Android API接口说明

1. APP用户登录接口
   1. 账号密码登录。
      1. 登录信息参数示例：  
         {  
          ‘loginName’:’loginName’,//账号  
          ’password’:’pwd’,//密码  
          ’phoneInfo’:{//手机设备信息  
          ’imei’:’’,//手机唯一标识  
          ’brand’:’’,//手机品牌  
          ’phoneModel’:’’,//手机型号  
          ’manufacture’:’’,//生产厂家  
          ’androidVersion’:’’//Android版本  
          }  
         }
      2. 登录接口描述  
         1）登录接口地址：/iot-seal-app-api/appLogin  
         2）接口完成功能：  
          ①账号密码验证登录（sys\_user）  
          ②账号手机绑定验证，是否跟上次登录保持一致，如果不一致进行短信验证，重新进行账号手机绑定。第一次登录默认绑定手机。  
          ③判断该账号是否为第一次登录，如果为第一次登录，则强制修改密码。（根据最后一次登录时间进行判断是否为第一次登录）  
          ④登录设备及账号信息保存（basics\_phone\_info）。  
          ⑤更新账号的登录时间（sys\_user：login\_date）。  
         3）返回信息：  
         {  
          ‘code’:’000000’,//返回结果状态值  
          ‘msg’:’’, //返回结果信息  
          ‘data’:{  
          userId:’’,//账号ID  
          loginName:’’,//账号所属人  
          loginName:’’,//账号  
          phone:’’,//账号绑定手机，中间四位以\*代替  
          appId:’’,//设备注册ID  
          }   
         }  
         000000：登录成功；100001：账号或密码错误；100002：登录设备发生改变；100003：用户第一次登陆；
   2. 设备变更短信认证接口
      1. 参数示例：  
         {  
          ‘loginName’:’’,//登录账号  
          ’phoneNum’:’’,//账号预留的手机号  
          ’codeNum’:’’//验证码  
         }
      2. 接口地址：/iot-seal-app-api/authChangeDeviceMsg
      3. 接口规则：  
         1）手机号为账号预留手机号；  
         2）在没有短信验证码时，为发送短信验证码，有验证码则验证验证码；  
         3）短信验证码为6为数字；  
         4）短信验证码存在redis数据库中，key为账号+“设备变更”+手机号，保存时间为10分钟；  
         5）同一手机号同一类型1分钟内不能重复发送验证码。
      4. 返回信息：  
         {  
          ‘code’:’000000’,//返回结果状态值  
          ‘msg’:’’, //返回结果信息  
          ‘data’:{}   
         }  
         000000：‘发送成功/验证成功’；100001：‘发送失败，原因（短信网关返回信息）’；100002：‘短信验证码失效，请重新验证’；100004：‘验证码错误’；100500：‘短信发送失败，原因（系统异常）’。
   3. 密码修改接口
      1. 参数示例：  
         {  
          ‘loginName’:’’,//账号  
          ‘curPwd’:’’,//旧密码  
          ’newPwd’:’’//新密码  
         }
      2. 接口地址：/iot-seal-app-api/updatePwd
      3. 接口规则：  
         1）验证旧密码是否正确；  
         2）按照密码生成规则，根据新密码重新加密；  
         3）系统日志记录。
      4. 返回信息：  
         {  
          ‘code’:’000000’,//返回结果状态值  
          ‘msg’:’’, //返回结果信息  
          ‘data’:{}   
         }  
         000000：’密码修改成功’；100001：’原密码错误’；100500：’密码修改失败，原因：’。
   4. 用户基本信息
      1. 参数示例：{‘loginName’:’’}
      2. 接口地址：/iot-seal-app-api/getUserInfo
      3. 接口规则：根据账号查询用户信息，返回用户实体信息。
      4. 返回信息：  
         {  
          ‘code’:’’,//状态码  
          ’msg’:’’,//状态描述信息  
          ’data’:’’//封装用户信息  
         }  
         000000：请求成功；100404：未找到该账号信息；100500：系统异常，原因（系统内部报错信息）；
   5. 当前货单列表
      1. 参数示例：{‘loginName’:’’}
      2. 接口地址：/iot-seal-app-api/findCurTaskListByUser
      3. 规则说明：  
         1）查询内容包括货单号、货单状态、发货人、发货时间、发货地点、货物信息、收货单位、收货人、施封时间、验封时间、验封地址、是否告警。  
         2）在当前任务单表中进行查询，关联设备的校验表；  
         3）查询账号关联的所属车辆的箱体的设备的任务；  
         4）查询账号关联的箱体的设备的任务；  
         5）查询账号关联的设备的任务；  
         6）查询设备的验证信息获取施封时间、验封时间、验封地址。  
         7）根据任务告警表判断任务是否告警。
      4. 返回信息：  
         {  
          ‘code’:’’,//状态码  
          ’msg’:’’,//状态描述  
          ’data’:[  
          {  
          taskId:’’,//货单ID  
          taskNumber:’’,//货单号  
          state:’’,//货单状态  
          consigner:’’,//发货人  
          deliverDatetime:’’,//发货时间  
          deliverAddress:’’,//发货地点  
          taskInfo:’’,//货物信息  
          consignee:’’,//收货人  
          receivingCompany:’’,//收货单位  
          sealTime:’’,//施封时间  
          verifyTime:’’,//验封时间  
          isAlarm:’’,//是否告警  
          verifyLocation:’’//验封地址  
          }  
          ]  
         }  
         000000：请求成功；100500：数据异常。
   6. 获取车辆信息
      1. 参数示例：{loginName:’’}
      2. 接口地址：/iot-seal-app-api/findCarByUser
      3. 接口规则：  
         1）查询内容包括车辆ID、车牌号、车主姓名、联系方式、所属部门、车辆状态。  
         2）根据车辆账号关系表查询账号的车辆列表信息。  
         3）根据车辆最后一个货单状态判断车辆状态。  
         4）根据车辆的管理账户判断车辆所属部门。
      4. 返回信息：  
         {  
          code:’’,  
          msg:’’,  
          data:[  
          {  
          carId:’’,//车辆ID  
          carNumber:’’,//车牌号  
          vehicleOwnerName:’’,//车主姓名  
          phoneNum:’’,//联系方式  
          departName:’’,//所属部门  
          carState:’’//车辆状态  
          }  
          ]  
         }  
         000000：请求成功；100500：数据异常。
   7. 获取箱体信息。
      1. 参数示例：{loginName:’’}
      2. 接口地址：/iot-seal-app-api/findContainerByUser
      3. 接口规则：  
         1）查询内容包括箱体ID，箱体编号、箱门数量、箱体类型、箱体状态、所属部门、绑定车辆。  
         2）根据账号箱体关系表查询箱体列表。  
         3）箱体所关联最新的货单状态为箱体状态。  
         4）箱体管理人员所在部门为所属部门。  
         5）从箱体车辆关系表中查询箱体锁绑定的车辆。
      4. 返回信息  
         {  
          code:’’,  
          msg:’’,  
          data:[{  
          containerId:’’,//箱体ID  
          containerNumber:’’,//箱体编号  
          doorCount:’’,//箱门数量  
          containerType:’’,//箱体类型  
          departName:’’,//所属部门  
          containerState:’’,//箱体状态  
          bindCarNumber:’’//绑定车辆编号  
          }]  
         }  
         000000：请求成功；100500：数据异常
   8. 获取设备信息。
      1. 参数示例：{loginName:’’}
      2. 接口地址：/iot-seal-app-api/findDeviceByUser
      3. 接口规则：  
         1）查询内容包括设备ID，设备编号、设备类型、用户权限、设备状态、所属部门、绑定箱体、校验时间。  
         2）根据账号设备关系表查询设备列表。  
         3）根据设备箱体绑定关系表查询设备绑定箱体号。
      4. 返回信息  
         {  
          code:’’,  
          msg:’’,  
          data:[{  
          deviceId:’’,//设备ID  
          deviceNumber:’’,//设备编号  
          deviceType:’’,//设备类型  
          departName:’’,//所属部门  
          deviceState:’’,//设备状态  
          privilege:’’,//用户权限  
          bindContainerNumber:’’//绑定箱体编号  
          }]  
         }  
         000000：请求成功；100500：数据异常
2. 货单号生成接口
   1. 定义货单号生成规则：  
      1）年月日时分秒+第一个验证设备+5位序号。物联网封为00，物联网锁为01。  
      2）序号存储在Redis中。  
      3）该接口同一时间只能被一个进行访问。
   2. 货单号生成接口地址：/iot-seal-app-api/generateTaskNumber
   3. 返回信息：  
      {  
       code:’’,  
       msg:’’,  
       data:{  
       taskNumber:’’,//货单号  
       }  
      }  
      000000：请求成功；100500：数据异常
3. 货单保存接口
   1. 参数示例：  
      {  
       taskNumber:’’,//货单号  
       lockTime:’’,//施封时间  
       deliverDatetime:’’,//发货时间  
       deliverAddress:’’,//发货地点  
       deliverLongitude:’’,//发货经度  
       deliverLatitude:’’,//发货纬度  
       state:’’,//货单状态  
       isAlarm:’’,//是否告警  
       carList:[{//运输车辆列表  
       carId:’’,//车辆ID  
       carNumber:’’,//车牌号  
       containerList:[{//箱体列表  
       containerId:’’,//箱体ID  
       containerNumber:’’,//箱体编号  
       deviceList:[{//设备列表  
       deviceId:’’,//设备ID  
       deviceType:’’,//设备类型  
       deviceNumber:’’,//设备编号  
       deviceState:’’,//设备状态  
       deviceStateTime:’’,//设备状态更新时间  
       longitude:’’,//设备状态更新时经度  
       latitude:’’,//设备状态更新时纬度  
       deviceLatestLocation:’’//设备状态更新时地址  
       }],  
       }],  
       }],  
       verifyList:[{//设备校验信息列表  
       deviceId:’’,//设备ID  
       deviceNumber:’’,//设备编号  
       deviceType:’’,//设备类型  
       verifyType:’’,//校验类型  
       verifyTime:’’,//校验时间  
       verifyAddress:’’,//校验地址  
       verifyLongitude:’’,//校验时经度  
       verifyLatitude:’’,//校验时纬度  
       verifyResult:’’,//设备返回结果  
       verifyUserId:’’,//校验操作账号ID  
       verifyLoginName:’’,//操作账号  
       verifyAppId:’’,//操作设备  
       fileList:[{//拍照列表  
       fileName:’’,//文件名称  
       fileSize:’’,//文件大小  
       fileMime:’’,//文件类型  
       fileBytes:’’,//文件Base64  
       }]  
       }],  
         
      }
   2. 接口地址：/iot-seal-app-api/saveTaskInfo
   3. 接口规则  
      1）保存货单基本信息。（basics\_task\_info）  
      2）保存货单车辆车箱设备绑定信息。（relation\_task\_car\_container\_device）  
      3）保存货单设备校验信息。（basics\_device\_verify\_info）  
      4）根据货单设备校验异常状态，生成货单告警信息，并保存至货单告警表（basics\_task\_alarm\_info）  
      5）保存照片记录信息。（basics\_file\_info，其中relatedEntity为BasicsDeviceVertifyInfo，relatedId为货单设备校验信息表ID）
   4. 返回信息：  
      {  
       code:’’,  
       msg:’’,  
       data:{  
       taskId:’’,  
       taskNumber:’’  
       }  
      }  
      000000：请求成功；100500：数据异常
4. 货单留言查询接口
   1. 参数示例：  
      {  
       taskId:’’,//货单ID  
       taskNumber:’’//货单编号  
      }
   2. 接口地址：/iot-seal-app-api/findTaskMsgList
   3. 接口规则：  
      1）查询货单留言表（basics\_task\_message\_info）  
      2）查询货单留言照片（basics\_file\_info）
   4. 返回信息  
      {  
       code:’’,  
       msg:’’,  
       data:[{  
       id:’’,//留言ID  
       taskId:’’,//货单ID  
       taskNumber:’’,//货单编号  
       message:’’,//留言信息  
       userId:’’,//留言人  
       msgTime:’’,//留言时间  
       fileList:[{  
       fileId:’’,//文件ID  
       fileUrl:’’,//文件路径地址  
       }]  
       }]  
      }  
      000000：请求成功；100500：数据异常
5. 货单留言保存接口
   1. 接口示例：  
      {  
       taskId:’’,//货单ID  
       taskNumber:’’,//货单编号  
       message:’’,//留言信息  
       userId:’’,//留言人ID  
       msgTime:’’,//留言时间  
       fileList:[{  
       fileBytes:’’,//fileBase64  
       }]  
      }
   2. 接口地址：/iot-seal-app-api/saveTaskMsg
   3. 接口规则：  
      1）保存货单留言表。（basics\_task\_message\_info）  
      2）保存照片文件表。（basics\_file\_info，其中relatedEntity为BasicsTaskMessageInfo，relatedId为货单留言表ID）
   4. 返回信息：  
      {  
       code:’’,  
       msg:’’,  
       data:{  
       taskMsgId:’’,//货单留言ID  
       }  
      }  
      000000：请求成功；100500：数据异常
6. 当前账号的历史货单查询
   1. 接口示例：{page:1,loginName:’’}
   2. 接口地址：/iot-seal-app-api/findHistoryTaskList
   3. 接口规则  
      1）从历史货单库中查询货单信息。（basics\_task\_info\_history）  
      2）从用户车辆对应关系表中查询用户当前管理车辆和历史曾经管理过的车辆。当前管理车辆判断条件为del\_flag为0，并且查询创建时间之后的货单信息。历史管理车辆筛选条件为del\_flag为2，并且以创建时间和更新时间为时间段查询货单。  
      3）货单信息查询内容：货单ID、货单编号、发货时间、发货地点、运输车辆、收货地址、收货时间、是否告警。
   4. 返回信息：  
      {  
       code:’’,  
       msg:’’,  
       data:{  
       page:1,  
       pageSize:10,  
       total:200,  
       data:[{  
       taskId:’’,//货单ID  
       taskNumber:’’,//货单编号  
       deliverDatetime:’’,//发货时间  
       deliverAddress:’’,//发货地址  
       cars:’’,//运输车辆  
       receivingDatetime:’’,//收货时间  
       receivingAddress:’’,//收货地点  
       isAlarm:’’//是否告警  
       }]  
       }  
      }  
      000000：请求成功；100500：数据异常
7. 货单详情接口
   1. 接口示例：{taskId:’’}
   2. 接口地址：/iot-seal-app-api/getHistoryTaskInfo
   3. 接口规则：  
      1）货单基本信息。（basics\_task\_info\_history）  
      2）货单校验信息。（basics\_device\_verify\_info\_history）  
      3）货单异常信息。（basics\_device\_abnormal\_info）  
      4）货单留言信息。（basics\_task\_message\_info）
   4. 返回信息：  
      {  
       code:’’,  
       msg:’’,  
       data:{  
       taskInfo:{  
       taskId:’’,//货单ID  
       taskNumber:’’,//货单编号  
       carNumber:’’,//车厢编号  
       carNumber:’’,//车牌号  
       deliverDatetime:’’,//发货时间  
       deliverAddress:’’,//发货地点  
       recivingCompany:’’,//收货单位  
       receivingDatetime:’’,//收货时间  
       receivingAddress:’’,//收货地点  
       },  
       taskVerifyInfos:[{  
       deviceNumber:’’,//设备编号  
       deviceType:’’,//设备类型  
       verifyType:’’,//验证类型  
       verifyTime:’’,//校验时间  
       verifyAddress:’’,//校验地址  
       verifyloginName:’’,//校验人  
       verifyImages:’’校验图片  
       }],  
       taskAlarmInfos:[{  
       alarmId:’’,//告警ID  
       alarmContent:’’,//告警内容  
       deviceNumber:’’,//告警设备  
       alarmTime:’’,//告警时间  
       abrogateAlarmLoginName:’’,//消警人  
       abrogateAlarmTime:’’,//消警时间  
       abrogateContent:’’,//消警说明  
       deviceId:’’,//设备ID  
       verifyId:’’,//校验ID  
       }],  
       taskMsgInfos:[{  
       msgId:’’,//留言ID  
       message:’’,//留言信息  
       msgTime:’’,//留言时间  
       }]  
       }  
      }  
      000000：请求成功；100500：数据异常
8. 货单轨迹信息
   1. 接口示例：{taskId:’’}
   2. 接口示例：  
      当前货单轨迹信息接口：/iot-seal-app-api/findTaskGPSInfo  
      历史货单轨迹信息接口：/iot-seal-app-api/findHistoryTaskGPSInfo
   3. 接口规则：  
      根据货单ID查询当前轨迹信息表（basics\_task\_gps\_info）或历史轨迹信息表（basics\_task\_gps\_info\_history）
   4. 返回信息：  
      {  
       code:’’,  
       msg:’’,  
       data:[{  
       longitude:’’,//经度  
       latitude:’’//纬度  
       }]  
      }  
      000000：请求成功；100500：数据异常
9. 货单拍照留言接口
   1. 接口示例：  
      {  
       taskId:’’,//货单ID  
       message:’’,//留言信息  
       imgList:[{//拍照信息  
       fileBytes:’’//Base64  
       }]  
      }
   2. 接口地址：/iot-seal-app-api/saveTaskMsg
   3. 接口规则：  
      1）获取当前登录账号信息，保存为留言人  
      2）根据货单ID获取到货单编号  
      3）以服务器时间为留言时间  
      4）文件保存到文件表中。
   4. 返回信息  
      {  
       code:’’,  
       msg:’’,  
       data:{  
       msgId:’’  
       }  
      }  
      000000：请求成功；100500：数据异常
10. 货单告警说明接口
    1. 接口示例：  
       {  
        taskId:’’,//货单ID  
        deviceId:’’,//设备ID  
        verifyId:’’,//校验ID  
        abnormalCause:’’,//异常说明  
        imgs:[{  
        files:’’//Base64照片  
        }]  
       }
    2. 接口地址：/iot-seal-app-api/saveAlarmAbnormal
    3. 接口规则：  
       1）保存设备反常说明表（basics\_device\_abnormal\_info）。  
       2）保存文件信息表
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:{  
        abnormalId:’’  
        }  
       }  
       000000：请求成功；100500：数据异常
11. 当前账号货单异常列表接口。
    1. 接口示例:  
       {  
        page:1,  
        pageSize:15,  
        loginName:’’,  
        searchContent:’’  
       }
    2. 接口地址：/iot-seal-app-api/findAbnormalTaskList
    3. 接口规则  
       1）参考正常货单查询接口  
       2）查询限制条件添加是否异常判断，判断异常货单信息。
    4. 返回信息（参考正常货单查询返回信息）
12. 当前账号设备异常列表接口。
    1. 接口示例：  
       {  
        page:1,  
        pageSize:15,  
        loginName:’’,  
        searchContent:’’  
       }
    2. 接口地址：/iot-seal-app-api/findAbnormalDeviceList
    3. 接口规则：  
       1）查询箱体编号、车牌号、设备数量、告警数量、第一次告警时间等。  
       2）查询箱体信息  
        1、账号箱体关系表中的箱体信息  
        2、账号车辆关系表中的车辆的当前货单中的货单车辆箱体关系表中的箱体信息。  
       3）箱体设备数量统计是货单车辆车箱设备关系表中设备数量。  
       4）告警次数是查询设备中的告警统计及第一次告警时间。  
       5）根据车牌号/箱体号条件查询（模糊）
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:[{  
        containerId:’’,//箱体ID  
        containerNumber:’’,//箱体编号  
        carNumber:’’,//车辆编号  
        alarmCount:’’,//设备告警数量  
        deviceCount:’’,//设备数量  
        alarmTime:’’,//设备第一次告警时间  
        }]  
       }  
       000000：请求成功；100500：数据异常
13. 根据车箱号查询最新的货单信息
    1. 接口示例  
       {  
        containerId:’’  
       }
    2. 接口地址：/iot-seal-app-api/getTaskInfoByContainerId
    3. 接口规则（参考根据货单ID查询货单信息）
    4. 返回信息（参考根据货单ID查询货单信息）
14. 当前账号的车辆列表接口。
    1. 接口示例  
       {  
        page:1,  
        pageSize:15,  
        loginName:’’//账号  
        searchContent:’’  
       }
    2. 接口地址：/iot-seal-app-api/findCarListByUser
    3. 接口规则  
       1）查询内容包括车辆ID、车牌号、车主姓名、联系电话、所属部门、车辆状态。  
       2）车辆信息表（basics\_car\_info）、账号车辆关系表（relation\_user\_car）。  
       3）判断条件：①账号关系表中del\_flag为0，查询create\_time之后的车辆；②账号车辆关系表中del\_flag为2，查询create\_time和update\_time之间的车辆。  
       4）所属部门为所属用户所在的部门  
       5）分页查询，车主手机号和车牌号条件查询。
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:{  
        page:1,//当前页码  
        pageSize:15,//分页大小  
        total:20,//车辆总数  
        data:[{  
        carId:’’,//车辆ID  
        carNumber:’’,//车辆编号  
        carType:’’,//车辆类型  
        carState:’’,//车辆状态  
        tonnage:’’,//允许吨位  
        vehicleOwnerName:’’,//车主姓名  
        phoneNum:’’,//手机号码  
        deptName:’’,//所属用户  
        }]  
        }  
       }  
       000000：请求成功；100500：数据异常
15. 当前账号的箱体列表接口。
    1. 接口示例  
       {  
        page:1,  
        pageSize:15,  
        loginName:’’//账号  
        searchContent:’’  
       }
    2. 接口地址：/iot-seal-app-api/findContainerListByUser
    3. 接口规则：  
       1）查询箱体ID，箱体编号、所属车辆、箱门数量、箱体类型、箱体状态等。  
       2）查询表包括箱体信息表（basics\_container\_info），账号箱体关系表（relation\_user\_container）。  
       3）判断条件：①账号关系表中del\_flag为0，查询create\_time之后的箱体；②账号车辆关系表中del\_flag为2，查询create\_time和update\_time之间的箱体。  
       4）分页查询，车箱号条件查询。
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:{  
        page:1,  
        pageSize:15,  
        total:20,  
        data:[{  
        containerId:’’,//箱体ID  
        containerNumber:’’,//箱体编号  
        carNumber:’’,//所属车辆  
        doorCount:’’,//箱门数量  
        containerType:’’,//箱体类型  
        containerState:’’,//箱体状态  
        }]  
        }  
       }  
       000000：请求成功；100500：数据异常
16. 当前账号的物联设备列表接口。
    1. 接口示例  
       {  
        page:1,  
        pageSize:15,  
        loginName:’’,  
        searchContent:’’  
       }
    2. 接口地址：/iot-seal-app-api/findDeviceListByUser
    3. 接口规则  
       1）查询信息包括设备ID、设备编号、设备类型、设备状态、绑定箱体、校验时间、所属部门等。  
       2）查询表包括设备信息表（basics\_device\_info），账号设备关系表（relation\_user\_device）。  
       3）查询条件：①账号关系表中del\_flag为0，查询create\_time之后的设备；②账号车辆关系表中del\_flag为2，查询create\_time和update\_time之间的设备。  
       4）分页查询，设备编号、部门条件查询
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:{  
        page:1,  
        pageSize:15,  
        total:20,  
        data:[{  
        deviceId:’’,//设备ID  
        deviceNumber:’’,//设备编号  
        deviceType:’’,//设备类型  
        deviceState:’’,//设备状态  
        verifyTime:’’,//校验时间  
        deptName:’’,//所属部门  
        bindContainerNumber:’’,//绑定箱体  
        }]  
        }  
       }  
       000000：请求成功；100500：数据异常
17. 车辆添加接口（暂不开发）
18. 箱体添加接口（暂不开发）
19. 设备添加接口（暂不开发）
20. 个人信息接口
    1. 接口示例  
       {  
        loginName:’’//账号  
       }
    2. 接口地址：/iot-seal-app-api/getUserInfo
    3. 接口规则  
       1）查询账号表（sys\_user）中的信息，包括用户姓名、账号名称、手机号码、头像、性别、上次登录时间、IP等
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:{  
        userId:’’,//账号ID  
        loginName:’’,//账号  
        userName:’’,//用户姓名  
        phonenumber:’’,//手机号码（中间四位以\*代替）  
        sex:’’,//性别  
        avatar:’’,//头像路径  
        loginIp:’’,//登录IP  
        loginDate:’’,//登录时间  
        deptName:’’,//所属部门  
        }  
       }  
       000000：请求成功；100500：数据异常
21. 头像上传接口
    1. 接口示例  
       {  
        loginName:’’,//账号  
        file:’’//头像文件Base64  
       }
    2. 接口地址：/iot-seal-app-api/updateUserAvatar
    3. 接口说明  
       1）将文件保存到文件服务器中进行管理，获取文件地址，将文件地址保存到账号表头像中。
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:{  
        fileUrl:’’//头像地址  
        }  
       }  
       000000：请求成功；100500：数据异常
22. 文章信息接口
    1. 接口示例  
       {  
        articleType:’’//文章类型  
       }
    2. 接口地址：/iot-seal-app-api/getArticleByType
    3. 接口规则  
       1）根据文章类型查询文章信息（basics\_article\_info）
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:{  
        articleTitle:’’,//文章标题  
        articleType:’’,//文章类型  
        articleContent:’’//文章内容  
        }  
       }  
       000000：请求成功；100500：数据异常
23. APP版本获取接口
    1. 接口示例
    2. 接口地址：/iot-seal-app-api/getAppVersion
    3. 接口规则  
       1）返回APP升级表中最新的apk的版本号
    4. 返回信息  
       {  
        code:’’,  
        msg:’’,  
        data:{  
        version:’’//版本号  
        }  
       }
24. APP下载接口
    1. 接口地址：/iot-seal-app-api/downloadApk
25. 用户注销接口
    1. 接口地址：/iot-seal-app-api/logout