版本. 可以为空. 例如 android 对应物"19"
os_ver_name: 操作系统名称. 可以为空. 例如: Yun Os
device_type: 设备类型. 可以为空. 包括: watch, phone, tv
app_name: 应用名称. 可以为空.
app_package_name: 应用包名. 可以为空.
app_version: 应用程序的版本. 可以为空.
app_channel: 应用程序的渠道, 可以为空.

HTTP RESPONSE:

```
{
  "jingdong_smart_api_auth_getqrcode_response": {
    "code": "0",
    "result":
    "{ \"code\": 200, \"data\": { \"auth_code\": \"https://smartcloud.jd.com/download?o=abf39af13fc67039d6d971dbb78cb170\", \"expires_in\": 120 }, \"errorMsg\": \"OK\" }"
  }
}
```

返回结果解释：

auth_code: 授权的二维码信息
expires_in: 二维码超时时间

获取二维码授权结果

功能：获取二维码授权结果

方法名：jingdong.smart.api.auth.getresult

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method= jingdong.smart.api.auth.getresult
```

HTTP POST BODY:

```
360buy_param_json={"device_id": "test-device-id", "user_qrcode": "https://smartcloud.jd.com/download?o=abf39af13fc6703170"}
```

请求参数解释：

device_id: 可以唯一作 appKey 上用户的标识
user_qrcode: 授权的二维码信息

HTTP RESPONSE:

```
{
  "jingdong_smart_api_auth_getresult_response": {
    "code": "0",
    "result":
    "{\"code\":200,\"data\":{\"access_token\":\"adsfasfsafasfaba-961f-3ffb444ee094\",\"code\":0,\"expires_in\":315535999,\"refresh_token\":\"e4cab69f-5ca2-4ada-9199-9778e6581c99\",\"scope\":\"read\",\"time\":\"1390909039384\",\"token_type\":\"bearer\",\"uid\":\"\",\"user_nick\":\"hello_kitty\",\"avatar\":\"https://i.jd.com/commons/img/no-img_mid.jpg\"},\"errorMsg\":\"OK\"}"
  }
}
```

返回结果解释：

access_token: 用户访问令牌
expires_in: 用户 access_token 超时时间
refresh_token: 用户的刷新 token, access_token 过期后, 可用其刷新, 每月限制为 30 次
time: 时间
uid: 京东唯一的用户 ID,
user_nick: 用户的昵称
avatar: 用户头像

4.3. 主控被控一体化授权

本授权用于赋能端既可以作为主控设备又作为接入 IOT 云的被控智能设备时，支持主控端用户授权。本文档假设设备已经完成被控设备接入 IOT 的认证和用户绑定，因此下文仅涉及设备作为主控设备时与京东授权中心的交互流程。

前提和安全约束

- 1) 用户在小京鱼 APP 上已经完成对设备的激活与绑定
- 2) 设备正常联网并且具有 HTTPS 访问能力。
- 3) 设备请求接口时，使用设备的 `access_key`（设备激活时与 `feed_id` 一起写入）代替 `appSecret` 来完成签名。
- 4) 设备在激活后，要求在 24 小时内完成授权信息的访问获取；
- 5) 设备在访问时，均需要携带随机种子 `random` 生成的签名信

接口

设备访问授权接口

功能：设备发起用户授权。

方法名：`jingdong.smart.api.device.auth`

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.device.auth
```

HTTP POST BODY:

```
360buy_param_json={"device_id":"test-device-id","feed_id":"13343434311", "random": "xxxxxxxxxx"}
```

请求参数解释：

`device_id`: 设备的 ID。
`feed_id`: 设备的 `feed_id`，京东在设备激活后分配
`random`: 用于签名的随机字符串，要求 ≥ 32 位

HTTP RESPONSE:

```
{
  "jingdong_smart_api_device_auth_response": {
    "code": "0",
    "result":
      "{\"code\":200,\"data\":{\"access_token\":\"adsfasfsafasfaba-961f-3ffb444ee094\",\"code\":0,\"expires_in\":315535999,\"refresh_token\":\"e4cab69f-5ca2-4ada-9199-9778e6581c99\",\"scope\":\"read\",\"time\":\"1390909039384\",\"token_type\":\"bearer\",\"uid\":\"\",\"user_nick\":\"hello_kitty\",\"avatar\":\"https://i.jd.com/commons/img/no-
```

```
img_mid_.jpg\"},\\"errorMsg\\":\\"OK\\"}"
    }
}
```

返回结果解释：

access_token: 用户访问令牌
 expires_in: 用户 access_token 超时时间
 refresh_token: 用户的刷新 token, access_token 过期后, 可用其刷新, 每月限制为 30 次
 time: 时间
 uid: 京东唯一的用户 ID
 user_nick: 用户的昵称
 avatar: 用户头像 URL

刷新授权接口

功能：刷新用户授权信息。

方法名：jingdong.smart.api.device.auth.refresh

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD179
01B5BB89&timestamp=2015-12-
25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&me
thod=jingdong.smart.api.device.auth.refresh
```

HTTP POST BODY:

```
360buy_param_json={"access_token":"26a1b61501a54a2e9abf66d0f4b10"
,"refresh_token":"2343311222fsafa","device_id":"test-device-id",
"feed_id": "13343434311", "random": "xcvvcvcasddd"}
```

请求参数解释：

access_token: 用户访问令牌
 device_id: 设备的 ID
 refresh_token: 用户的刷新令牌
 feed_id: 设备的 feed_id, 京东在设备激活后分配
 random: 用于签名的随机字符串, 要求>=32 位

HTTP RESPONSE:

```
{
  "jingdong_smart_api_device_auth_refresh_response": {
    "code": "0",
    "result":
    "{\\"code\\":200,\\"data\\":{\\"access_token\\":\\"adsfasfsafasfaba-
961f-
3ffb444ee094\\",\\"code\\":0,\\"expires_in\\":315535999,\\"refresh_toke
n\\":\\"e4cab69f-5ca2-4ada-9199-
9778e6581c99\\",\\"scope\\":\\"read\\",\\"time\\":\\"1390909039384\\",\\"to
ken_type\\":\\"bearer\\",\\"uid\\":\\"\\",\\"user_nick\\":\\"hello_kitty\\",
\\"avatar\\":\\"https://i.jd.com/commons/img/no-
img_mid_.jpg\\",\\"errorMsg\\":\\"OK\\"}"
  }
}
```

返回结果解释：

access_token: 用户访问令牌

expires_in: 用户 access_token 超时时间
 refresh_token: 用户的刷新 token, access_token 过期后, 可用其刷新, 每月限制为 30 次
 time: 时间
 uid: 京东唯一的用户 ID
 user_nick: 用户的昵称
 avatar: 用户头像 URL

解除授权接口

功能: 解除用户授权信息。

方法名: jingdong.smart.api.device.auth.release

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.device.auth.release
```

HTTP POST BODY:

```
{
  "360buy_param_json": {
    "access_token": "26a1b61501a54a2e9abf66d0f4b10",
    "device_id": "test-device-id",
    "feed_id": "13343434311",
    "random": "xcvvcvcasddd"
  }
}
```

请求参数解释:

access_token: 用户访问令牌
 device_id: 设备的 ID
 feed_id: 设备的 feed_id, 京东在设备激活后分配
 random: 用于签名的随机字符串, 要求>=32 位

HTTP RESPONSE:

```
{
  "jingdong_smart_api_device_auth_release_response": {
    "code": "0",
    "result": "{\\\"code\\\":200,\\\"errorMsg\\\":\\\"OK\\\"}"
  }
}
```

4.4. 解除授权

功能: 解除用户授权信息。

方法名: jingdong.smart.api.auth.release

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.auth.release
```

HTTP POST BODY:


```
360buy_param_json={"access_token":"26a1b61501a54a2e9abf66d0f4b10",
"device_id":"test-device-id"}
```

请求参数解释：

access_token：用户访问令牌
device_id：可以唯一作 appKey 上用户的标识。

HTTP RESPONSE:

```
{
  "jingdong_smart_api_auth_release_response": {
    "code": "0",
    "result": "{\"code\":200,\"errorMsg\":\"OK\"}"
  }
}
```

4.5. access_token 使用和刷新

access_token 使用

当接口需要授权才能访问时，接口参数中需要使用 access_token。如接口说明中『是否授权』选择『是』的时候。

token 失效及处理策略

- 1) 访问需要授权的接口时，接口返回错误码：

30005	用户 token 已过期
-------	--------------

则说明 access_token 过期，则使用『刷新授权接口』接口重新获取用户授权等信息。

- 2) refresh_token 失效：当接口返回 refresh_token 失效，则应引导用户重新登录授权。如调用『刷新授权接口』时返回如下错误码：

30007	用户 refreshToken 已过期
-------	---------------------

- 3) 对于 access_token 失效短的情况，系统应在 token 失效前，定期主动调用『刷新授权接口』。例如，30 分钟失效的 token，建议在 25 到 30 分钟之间请求刷新接口，避免用户使用时出现失效，提升用户体验。
- 4) 京东用户修改密码后，所有 token 会失效，需要重新引导用户登录授权。

刷新授权接口（适用于 Server 和二维码授权）

功能：根据 refresh token，在 access_token 即将过期的时候换取新的 access_token。

方法名：jingdong.smart.api.auth.refresh

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.auth.refresh
```

HTTP POST BODY:

```
360buy_param_json={"access_token":"26a1b61501a54a2e9abf66d0f4b10","refresh_token":"edb96e3c874e472454a6f0", "device_id":"test-device-id" }
```

请求参数解释：

access_token: 用户访问令牌，必传
device_id: 可以唯一作 appKey 上用户的标识。
refresh_token: 用户的刷新 token，access_token 过期后，可用其刷新，每月每个用户限制为 30 次

HTTP RESPONSE:

```
{
  "jingdong_smart_api_auth_refresh_response": {
    "code": "0",
    "result": "{ \"code\":200, \"data\":{ \"access_token\": \"adsfasfsafasfaba-961f-3ffb444ee094\", \"code\":0, \"expires_in\":315535999, \"refresh_token\": \"e4cab69f-5ca2-4ada-9199-9778e6581c99\", \"scope\": \"read\", \"time\": \"1390909039384\", \"token_type\": \"bearer\", \"uid\": \"\", \"user_nick\": \"hello_kitty\", \"avatar\": \"https://i.jd.com/commons/img/no-img_mid.jpg\" }, \"errorMsg\": \"OK\" }"
  }
}
```

返回结果解释：

access_token: 用户访问令牌
expires_in: 用户 access_token 超时时间
refresh_token: 用户的刷新 token，access_token 过期后，可用其刷新，每月限制为 30 次
time: 时间戳
uid: 京东唯一的用户 ID,
user_nick: 用户的昵称
avatar: 用户头像

4.6. 授权业务错误码

错误码	中文描述
30002	等待用户扫码
30003	获取用户授权信息超出限制
30004	未找到授权码对应的信息
30005	用户 token 已过期
30006	用户 refreshToken 错误
30007	用户 refreshToken 已过期
30008	系统繁忙，请重试！
30009	操作失败，请重试！
30010	用户授权二维码失效，请刷新二维码
30011	APP 不支持二维码授权
30012	二维码已扫描
30013	用户刷新 accessToken 次数达到限制！
30014	未发现授权用户！
30015	授权已取消！
30016	请求参数非法
30017	App 非法

5 设备访问服务 OpenAPI

设备访问接入相关 API，默认情况下均需要用户授权。

5.1 设备访问域名与 API 列表

域名: smartopen.jd.com

API 列表:

jingdong.smart.api.userinfo.get	获取用户基本信息
jingdong.smart.api.device.list	获取设备列表
jingdong.smart.api.control	控制设备
jingdong.smart.api.snapshot.get	获得设备信息
jingdong.smart.api.snapshot.batch.get	获得批量设备信息
jingdong.smart.api.user.scenes.list	获取场景列表
jingdong.smart.api.user.cards.list	获取卡片列表
jingdong.smart.api.scene.execute	执行场景
jingdong.smart.api.scene.detail.get	获取场景详情
jingdong.smart.api.user.detail.get	获取用户详情

5.2 设备访问 API 说明及示例

获取用户基本信息

功能: 获取用户基本信息概要, 包括昵称、头像等

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.userinfo.get&access_token=d757ad6e-2be1-4495-b11b-4bb3e3b9a206
```

HTTP POST BODY:

```
360buy_param_json={}
```

HTTP RESPONSE:

```
{
  "jingdong_smart_api_userinfo_get_response": {
    "code": 0,
    "result": "{\"code\": \"200\", \"data\": {\"avatar\": \"https://i.jd.com/commons/img/no-img_mid.jpg\", \"user_nick\": \"doge\"}, \"errorMsg\": \"ok\"}"
  }
}
```

返回结果解释：

```
avatar: 用户头像
user_nick: 用户昵称
```

获取设备列表

功能：获取当前用户授权的设备列表

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=0AB28F861CFD4BD3EA8FD179
01B5BB89&timestamp=2015-12-
25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&me
thod=jingdong.smart.api.device.list&access_token=d757ad6e-2be1-
4495-b11b-4bb3e3b9a206
```

HTTP POST BODY:

```
360buy_param_json={}
```

HTTP RESPONSE:

```
{
  "jingdong_smart_api_device_list_response": {
    "code": "0",
    "result":
    [{"code": "200", "data": [{"count": 1, "list": [{"device_name": "togic-w30", "id": "UUIA-13443-DFAACF", "p_description": "泰捷 WEBOX WE30 年度旗舰版采用 8 核 64 位处理器，超级双频双天线，扇翼散热，支持 4K，硬解 H.265，且采用高端游戏加速引擎。", "p_img_url": "http://img30.360buyimg.com/smartcloud/s200x200_jfs/t2086/45/122841343/9338/1a0a9109/55efeca5N87468a0c.jpg", "product_uuid": "NCN67S", "p_type": "空调", "status": "0"}]}], "errorMsg": "ok"}
  ]
}
```

返回结果解释：

```
count: 设备列表数量
device_name: 设备名称
id: 设备唯一 id
p_description: 产品描述
p_img_url: 产品图片地址
product_uuid: 产品唯一 uuid，用来区分产品
status: 0 不在线，1 在线
```

控制设备

功能：控制用户授权的设备

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=A7F100DEB192A79C9814B4FC
0F38C945&timestamp=2015-12-
```

```
28+11%3A49%3A13&v=2.0&app_key=A8064DFC0B834B7FA5A2F92B005FB27A&method=jingdong.smart.api.control&access_token=d757ad6e-2be1-4495-b11b-4bb3e3b9a206
```

HTTP POST BODY:

```
360buy_param_json={"command":["stream_id":"power","current_value":"1"],{"stream_id":"fan_speed","current_value":"2"}],"id":"UIA-13443-DFAACF"}
```

输入参数解释：

command: 功能参数名称及控制的值信息, 格式为 JSON 数组转换的数组字符串
id: 设备唯一 ID

HTTP RESPONSE:

```
{
  "jingdong_smart_api_control_response": {"code": "0", "result": {"code": "200", "errorMsg": "ok"}}
}
```

获得设备信息

功能：获取单个设备当前快照，返回当前设备的在线状态以及设备功能参数(stream)的最新数据

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?sign=44F19909AD06B9E81999FFF2A5F8C661&timestamp=2015-12-25+17%3A02%3A39&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.snapshot.get&access_token=d757ad6e-2be1-4495-b11b-4bb3e3b9a206
```

HTTP POST BODY:

```
360buy_param_json={"id":"UIA-13443-DFAACF","pull_mode":0}
```

输入参数解释：

id: 设备唯一 id
pull_mode: 1 从设备拉取快照, 0 从云端拉取。从设备拉取快照设备必须支持, 否则返回不支持错误提示。(可选)

HTTP RESPONSE:

```
{
  "jingdong_smart_api_snapshot_get_response": {
    "code": "0",
    "result": {"code": "200", "data": {
      "status": "1", "digest": "13234311", "streams": [
        {"current_value": "1", "stream_id": "power"}
      ]}, "errorMsg": "ok"}
  }
}
```

返回结果解释：

status: 设备是否在线, 1 表示在线, 0 表示不在线。
streams: 该设备支持的功能参数, 包括受控与不受控参数。
stream_id: 功能参数名

`current_value`: 功能参数当前值

获得批量设备信息

功能: 获取多个设备当前快照, 返回当前多个设备的在线状态以及设备功能参数(stream) 的最新数据

HTTPS 调用示例

https://smartopen.jd.com/routerjson?sign=44F19909AD06B9E81999FFF2A5F8C661×tamp=2015-12-25+17%3A02%3A39&v=2.0&app_key=F8D5EFE9A98F486AAC39EC1B401E1D73&method=jingdong.smart.api.snapshot.batch.get&access_token=d757ad6e-2be1-4495-b11b-4bb3e3b9a206

HTTP POST BODY:

```
360buy_param_json={"dev_ids":["152084097376714868","146787513680952321"],"pull_mode":0}
```

输入参数解释:

`devIds`: 设备唯一 id 的列表。
`pull_mode`: 1 从设备拉取快照, 0 从云端拉取。从设备拉取快照设备必须支持, 否则返回不支持错误提示。(可选)

HTTP RESPONSE:

```
{
  "jingdong_smart_api_snapshot_batch_get_response": {
    "code": "0",
    "result":
    "{ \"code\": \"200\", \"data\": { \"snapshots\": [ { \"streams\": [ { \"stream_id\": \"ptcheat\", \"current_value\": \"\" }, { \"stream_id\": \"mark speed\", \"current_value\": \"7\" } ], \"digest\": \"614335009\", \"query_code\": \"200\", \"id\": \"152084097376714868\", \"status\": \"0\" }, { \"streams\": [ { \"stream_id\": \"light\", \"current_value\": \"\" }, { \"stream_id\": \"power\", \"current_value\": \"\" } ], \"digest\": \"-1918413617\", \"query_code\": \"200\", \"id\": \"146787513680952321\", \"status\": \"0\" } ] }, \"errorMsg\": \"ok\" }"
  }
}
```

返回结果解释:

`code`: `result` 中的 `code` 表示批量获取多个设备的当前快照信息的请求是否全部成功, 200 表示全部成功, -1 表示全部失败
`snapshots`: 批量设备的当前快照信息。
`id`: 设备唯一 id。
`query_code`: 当前单个设备快照信息获取结果, 200 表示获取成功。
`status`: 设备是否在线, 1 表示在线, 0 表示不在线。
`streams`: 该设备支持的功能参数, 包括受控与不受控参数。
`stream_id`: 功能参数名
`current_value`: 功能参数当前值

获取卡片列表

- 请求方法: POST

- 请求体(JSON):

请求参数:

参数名	参数类型	必填	参数说明	取值
-----	------	----	------	----

请求示例:

卡片列表:

https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&sign=0AB28F861CFD4BD3EA8FD17901B5BB89×tamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.user.cards.list

请求 Body:

```
360buy_param_json={}
```

返回:

```
{
  "count": 2,
  "cache_time": "2018-08-06 09:34:25",
  "cards": [
    {
      "c_image_url": [
        {
          "name": "app_pic_url",
          "url": "https://img30.360buyimg.com/smartcloud/jfs/t20545/318/837345699/18660/1c0ce7c1/5b19ed5eNfeb066b3.png"
        },
        {
          "name": "app_click_url",
          "url": ""
        }
      ],
      "card_control": [
        {
          "stream_id": "power",
          "options": [
            {
              "0": "关"
            },
            {
              "1": "开"
            }
          ]
        }
      ],
      "card_desc": [ ],
      "card_name": "客厅长虹 3P 空调",
      "cid": 101001,
      "config_type": 1109,
      "create_time": 1530962886000,
      "feed_id": 153096299945635300,
      "puid": "AS2U4V",
    }
  ]
}
```

```

        "seq": 1,
        "snapshot": {
            "verdir": "0",
            "outdoortemp": "35",
            "hordir": "1",
            "timing_off": "0-00:00-01:34",
            "mark": "0",
            "airclean": "1",
            "voice_onoff": "1",
            "mode": "2",
            "volume": "51",
            "power": "0",
            "sleepmode": "0",
            "indoortemp": "29.400000",
            "ptcheat": "0",
            "settemp": "26",
            "timing_on": "0-00:00-01:34"
        },
        "status": 1,
        "stream_type_list": [ ],
        "version": "2.0"
    },
    {
        "c_image_url": [
            {
                "name": "app_pic_url",
                "url":
                "https://img30.360buyimg.com/smartcloud/jfs/t20545/318/837345699/18660/1c0ce7c1/5b19ed5eNfeb066b3.png"
            },
            {
                "name": "app_click_url",
                "url": ""
            }
        ],
        "card_control": [ ],
        "card_desc": [ ],
        "card_name": "海尔空调",
        "cid": 101001,
        "config_type": 1004,
        "create_time": 1499088713000,
        "feed_id": 149908871777138600,
        "puid": "D2AAGG",
        "seq": 2,
        "snapshot": {
            "horizontalWindMode": "fixed",
            "wind": "OFF",
            "selfCleaning": "OFF",
            "sleepCurve": "FALSE",
            "humanSenseMode": "FALSE",
            "beep": "TRUE",
            "indoorHumidityValue": "0",
            "indoorTemperatureValue": "24",
            "mode": "cool",
            "temperature": "23",
            "verticalWindMode": "fixed",
            "outdoorTemperatureValue": "65",
            "windSpeed": "auto",
    
```



```

        "humidity": "0",
        "forceRemoveStatus": "",
        "elock": "OFF",
        "airQualityStatus": "A",
        "humidify": "OFF",
        "electricHeating": "OFF",
        "power": "ON",
        "runningPowerValue": "3",
        "anion": "ON"
    },
    "status": 1,
    "stream_type_list": [ ],
    "version": "2.0"
}
]
}

```

返回参数:

参数名	参数类型	必填	参数说明	取值
app_pic_url	string	Y	卡片图片	
card_control	array	Y	卡片控制快捷方式	
card_name	string	Y	卡片名称	
status	int	Y	是否在线	1 在线, 0 不在线
snapshot	json	Y	设备快照	
feed_id	long	Y	设备 id	
cid	int	Y	三级类目 id	
puid	string	Y	产品 uuid	

获取场景列表

- 请求方法: POST
- 请求体(JSON):

请求参数:

参数名	参数类型	必填	参数说明	取值
-----	------	----	------	----

请求示例:

场景列表:

https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&sign=0AB28F861CFD4BD3EA8FD17901B5BB89×tamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.user.scenes.list

请求 Body:

```
360buy_param_json={}
```

返回:

```
[
  {
    "status": 0,
    "name": "开空净"
  },
  {
    "timers": [ ],
    "device_count": 1,
    "status": 1,
    "name": "起床 01",
    "enable": true,
    "scenario_id": "scenario1530290823423869227224895",
    "scene_type": 0,
    "script_id": "B81B6A445A03F2FC008C3DCFFEF9AC99"
  },
  {
    "timers": [
      {
        "status": 1,
        "next": "2018-08-06 15:00:00"
      },
      {
        "status": 1,
        "next": "2018-08-06 15:01:00"
      },
      {
        "status": 1,
        "next": "2018-08-06 15:02:00"
      },
      {
        "status": 1,
        "next": "2018-08-06 15:03:00"
      },
      {
        "status": 1,
        "next": "2018-08-06 15:04:00"
      },
      {
        "status": 1,
        "next": "2018-08-06 15:05:00"
      },
      {
        "status": 1,
        "next": "2018-08-06 15:06:00"
      }
    ],
    "device_count": 1,
    "status": 1,
    "name": "睡觉模式",
    "enable": true,
    "scenario_id": "scenario1530262586523908788977879",
    "scene_type": 0,
    "script_id": "8C2761FDCAC7A5A245737C00958B83344"
  }
]
```

返回参数:

参数名	参数类型	必填	参数说明	取值
device_count	int	N	场景设备数	
status	int	Y	场景是否有效	0 无效, 1 有效
name	string	Y	场景名称	
enable	boolean	N	场景是否开启	false 未开启, true 开启
scenario_id	string	N	场景执行 Id	
script_id	String	N	场景脚本	
scene_type	int	N	场景类型	0 自定义场景 1 睡觉模式 2 起床模式 3 回家模式 4 离家模式

执行场景

- 请求方法: POST
- 请求体(JSON):

请求参数:

参数名	参数类型	必填	参数说明	取值
scene_id	string	Y	场景执行 Id	

请求示例:

```
https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.scene.execute
```

请求 Body:

```
{
  "scene_id": "scenario1530262586523908788977879"
}
```

返回:

返回参数:

参数名	参数类型	必填	参数说明	取值
-----	------	----	------	----

获取场景详情

- 请求方法: POST

- 请求体(JSON):

请求参数:

参数名	参数类型	必填	参数说明	取值
script_id	string	Y	场景脚本 id	

请求示例:

场景详情:

```
https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.scene.detail.get
```

请求 Body:

```
{
  "360buy_param_json": {
    "script_id": "8C2761FDCAC7A5A245737C00958B83344"
  }
}
```

返回:

```
{
  "enable": true,
  "actions": [
    {
      "name": "长虹空调-挂机",
      "action": "送风",
      "delay": 660000,
      "pic": "https://img30.360buyimg.com/smartcloud/jfs/t20545/318/837345699/18660/1c0ce7c1/5b19ed5eNfeb066b3.png"
    }
  ]
}
```

返回参数:

参数名	参数类型	必填	参数说明	取值
name	string	Y	设备名称	
action	string	Y	设备执行内容	
delay	long	Y	是否延时执行(单位毫秒)	
pic	string	Y	设备图片	

获取用户详情

- 请求方法: POST

- 请求体(JSON):

请求参数:

参数名	参数类型	必填	参数说明	取值
-----	------	----	------	----

请求示例:

```
https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.user.detail.get
```

请求 Body:

```
360buy_param_json={"script_id":"8C2761FDCAC7A5A245737C00958B83344"}
```

返回:

```
{
  "photo_url":
  "https://img30.360buyimg.com/smartcloud/jfs/t20545/318/837345699/18660/1c0ce7c1/5b19ed5eNfeb066b3.png",
  "level": "51",
  "nickname": "testname",
}
```

返回参数:

参数名	参数类型	必填	参数说明	取值
nickname	string	Y	用户昵称	
photo_url	string	Y	用户头像	
level	string	Y	用户等级	

6

数据推送服务 OpenAPI

6.1 概述

第三方参与的整体流程说明:

1. 第三方需要搭建 **http service** 来接受推送（要求使用 **https** 协议）。同时提供一个固定的外网可访问的 **URL** 给京东智能服务进行配置，之后京东智能服务平台会将数据主动 **POST** 该 **URL**。

2. 第三方如果需要相应的消息通知在智能云开放平台申请订阅的相应消息通知，包括：消息类型，回调地址。（PS：目前由开放平台工作人员提供申请）
3. 智能云开放平台对申请进行审核
4. 审核通过后，第三方实现回调 URL 以便接收通知消息。回调的请求返回的信息类型及格式请参见下文。

6.2 数据推送订阅

按类型订阅数据

功能：订阅需要推送的数据类型。

方法名：jingdong.smart.api.datapush.sub

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.datapush.sub
```

HTTP POST BODY:

订阅设备消息:

```
360buy_param_json={"sub_keys":["138213782131231","12932132332323"],"sub_type":"FEED_ID","msg_type": "device.status"}
```

订阅场景消息:

```
360buy_param_json={"sub_keys":["995F16591EF3570553FAB1E097A49ACA10000003","1639232F4B03B0B1F9A1747FF745FB18"],"sub_type":"SCENE_ID","msg_type": "scene.msg"}
```

请求参数解释：

sub_keys: 订阅主题的列表，比如设备 feed_id 列表和场景 id 列表
sub_type: 订阅的主题类型，支持 FEED_ID, SCENE_ID
msg_type: 订阅的消息类型，支持 device.status, scene.msg 订阅设备数据快照，包括设备属性最新值、上线和离线状态

HTTP RESPONSE:

```
{
  "jingdong_smart_api_datapush_sub_response": {
    "code": "0",
    "result": "{\\"code\\":200,\\"errorMsg\\":\\"OK\\"}"
  }
}
```

按用户订阅数据

功能：订阅需要推送的数据类型。

方法名：jingdong.smart.api.datapush.user

是否需要授权：需要进行授权

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.datapush.user
```

HTTP POST BODY:

订阅设备消息:

```
360buy_param_json={"msg_type": "user.msg"}
```

请求参数解释：

user.msg:包括已有的

设备的消息：device. status device.bind device.unbind device.rename

场景的消息：updateScene addScene delScene scene.msg

msg_type: 订阅的消息类型,支持 订阅设备用户新增设备（新添加或共享）、设备快照更新、设备离线或设备名称变更时推送信息,包括设备属性最新值
当场景新增、删除或场景变更时推送消息。

HTTP RESPONSE:

```
{
  "jingdong_smart_api_datapush_user_response": {
    "code": "0",
    "result": "{\"code\":200,\"errorMsg\":\"OK\"}"
  }
}
```

取消订阅

功能：订阅需要推送的数据类型。

方法名：jingdong.smart.api.datapush.unsub

HTTPS 调用示例

```
https://smartopen.jd.com/routerjson?access_token=12345678-b0e1-4d0c-9d10-a998d9597d75&sign=0AB28F861CFD4BD3EA8FD17901B5BB89&timestamp=2015-12-25+16%3A57%3A02&v=2.0&app_key=AB064DFC0B834B7FA5A2F92B005FB37A&method=jingdong.smart.api.datapush.unsub
```

HTTP POST BODY:

退订设备消息:

```
360buy_param_json={"sub_keys":
["138213782131231","12932132332323"],"sub_type":"FEED_ID",
"msg_type": "device.status"}
```

退订场景消息:

```
360buy_param_json={"sub_keys":
["995F16591EF3570553FAB1E097A49ACA10000003","1639232F4B03B0B1F9A1
747FF745FB18"],"sub_type":"SCENE_ID",
"msg_type": "scene.msg"}
```

退订用户消息:

```
360buy_param_json={"sub_keys": null,"sub_type":null,
"msg_type": "user.msg"}
```

请求参数解释:

sub_keys: 取消订阅主题的列表, 比如设备 feed_id 列表
sub_type: 取消订阅的主题类型, 支持 FEED_ID
msg_type: 取消订阅的消息类型, 支持 device.status, 订阅设备数据快照, 包括设备属性最新值、上线和离线状态

HTTP RESPONSE:

```
{
  "jingdong_smart_api_datapush_unsub_response": {
    "code": "0",
    "result": "{\"code\":200,\"errorMsg\":\"OK\"}"
  }
}
```

6.3 数据推送规范

数据签名算法

回调第三方接口, 数据签名算法支持以下几种类型。签名算法默认为 MD5, 若要调整需要在云端进行配置。

序号	签名算法
1	MD5
2	SHA-1
3	SHA-256
4	HMAC-SHA-1
5	HMAC-SHA-256

回调第三方的数据返回格式

在回调第三方后, 第三方需要返回统一的数据格式标识请求成功或失败。返回的格式如下:

字段名	类型	必填	描述
code	Integer	Y	标识请求结果。为 0 时, 表示成功, 其它失败
message	String	Y	请求结果说明
desc	String	N	请求结果的详细描述

成功示例:

```
{
  "code":0,
  "message":"ok",
  "desc": "成功调用"
```



```

}
失败示例:
{
    "code":1,
    "message":"签名验证错误",
    "desc": "消息签名错误."
}
    
```

回调第三方返回的常见错误及说明

错误码	描述
-1	系统错误
1	请求参数缺失
2	AppKey 不匹配
3	请求时间戳不匹配
4	不支持的 method
5	请求数据格式错误
6	签名验证错误
7	请求数据错误

6.4 数据推送接口

设备数据快照推送

功能：推送设备的数据快照给第三方

HTTPS 调用示例

```

https://domain/path?timestamp=2017-07-
05+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7
E75C0AA9E798E61&method=device.status
    
```

HTTP POST BODY:

```

360buy_param_json=[
{"user_id":"jack","feed_id":"121231231231","status":"1","digest":"13234311","streams"
: [{"current_value": "1","stream_id": "power"}]},
{"feed_id":"135723123131","status":"1","digest":"2344578","streams":
[{"current_value": "1","stream_id": "switch"}]}
]
    
```

具体含义参考《微联智能云设备访问服务 API》中“获得设备信息”章节的返回数据。

HTTP RESPONSE:

```

{
    "code":0,
    "message":"ok",
    "desc": "成功调用"
}
    
```

设备重命名推送

功能：推送设备的数据重命名给第三方

HTTPS 调用示例

```
https://domain/path?timestamp=2017-07-05+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=device.rename
```

HTTP POST BODY:

```
360buy_param_json=[
{"user_id":"jack","feed_id":"121231231231","device_name":"设备名字"},
具体含义参考《微联智能云设备访问服务 API》中“获得设备信息”章节的返回数据。
```

HTTP RESPONSE:

```
{
  "code":0,
  "message":"ok",
  "desc": "成功调用"
}
```

设备绑定推送

功能：推送用户绑定的设备消息给第三方

HTTPS 调用示例

```
https://domain/path?timestamp=2018-09-28+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=device.bind
```

HTTP POST BODY:

```
360buy_param_json=
{"user_id":"jack","feed_id":"121231231231","device_name":"设备名字"}
```

HTTP RESPONSE:

```
{
  "code":0,
  "message":"ok",
  "desc": "成功调用"
}
```

设备分享推送

功能：推送设备的分享消息给第三方

HTTPS 调用示例

```
https://domain/path?timestamp=2018-09-28+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=device.share
```

HTTP POST BODY:

```
360buy_param_json=
{"user_id":"jack","feed_id":"121231231231","device_name":"设备名字"}
```

HTTP RESPONSE:

```
{
  "code":0,
  "message":"ok",
  "desc": "成功调用"
}
```

```
}

```

设备解绑推送（关联设备数据快照推送）

功能：推送设备的绑定或者解绑消息给第三方

说明：设备解绑推送的前提是**该设备的快照被订阅**，设备快照未被订阅的设备不对第三方推送设备解绑消息

HTTPS 调用示例

```
https://domain/path?timestamp=2018-09-28+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=device.unbind
```

HTTP POST BODY:

```
360buy_param_json=
{"feed_id":"121231231231"}
```

HTTP RESPONSE:

```
{
  "code":0,
  "message":"ok",
  "desc": "成功调用"
}
```

设备解除分享推送

功能：推送设备的解除分享消息给第三方

HTTPS 调用示例

```
https://domain/path?timestamp=2018-09-28+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=device.unshare
```

HTTP POST BODY:

```
360buy_param_json=
{"user_id":"jack","feed_id":"121231231231"}
```

HTTP RESPONSE:

```
{
  "code":0,
  "message":"ok",
  "desc": "成功调用"
}
```

设备解绑子设备推送

功能：推送设备的解绑子设备消息给第三方

HTTPS 调用示例

```
https://domain/path?timestamp=2018-09-28+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=device.unbindsub
```

HTTP POST BODY:

```
360buy_param_json=
{"user_id":"jack","feed_id":"121231231231"}
```

HTTP RESPONSE:

```
{
  "code":0,
```

```

        "message": "ok",
        "desc": "成功调用"
    }

```

场景新增消息推送

功能：推送场景的新增给第三方

HTTPS 调用示例

```

https://domain/path?timestamp=2017-07-05+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=scene.addScene

```

HTTP POST BODY:

```

360buy_param_json=
{"user_id": "jack", "scenario_id": "场景 id", "scene_name": "场景名字", "script_id": "995F16591EF3570553FAB1E097A49ACA"}。

```

HTTP RESPONSE:

```

{
    "code": 0,
    "message": "ok",
    "desc": "成功调用"
}

```

场景删除消息推送

功能：推送场景的删除快照给第三方

HTTPS 调用示例

```

https://domain/path?timestamp=2017-07-05+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=scene.delScene

```

HTTP POST BODY:

```

360buy_param_json=
{"user_id": "jack", "scenario_id": "场景 id", "script_id": "995F16591EF3570553FAB1E097A49ACA", "scene_name": "场景名字", }。

```

HTTP RESPONSE:

```

{
    "code": 0,
    "message": "ok",
    "desc": "成功调用"
}

```

场景变更消息推送

功能：推送场景的变更消息给第三方

HTTPS 调用示例

```

https://domain/path?timestamp=2017-07-05+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7E75C0AA9E798E61&method=scene.updateScene

```

HTTP POST BODY:

```
360buy_param_json=
{"user_id":"jack","scenario_id":"场景 id","script_id":
"995F16591EF3570553FAB1E097A49ACA","scene_name","场景名字"}。
```

HTTP RESPONSE:

```
{
  "code":0,
  "message":"ok",
  "desc": "成功调用"
}
```

场景执行消息推送

功能：推送设备的数据快照给第三方

HTTPS 调用示例

```
https://domain/path?timestamp=2017-07-
05+15%3A06%3A23&sign=1467EFB7AC7FA8AB1469E0A29AA84CF4&v=1.0&app_key=E6DDDC9236D482EE7
E75C0AA9E798E61&method=scene.msg
```

HTTP POST BODY:

```
360buy_param_json=
{"user_id":"jack","msg_content":"回家场景有 1 台设备执行失败
","record_id":"1DEBDWEFGE","trigger_type":"1","scene_result":"1","script_id":
"995F16591EF3570553FAB1E097A49ACA","logic_id": "10000001"},}。
```

Trigger_type:1 手动触发、2 定时触发、3 设备触发

Scene_result:0 全部成功、1 全部失败、2 部分失败

Record_id:场景执行记录 id

HTTP RESPONSE:

```
{
  "code":0,
  "message":"ok",
  "desc": "成功调用"
}
```

7

设备访问服务 SDK

请参看京东开放服务 SCMSDK Android/iOS
中说明文档与 Demo

<http://smartdev.jd.com/docCenterDownload/list/2>

开放服务-京东开放服务 SCMSDK 安卓版/iOS
版