

# 查询接口

系统支持2种基础协议的查询功能:

- 基于tcp协议
- 基于http协议

其中tcp协议的条码查询功能 *已实现*.

## http查询请求

通过http协议,以post方法向服务器发出比对条码的请求.并接收/解析服务器返回的json数据.

### 条码查询

- **协议** http 2.0/1.01
- **port** 7012
- **url** /query
- **method** post/get
- **encoding**: utf-8
- **args**:
  - 字段名: sn, 类型: 字符串. 含义: 待查询的条码信息
- **return**: 返回值为json格式的键值对.
  - 一般的返回状态是 {"message": "success", "result": result} 表示请求成功.,result是查询的结果.和原来的tcp文档中保持一致.
    - 10401911001201805011536541033317,1 条码结果合格,为本产品当前合格条码
    - 10401911001201805011536541033317,2 不合格,读取的条码已被生产使用
    - 10401911001201805011536541033317,3 不合格,读取的条码不是本产品条码
    - 10401911001201805011536541033317,4 读取的条码格式不符合
  - 如果系统处于不正确的状态.message的值就是相关错误的的消息.会返回类似 {"message":"系统错误"}等错误信息.注意需要解码方可正常显示.
- **example**:

```
# 一个发送请求的例子
import requests

client = requests.Session()          # 获取一个客户端
sn = "待查询条码内容的字符串"
data = {"sn": sn}                   # sn作为查询参数名,
host = "服务器地址:服务器端口"     # 测试的地址是47.99.105.196,端口为7012
u = "http://" + host + ":" + port + "/query" # 服务器接受查询的接口地址.
resp = client.get(u, data=data)     # 用get的方式向服务器发送请求.post也可以的
status = resp.status_code           # 获取服务器的相应状态
if status == 200:
    """服务器正确响应"""
    result = resp.json()             # 获取返回内容并使用json解码
    message = result['message']      # 查询结果标志位.等于success就代表正确
    if message == "success":
        """查询成功,条码是有效的"""
        do something.....
    else:
        """查询失败"""
        process error.....         # 处理错误的函数
```

## 上传合格条码信息

待定

**tips:**

- 测试阶段,可以使用服务器47.99.105.196来做测试. 返回的结果不是真实的.但可以验证通讯格式是否正确. 比如在浏览器打开<http://47.99.105.196:7012/query?sn=112121212>看看?
- 异步模式下,返回查询结果的顺序不一定是和发送的查询请求的顺序一致.
- 如果某些实现有困难,可以重新进行协商

## 扫描枪支持

让嵌入式设备可以插入扫描枪并使用扫描枪查询单个条码状态.

要求:

1. 正确识别扫描枪
2. 提供一个在嵌入式设备上查询条码状态的功能.用户可以使用扫描枪输入条码.向服务器查询条码状态,查询方法要求支持TCP和HTTP两种.

## 一些问题请教一下

- UploadTraceCodeToDb接口每次工作结束后上传合格数据的时候,由于是批量上传.一旦处理发生意外(比如连接中断,或者其中的某个条码识别错误)可有什么预防/避免/重试的机制? 是否需要我这边做些相应的配合或者有什么注意事项?
- 合格数据的文件尺寸比较大,可否考虑过压缩上传?比如gzip之类的.
- 合格数据是隔一段时间上传一次还是一次性全部上传?