**接口自动化测试**

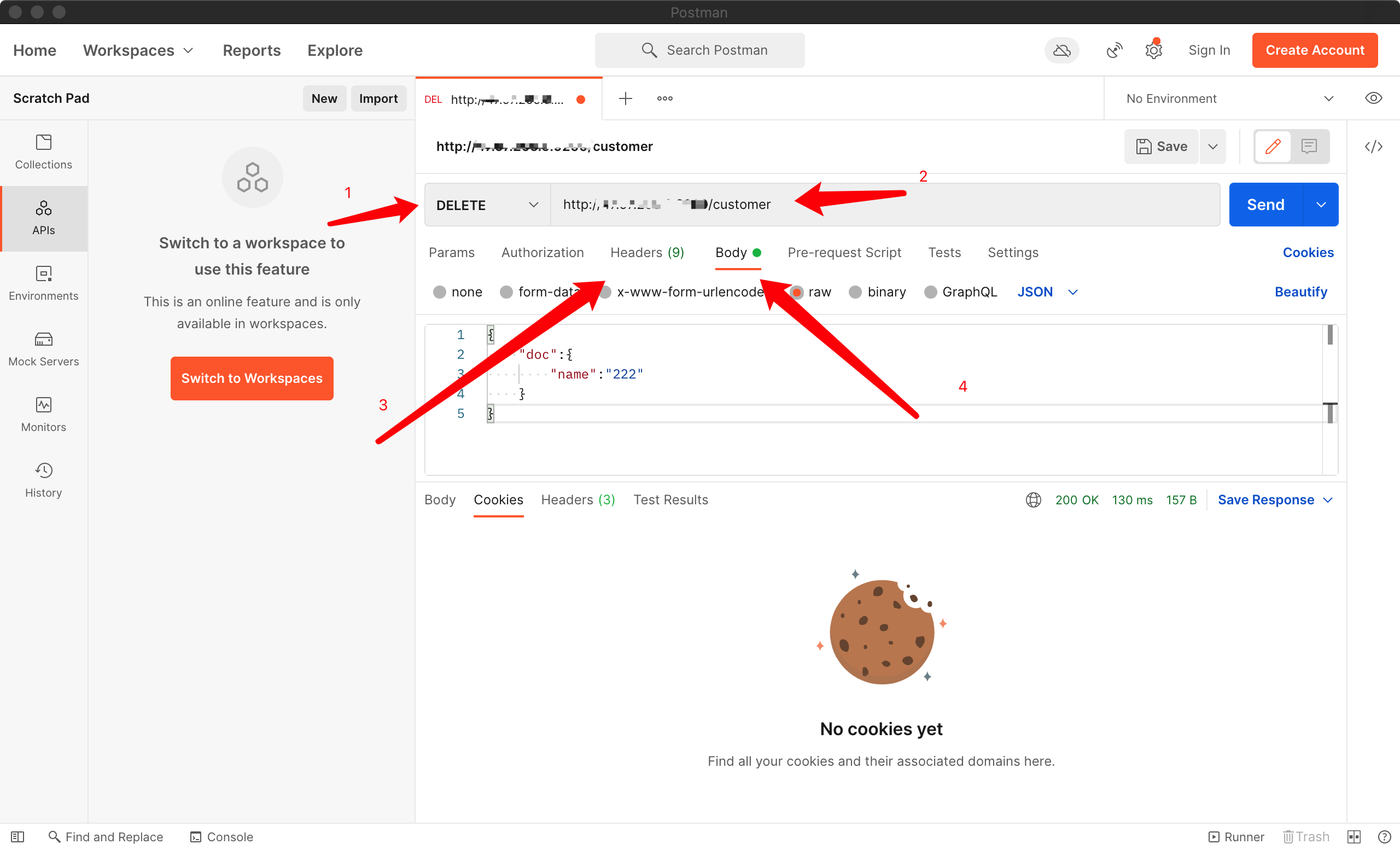
1.Http报文结构(了解)

http请求报文由三部分组成： 请求行、请求头 、请求正文。请求行、每个请求头与请求正文都由CRLF（回车换行，也即\r\n）分割开来，首行为请求行，余下的为请求头和请求正文，而请求正文有可能会为空，也可能包含CRLF情况，因此不能通过一个CRLF与请求头区别开来，所以采用两个CRLF来间隔请求头和请求正文

https://www.cnblogs.com/New-HackerHK/p/15033617.html书签：[HTTP 报文结构 - New\_HackerHK - 博客园](https://www.cnblogs.com/New-HackerHK/p/15033617.html)

https://blog.csdn.net/qq\_40933663/article/details/90045869书签：[HTTP协议格式详解（总结）\_Sasura\_321的博客-CSDN博客\_http](https://blog.csdn.net/qq_40933663/article/details/90045869)

在Postman中体现就是请求方式，请求URL，请求头，请求体。



而在AW编写中也就为了写出这几个key,可能业务的不同会有所差别,但至少是3个,其中请求方式可以写死在程序里,像ws协议发送音频可能需要多一个audio的key,然后每个key的具体内容value值由我们自己手工填充,比如请求方式的value值可以是Get,Post等等。

2.[websocket报文结构](https://blog.csdn.net/LL845876425/article/details/106393358/)

2.AW如何编写

2.1.Swagger编写（yaml文件）

##必要字段！Swagger规范版本，必须填2.0，否则该YAML将不能用于Swagger其他组件  
swagger: "2.0"  
#必要字段！描述API接口信息的元数据  
info:  
 #接口标题  
 title: 设备管理服务  
 #接口文档的描述  
 description: 包含设备管理接口  
 #版本号，自定义  
 version: 1.0.0  
#服务地址，必要，之后统一填集团能力桥接服务网关地址  
host: 127.0.0.1:30006  
##定义的api的前缀，必须/开头，可以不设置该参数，建议设置  
basePath: /iot/device/get  
#接口分类标签，建议设置，可以定义多个，-表示是个数组元素  
tags:  
 - name: device  
 description: 设备管理  
# 指定调用接口的协议，必须是:"http", "https", "ws", "wss"．  
# -表示是个数组元素，即schemes接受一个数组参数  
schemes:  
 - http  
#对应与http协议头request的Accept，调用者产生类型。  
#默认是\*/\*,定义的类型必须是http协议定义的 Mime Types  
#RestfulAPI一般定义成application/json  
#这两个是对所有接口的全局设置  
#在细化的接口中还可以对应这两个属性来覆盖全局属性  
produces:  
 - application/json  
#必要字段!定义可操作的API  
paths:  
 #具体url接口调用路径（若是上文中没有设置basePath的话，这里就需要写全为/iot/device/get/test）  
 /test:  
 ##必要字段!定义HTTP操作方法，必须是http协议定义的方法（get、post、put、delete）  
 get:  
 #上面tags中定义的tag标签，方便快速过滤出某种标签下的接口  
 tags:  
 - device  
 ##接口概要  
 summary: 查询普通设备详情  
 #接口描述  
 description: 查询普通设备的详细信息  
 #接口唯一标识，可以类似于接口的方法名  
 operationId: queryDevice  
 #接口产生类型，局部生效  
 produces:  
 - application/json  
 #请求参数  
 parameters:  
 ##参数key，该参数为deviceId  
 - name: deviceId  
 ## 传递方法，formData表示表单传输，  
 ## query表示url拼接传输，  
 ## path表示作为url的一部分  
 ## body表示http头承载参数(body只能有一个,有body不能再有其他的)  
 ## 一般用query（url拼接传参）、body（类似body体传参）  
 ## header表示在请求头中的参数，比如时间戳，签名  
 in: query  
 #是否必传  
 required: **true**  
 #参数类型  
 type: string  
 #描述  
 description: 设备ID  
 #是否允许传空值，默认false，可以不设置  
 allowEmptyValue: **false**  
 ##返回值描述，必要  
 responses:  
 ##返回的http状态码，自定义  
 '200':  
 description: "successful operation"  
 ##描述返回值  
 schema:  
 ##引用在definitions下定义的ResultBean公共参数，注意此时后面不能再接description属性，因为其definitions下定义ResultBean时已经有了description属性  
 $ref: "#/definitions/ResultBean"  
 "401":  
 description: "Unauthorized"  
 "403":  
 description: "Forbidden"  
 "404":  
 description: "Not Found"  
 default:  
 description: "Unexpected error"  
#定义公共参数  
definitions:  
 #参数名，可以类似于实体类的ResultBean类  
 ResultBean:  
 #类型：string、interger、object、array、boolean、number  
 type: object  
 #属性标题，建议与上面的参数名等同  
 title: ResultBean  
 #属性描述  
 description: 接口请求返回结果  
 #哪些必传字段  
 required:  
 - data  
 - errorCode  
 - errorMsg  
 ##定义属性  
 properties:  
 #属性名，自定义的data字段  
 data:  
 #引用在definitions下定义的DeviceInfo公共参数  
 $ref: "#/definitions/DeviceInfo"  
 errorCode:  
 #类型  
 type: integer  
 #格式  
 format: int32  
 description: 请求结果code  
 errorMsg:  
 type: "string"  
 description: "请求结果描述"  
 #设置参数名，可以类似于实体类的DeviceInfo类  
 DeviceInfo:  
 type: object  
 title: DeviceInfo  
 description: 设备详情  
 required:  
 - id  
 - serialNumber  
 - deviceId  
 - deviceKey  
 - projectCode  
 - venderId  
 - corporation  
 - corporationText  
 - model  
 - modelText  
 - fwVersion  
 - cpVersion  
 - protocolCode  
 - deviceName  
 - deviceType  
 - ipAddr  
 - port  
 - status  
 - acquisitionFrequency  
 - isOnline  
 properties:  
 id:  
 type: integer  
 format: int64  
 description: 主键ID  
 #默认值设为1  
 default: 1  
 serialNumber:  
 type: string  
 description: 唯一编号或mac（不带“-”）  
 deviceId:  
 type: string  
 description: 设备ID  
 deviceKey:  
 type: string  
 description: 设备密钥  
 projectCode:  
 type: string  
 description: 项目编号  
 venderId:  
 type: string  
 description: 供应商ID  
 corporation:  
 type: string  
 description: 厂家编号  
 corporationText:  
 type: string  
 description: 厂家名称  
 model:  
 type: string  
 description: 型号编号  
 modelText:  
 type: string  
 description: 型号名称  
 fwVersion:  
 type: string  
 description: 固件版本号  
 cpVersion:  
 type: string  
 description: 组件版本号  
 protocolCode:  
 type: string  
 description: 对接协议版本号  
 deviceName:  
 type: string  
 description: 设备名称  
 deviceType:  
 type: string  
 description: 设备类型  
 ipAddr:  
 type: string  
 description: IP地址  
 port:  
 type: integer  
 format: int32  
 description: 端口号  
 status:  
 type: integer  
 format: int32  
 description: 状态，1：有效，9：已刪除  
 acquisitionFrequency:  
 type: integer  
 format: int32  
 description: 设备心跳频率，单位分钟，默认10分钟  
 default: 10  
 isOnline:  
 type: integer  
 format: int32  
 description: 是否在线，0:不在线，1:在线  
 #枚举  
 enum:  
 - 0  
 - 1

YAML

2.2.Java编写（jar包与json文件）

(1)常用API

对象转json (JSON.toJSONString)

String str = JSON.toJSONString(recVo);

Java

json 转对象（JSON.parseObject）

Test Obj = JSON.parseObject(str, Test .class);

Java