

## **bind**

### **相关模块**

- device
  - unbindDevice
  - isDeviceBoundByUser
  - getUserDevice
- product
  - createDevicePropertyTable
  - addDeviceProperty
  - modifyDeviceProperty
  - deleteDeviceProperty

## **client**

- product
- timerTask
- warehouse
- dstore
- gateway
- platform
- account
- deviceService

## **access point**

- getAccessPoint
  - 通过逻辑ID获取设备信息
  - 如果是用户检查设备是否绑定
    - 检查是否在本地图存(分区)
      - 有缓存且已过期或者无缓存记录则更新缓存,

- 没有使用缓存直接走数据库
- 返回新的accessPoint

## 设备在线状态

- isDeviceOnline
  - 开发者使用物理ID查询,由gateway客户端发送online请求
  - 非开发者使用物理ID查询,通过物理ID获取设备信息(主要是逻辑ID),而后使用gateway获取设备在线状态
  - 非开发者通过逻辑ID查询设备物理ID,而后使用gateway获取在线状态
- sendToDevice
  - 若传入的是物理ID则换成逻辑ID,
  - 不是传入的物理ID,获取accessDevice,检查设备,若是用户检查是否绑定
  - gateway发送control指令
- openGatewayMatch
  - 通过传入的设备逻辑ID查询子设备信息,检查子设备gateway逻辑ID是否与传入的逻辑ID一致,若是开发者不检查子设备的所有者是否是当前用户的UID
  - 通过设备逻辑ID获取子设备信息,检查子设备的gatewayID是否与设备逻辑ID相等,若是用户检查子设备的所有者是否是当前用户
  - 通过逻辑ID获取设备信息,取得物理ID
  - 使用device rpc client发送openGatewayMatch
- closeGatewayMatch
  - 通过传入的逻辑ID获取子设备信息
  - 若非开发者判断用户是否是所有者
  - 通过设备逻辑ID获取设备的物理ID
  - 通过gateway客户端发送closeGatewayMatch的请求
- evictSubDevice
  - 通过逻辑ID获取子设备信息
  - 非开发者检查是否是子设备所有者
  - 通过设备逻辑ID获取设备信息,物理ID
  - 通过device service rpc client发送evictSubDevice

## 检查设备绑定状态

- idDeviceBound
  - 通过物理ID获取设备

## 定时任务

- addTimerTask
  - 通过逻辑ID获取设备信息,物理ID
  - 通过逻辑ID获取子设备信息
  - 如果子设备的gatewayDid等于逻辑ID,往device service rpc client发送一个addTimerTask请求
  - 否则先获取设备的物理ID,而后往device service发送addTimerTask请求
- deleteTimerTask
  - 通过逻辑ID获取设备信息,物理ID
  - 通过逻辑ID获取子设备信息
  - 如果子设备的gatewayDid等于逻辑ID,往device service rpc client发送一个deleteTimerTask请求
  - 否则先获取设备的物理ID,而后往device service发送deleteTimerTask请求
- cleanDeviceTasks
  - 删除所有的定时任务,子设备的子域,设备ID传空

## 设备管理

- isDeviceBoundBuyUser
  - 传入逻辑ID则先获取物理ID
  - 使用物理ID检查设备是否已经绑定
- listDevices
  - 列出某个用户的所有设备
  - 对每一个设备进行遍历
    - 通过逻辑ID获取设备信息
    - 获取设备绑定信息
    - 通过设备的子域ID获取其子域信息

- 如果需要设备必须在线
  - 如果设备的gatewayId等于设备逻辑ID则直接检查设备的在线状态
  - 否则设备是子设备,通过gatewayId获取主设备信息,通过gateway查询子设备的在线状态
- listDevicesExt
  - getDevicesOfUserExt获取用户绑定的设备的基础信息及其绑定时间、扩展属性、故障相关信息等。
  - subDomainId可以为空字符串,表示不区分设备子域;不为空时表示只查询该子域下的设备。
  - 列出某个用户的所有设备,从bind\_info和device\_info两个表获取数据
  - 对每一个设备进行遍历
    - 通过逻辑ID获取设备信息
    - 获取设备绑定信息
    - 通过设备的子域ID获取其子域信息
    - 如果需要设备必须在线
      - 如果设备的gatewayId等于设备逻辑ID则直接检查设备的在线状态
      - 否则设备是子设备,通过gatewayId获取主设备信息,通过gateway查询子设备的在线状态
    - 从绑定信息中获取设备绑定时间
    - 获取该设备绑定的所有用户
    - 通过设备获取所有产品属性
      - product rpc client发送listProductAttribute
      - dstore获取设备状态
    - 获取设备扩展信息,字段不确定性
    - 获取设备故障状态,通过device service rpc发送getDeviceFaultStatus
- listUsers
  - 获取子设备信息,先从分区缓存中获取
  - 获取某个设备绑定的所有用户
  - 非开发者则检查用户是否绑定这个设备
  - 获取所有用户信息
  - 获取所有用户profile信息

- 对每个用户
  - 为用户设置profile
  - 增加绑定时间
- getDeviceCount
  - 获取主域或者某个子域的设备数量
- getAllDevices
  - 获取所有设备
- getDeviceCountOfUser
  - 获取用户绑定的设备数量
  - 优先取请求body里面的参数
- getDevicesOfUser
  - 获取用户的所有设备
  - 优先去请求body里面的数据
- getDevicesOfUserExt
  - getDevicesOfUserExt获取用户绑定的设备的基础信息及其绑定时间、扩展属性、故障相关信息等。
  - subDomainId可以为空字符串，表示不区分设备子域；不为空时表示只查询该子域下的设备。
  - 获取用户的所有设备
  - 获取每个设备的绑定信息,子域名称,在线状态,绑定时间,扩展属性,扩展信息,故障相关数据等
- bindDevice
  - 绑定设备
    - 通过物理ID获取设备绑定信息
    - 如果不存在设备绑定信息
      - 非开发者非蓝牙设备且非强制绑定
      - 检查设备是否在线
      - 绑定设备
        - 往bind\_info里面添加记录,先看缓存,后网device\_sub\_info里面添加数据,绑定的用户设置为管理员
      - 当开发者绑定时往warehouse里面注册一个设备,registerDevice
      - 非开发者非蓝牙设备且非强制绑定,换token
    - 设备曾经被绑定过
      - 检查gatewayId是否和逻辑ID一致

- 检查用户是否是设备所有者
- 检查当前用户以及设备逻辑ID是否绑定
- 设备被非当前用户绑定,且非强制绑定,返回已经被别人绑定
- 绑定用户和设备
- 往设备扩展属性设置设备位置
- 记录设备绑定事件
- bindDeviceWithoutUser
  - 通过物理ID获取设备信息
  - 不存在设备信息,即没有被绑定过
    - 非开发者检查设备是否在线,设置设备状态为预绑定
    - 绑定设备,ownerID为0,rid为0
    - 当开发者绑定时往warehouse里面注册一个设备,registerDevice
    - 非开发者换token
  - 设备曾经被绑定过
    - 检查gatewayId是否和逻辑ID一致
    - 检查用户是否是设备所有者
    - 检查当前用户以及设备逻辑ID是否绑定
    - 绑定用户和设备,rid,role为0
  - 往设备扩展属性设置设备位置
- unbindDevice
  - 若是用户,通过逻辑ID获取物理ID
  - 开发者或者内部服务调用
    - 通过物理ID获取设备
    - 设备owner可能为0,可能是不带用户的绑定设备
  - 如果用户则获取设备绑定信息,判断rid是否有效,有效返回错误
  - 解绑设备
    - 设备是网关设备获取所有子设备
    - 非网关设备则将该设备设置为第一个设备
    - 若用户是设备的所有者则获取该设备的所有用户
    - 否则将其设为第一个用户
    - 遍历设备
      - 判断设备逻辑ID与遍历的逻辑ID是否不一样,不一样则获取设备的逻辑ID(对网关设备)

- 检查用户对该设备的角色,判断是用户还是所有者
  - 解绑设备
    - 非所有者,删除绑定信息,
    - 如果设备绑定时没有拥有者,且是最后一个删除设备的用户,则删除设备信息
    - 是拥有者则删除设备
    - 清除设备的任务
- 看设备是否需要反激活
  - 当前用户不是设备所有者,则以所有者的身份去解绑设备
  - 在warehouse反激活设备
- changeOwner
  - 获取设备组信息,设备在组里面则返回error,设备逻辑ID必须是网关设备
  - 通过设备逻辑ID获取子设备信息
  - 判断设备所有者是否是当前请求所带的用户ID
  - 判断设备所有者是否是请求body所带的用户ID,是则直接返回
  - 检查传入的uid是否与设备绑定
  - 更新所有的子设备所有者
- changeDevice
  - 通过逻辑ID获取设备,传入的物理ID不能是设备正在使用中的设备(当前传入的逻辑ID)
  - 获取子设备,子设备拥有者非当前用户,返回错误
  - 非开发者
    - 子设备是网关则更换token
    - 仅是子设备则先获取网关设备
    - 检查网关设备是否在线
  - 通过物理ID获取设备信息
  - 要绑定的设备已激活则删除子设备信息
  - 更新设备的物理ID
- modifyDevice
  - 获取设备及子设备信息
  - 非开发者则检查用户是否是其拥有者(京东接口)
  - 更新设备名
- getShareCode
  - 通过逻辑ID获取子设备,检查用户是否是其拥有者

- 获取共享码,如果没有则新生成一个,共享码已经过期则更新其有效时间
- fetchDeviceShareCode
  - 获取子设备,检查当前用户是否是管理员
  - 取存在的二维码检查是否过期
  - 不存在则新建设备二维码
- refreshDeviceShareCode
  - 通过逻辑ID取子设备信息,检查是否是管理员
  - 生成新的二维码,如果旧的存在则替换
  - 返回新的二维码
- bindDeviceWithShareCode
  - 检查是否是京东发起的内部服务
  - 不是则切分二维码url,获取设备逻辑ID
  - 通过逻辑ID获取设备信息,检查是否是管理员
  - 非内部服务则检查二维码是否有效
  - 检查设备是否在组
  - 如果没有绑定过
    - 非网关设备则绑定设备
    - 网关设备则绑定所有子设备
- bindDeviceWithUser
  - 通过传入的用户id或者用户名获取用户数据
  - 检查请求的用户或者请求body里面的用户是否是管理员,前者必须是管理员,不允许普通用户绑定其它设备
  - 未绑定则绑定设备,非网关则直接绑定,网关则依次绑定子设备
- unbindDeviceWithUser
  - 获取设备
  - 检查当前用户是否是设备管理员,传入的用户id不能说管理员id
  - 检查用户与设备的绑定信息,设备和用户不能在同一个组
  - 解绑设备
- updateAccessKey
  - 通过逻辑ID获取设备,子设备
  - 检查设备是否已经激活
  - 已绑定,非活跃,检查是否在线
  - 生成token,由device service 更换token
  - 更新本地token



- bindGateway
  - 绑定网关设备
  - 设置缓存
- unbindGateway
  - 既是用户又是开发者,且组id大于0,先解绑组
  - 解绑设备
  - 需要反激活设备则反激活设备
- addSubDevice
  - 检查gateway信息,检查用户是否是其管理员
  - 检查子设备是否已存在
  - 非开发者,检查设备是否在线
  - 绑定子设备
  - 绑定子设备组
  - 设置设备位置
- deleteSubDevice
- listNewDevices
  - 列出未绑定的子设备
- setDeviceProfile
  - 设置设备扩展属性

分区用于处理缓存,bind有些接口有缓存,设备下次访问需要重新向原来的机器发请求.

激活设备,APP绑定设备连云端,设备必须在线  
 授权,开发者注册设备,不一定已授权,也不一定练过云端  
 设备导入但是不一定连接云端