

OpenAPI 接口说明文档

广州黑格智造信息科技有限公司

文件状态	受控状态	文件标识:	
☐ 草稿 ☐ 正式发布 ☒ 正在修改	☐ 受控 ☒ 非受控	当前版本:	V0.27

版本历史记录

日期	版本号	修订说明
2024-07-02	V0.1	整体文档结构，添加 API 授权方式说明，并介绍如何使用授权调用其他业务接口。
2024-07-05	V0.2	补充业务接口说明。
2024-08-29	V0.3	修改部分接口描述、添加接入实际参数。
2024-09-03	V0.4	修改 2.3.1、2.3.3、2.4.1 请求头参数。
2024-09-04	V0.5	生产中心接口地址修改。
2024-09-05	V0.6	设备中心接口参数修改。
2024-11-05	V0.7	设备中心接口参数修改。
2024-11-20	V0.8	设备中心接口参数修改。
2024-12-03	V0.9	设备中心新增接口。
2025-01-08	V0.10	设备中心接口参数修改。
2025-04-11	V0.11	生产中心新增自动前处理接口。
2025-04-11	V0.12	3.2.3 新增批次号 batchNo
2025-08-20	V0.13	新增用户中心接口，新增密码式授权
2025-08-20	V0.14	3.1.2、3.3 新增管理员操作的相关接口
2025-12-03	V0.15	2.4.1 新增设备树脂、任务队列等信息
2025-12-08	V0.16	生产中心的接口更新入参和返回结果。

2025-12-08	V0.17	授权资源添加 PARTLIB (生产中心零件库)
2025-12-11	V0.18	增加 3.4 零件库模型上传的相关接口; 增加 2.3.1、2.3.3 接口限流控制, 修改 pageSize 入参最大值的设置
2026-01-23	V0.19	增加设计中心订单和文件上传接口
2026-02-25	V0.20	增加消息订阅对接说明
2026-02-26	V0.21	修改设计中心 4.1.1、4.1.2、4.17 订单接口; 增加 5.4.1 设计中心订阅消息体
2026-06-22	V0.22	添加 STM 相关接口内容
2026-07-06	V0.23	添加授权资源 STM
2026-07-09	V0.24	增加设计中心创建普通订单相关接口
2026-07-23	V0.25	完善消息订阅说明和 STM 接口
2026-07-31	V0.26	修改文档受控状态为非受控
2026-08-03	V0.27	完善 STM 接口相关前置条件

1. 前言

1.1. 适用范围

需要与 CLOUD 平台集成，实现通过互联网进行数据通信的合作方，相关开发人员可以参考此文档进行开发接入。

1.2. 术语、定义和缩略语

名词	释义
Client ID	请求接入的应用的应用 id
Client Secret	请求接入的应用的密钥
Scope	请求的资源，例如 NOTIFY、USER、ORDER 等
Redirect Uri	授权回调地址

2. 接口使用说明

2.1. 授权介绍

2.1.1. 准备工作

应用开始接入前，需联系 HeyGears 技术人员开通应用，即获取到 Client ID 和 Client Secret，如果使用授权码模式或者隐藏式接入，需要提供授权回调地址用以接收授权码和 AccessToken。

此外，需合作方公司提供一个回调地址用以接收授权码，该链接地址需要在外网环境下可以正常访问。需将此回调地址提供给 HeyGears 技术人员。HeyGears 技术人员需在开放平台系统中进行配置。

2.1.2. 服务环境

目前 HeyGears Cloud OpenAPI 服务对应的服务域名如下表所示。

环境	服务地址
中国大陆	https://cnopenapi.heygears.cloud/open-platform
日本	https://tkyopenapi.heygears.cloud/open-platform
欧洲	https://paropenapi.heygears.cloud/open-platform
美国	https://usopenapi.heygears.cloud/open-platform

2.1.3. 资源获取

授权成功后可获取访问令牌 AccessToken。请求资源接口时,请在请求头中添加 Authorization: Bearer {AccessToken}, 即可获取相应资源 API 的响应数据。

- 请求示例如下

```
POST https://cnopenapi.heygears.cloud/open-platform/op/{对应业务请求地址}
HTTP/1.1
Authorization: Bearer pairsqzj494lbjiby96
Content-Type: application/json
{
  "xxx1": "xxx",
  "xxx2": 0
}
{
  "code": 200,
  "data": {
  },
  "message": "成功",
  "detailMessage": null,
  "error": false,
```

```
"success": true  
}
```

2.1.4. 授权资源 (Scope) 与接口对应关系

Scope	Api
DEVICE	/device/openApi/v1/queryDeviceList
	/device/openApi/v1/queryDeviceErrorInfo
USER	/user-basic/external/v1/getOrgUsers
PRODUCTION	/production-lab-api-service/listStrategy
	/production-lab-api-service/modelQuery
	/production-lab-api-service/sliceQuery
	/production-lab-api-service/listStrategyAdmin
	/production-lab-api-service/getUploadUrl
	/production-lab-api-service/uploadModelFromS3
	/production-lab-api-service/uploadCompleteConfirm
	/production-lab-api-service/getUploadUrlAdmin

	/production-lab-api-service/uploadModelFromS3Admin
	/production-lab-api-service/uploadCompleteConfirmAdmin
PARTLIB	/production-lab-partlib/getUploadUrl
	/production-lab-partlib/completeUpload
DESIGN	/design-basic-service/openApi/v1/getOrderPageList
	/design-basic-service/openApi/v1/orderInfo
	/design-basic-service/openApi/v1/addOrderByFast
	/design-basic-service/openApi/v1/getDesignTypes
	/design-basic-service/openApi/v1/addOrderByGeneral
DESIGNFILE	/design-file-service/platformS3/v1/getUploadUrl
	/design-file-service/platformS3/v1/completeUpload
	/design-file-service/platformS3/v3/getDownloadUrl
STM	/stm-service/openApi/v1/getPlanList
	/stm-service/openApi/v1/createOrder

/stm-service/openApi/v1/getOrderList

2.2. 授权方式

2.2.1. 授权码式 (Authorization Code)

① 前端跳转至授权登录页

- 接口地址

请求地址	请求类型
/oauth2/authorize	GET

- 请求参数

参数	必填	说明
scope	否	具体请求的权限，多个用逗号隔开（填写 PRODUCTION,DEVICE,USER,PARTLIB,DESIGN,DESIGNFILE,STM 即可）
client_id	是	应用 id
state	否	状态值，此参数会在重定向时追加到 url 末尾，不填不追加
redirect_uri	是	用户确认授权后，重定向的回调地址（参考 2.1.1 准备工作中的内容）
grant_type	是	返回类型，这里请填写：code

- 访问示例

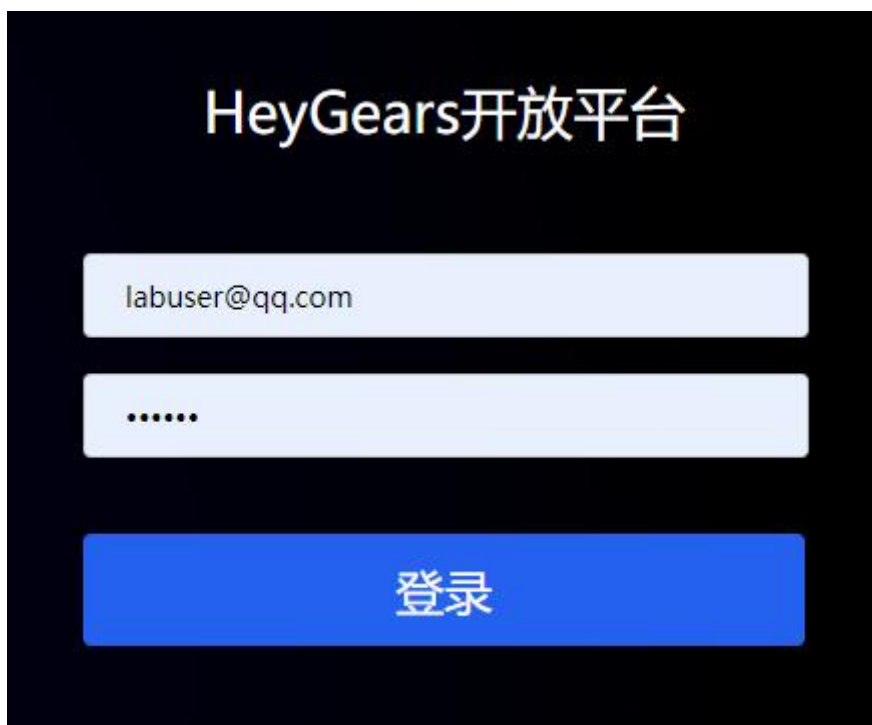
前端通过浏览器访问地址如下

https://cnopenapi.heygears.cloud/open-platform/oauth2/authorize?client_id={client_id}&scope=

PRODUCTION,DEVICE,USER,PARTLIB,DESIGN,DESIGNFILE,STM&grant_type=code&redirect_uri={callback_url}&state={state}

② 开放平台登录

前端访问上述地址，如未经过认证，则跳转到开放平台登录页。暂定如下图所示



HeyGears开放平台

labuser@qq.com

.....

登录

③ 确认授权

使用 HeyGears Cloud 的账号登录成功后，需确认同意授权。如下图所示



操作	跳转	说明
点击拒绝授权	{redirect_uri}?handle=reject	跳转至客户回调地址，并添加参数?handle=reject 作为标识
点击同意授权	{redirect_uri}?code=d1dc2d79c5744bbfad3aaf6950199187&state=123456	跳转至客户回调地址，并添加参数?code=xxx 将授权码携带返回

④ 使用授权码获取 AccessToken

客户通过回调路径，拿到授权码后，需请求以下接口，使用授权码获取 AccessToken。

- 接口地址

请求地址	请求类型
/oauth2/getAccessTokenByCode	GET

- 请求头

参数	内容	说明	示例
Authorization	Basic {base64(client_id:client_secret)}	将 clientId:clientSecret 组成的字符串进行 Base64 编码,并在编码结果前添加 “Basic ” (Basic 后有一个空格)。	Basic xxxxxxxx

- 请求参数

参数	是否必填	说明
code	是	授权码

- 响应字段

参数	说明
accessToken	访问令牌
refreshToken	刷新令牌
accessTokenExpireTime	访问令牌过期时间 (毫秒级时间戳)
refreshTokenExpireTime	刷新令牌过期时间 (毫秒级时间戳)
loginId	HeyGears Cloud 用户编码

- 请求示例如下

```
GET /oauth2/getAccessTokenByCode?code=aab30390bc2c4044a6e096d5f71c2879
HTTP/1.1
Authorization: Basic {base64(client_id:client_secret)}
```

```
Content-Type: application/json

{
  "code": 200,
  "data": {
    "accessToken": "jyhowwhja0gxadwua6us",
    "refreshToken": "1siagxmyrp28a4g81m5h",
    "accessTokenExpireTime": 1719912131738,
    "refreshTokenExpireTime": 1722496931738,
    "loginId": 1010221
  },
  "message": "成功",
  "detailMessage": null,
  "error": false,
  "success": true
}
```

2.2.2. 隐藏式 (Implicit)

暂时不提供，有需要可以洽谈。

2.2.3. 密码式 (Password)

使用 Cloud 账户和密码直接获取授权信息。

- 接口地址

请求地址	请求类型
/oauth2/token	GET

- 请求头

参数	内容	说明	示例
Authorization	Basic base64(clientId:c	将 clientId:clientSecret 组成的字符串进行	Basic xxxxxxxx

	lientSecret)	Base64 编码,并在编码结果前添加 “Basic ” (Basic 后有一个空格) 。	
--	--------------	---	--

- 请求参数

参数	是否必填	说明
scope	是	请求授权的资源
grantType	是	password
userName	是	Cloud 用户邮箱账号
password	是	Cloud 用户密码经过 md5 编码

- 响应字段

参数	说明
accessToken	访问令牌
refreshToken	刷新令牌
accessTokenExpireTime	访问令牌过期时间 (毫秒级时间戳)
refreshTokenExpireTime	刷新令牌过期时间 (毫秒级时间戳)
loginId	HeyGears Cloud 用户编码

- 请求示例如下

```
GET
/oauth2/token?scope=USER,DEVICE,PRODUCTION,PARTLIB,DESIGN,DESIGNFILE,STM&grantType=password&userName=labuserkang@dev.com&password=e10adc3949ba59abbe56e057f20f883e
Authorization: Basic {base64(client_id:client_secret)}
Content-Type: application/json

{
  "code": 200,
  "data": {
    "accessToken": "jyhowwhja0gxadwua6us",
    "refreshToken": "1siagxmyrp28a4g81m5h",
    "accessTokenExpireTime": 1719912131738,
    "refreshTokenExpireTime": 1722496931738,
    "loginId": 1010221
  },
  "message": "成功",
  "detailMessage": null,
  "error": false,
  "success": true
}
```

2.2.4 刷新 Token

Access-Token 的有效期较短，如果每次过期都需要重新授权的话，会比较影响用户体验，因此我们可以在后台通过 Refresh-Token 刷新 Access-Token。

- 接口地址

请求地址	请求类型
/oauth2/refresh	GET

- 请求头

参数	内容	说明	示例
Authorization	Basic base64(clientId:clientSecret)	将 clientId:clientSecret 组成的字符串进行	Basic xxxxxx

		Base64 编码，并在编码结果前添加 “Basic ” (Basic 后有一个空格)。	
--	--	--	--

- 请求参数

参数	是否必填	说明
refreshToken	是	刷新令牌

- 响应字段

参数	说明
accessToken	访问令牌
refreshToken	刷新令牌
accessTokenExpireTime	访问令牌过期时间（毫秒级时间戳）
refreshTokenExpireTime	刷新令牌过期时间（毫秒级时间戳）
loginId	HeyGears Cloud 用户编码

- 请求示例如下

GET /oauth2/refresh?refreshToken=1siagxmyrp28a4g81m5h HTTP/1.1 Authorization: Basic {base64(client_id:client_secret)} Content-Type: application/json
{ "code": 200, "data": { "accessToken": "9uk7nouzoil8xohgtqcg", "refreshToken": "1siagxmyrp28a4g81m5h",

```

      "accessTokenExpireTime": 1719912337695,
      "refreshTokenExpireTime": 1722496931738,
      "loginId": 1010221
    },
    "message": "成功",
    "detailMessage": null,
    "error": false,
    "success": true
  }

```

2.2.5 回收 Token

在 Access-Token 过期前主动将其回收

- 接口地址

请求地址	请求类型
/oauth2/revokeAccessToken	GET

- 请求参数

参数	是否必填	说明
accessToken	是	令牌
clientId	是	应用 id

- 请求示例如下

GET /oauth2/revokeAccessToken?clientId=4d3a7e8775df4c67a817c56ef5cb1513&accessToken=9uk7nouzoil8xohgtqcg HTTP/1.1
{ "code": 200, "data": true, "message": "成功",

```
"detailMessage": null,  
"error": false,  
"success": true  
}
```

2.3. 响应码

响应码	说明
200	操作成功
500	系统内部异常
401	未授权
403	禁止访问
405	方法不允许
415	不支持的媒体类型
400	请求参数错误
66000001	不支持的授权方式
66000002	客户端不存在
66000003	客户端密钥错误
66000004	请求授权范围超出或无效
66000005	重定向地址错误
66000006	用户密码错误

66000007	无效 refreshToken
66000008	无效 code
66000009	code 码与登录用户不匹配
66000010	code 码与 clientId 不匹配
66000011	授权未同意
66000012	AccessToken 不存在
66000013	AccessToken 与 clientId 不匹配
70039005	传入的应用类型参数有误
70039006	获取方案列表数据出现异常
70039007	获取方案列表数据失败
65000200	当前用户不是管理员
70039008	订单不存在
70039009	订单状态流转不合法
70039010	接收用户无权下单
70039011	方案不存在或不属于该用户

3. 消息订阅说明

3.1. 基础说明

第 15 页 共 134 页

www.heygears.com

sales@heygears.com

Tel: +1(318)353-4295 (Global) / +1(949)418-9418 (USA) / +44148-396-8549 (Europe)

USA: 17931SkyParkCircle, SuiteE,Irvine, CA, 92614

CHN: BlockB2, 501, 601, EnterpriseAccelerator,KaifaDistrict, Guangzhou, Guangdong,China

开放平台对外提供两种消息订阅方式：

方式	协议	特点	适用场景
WebSocket 长连接	STOMP over WebSocket	服务端实时推送	需要低延迟、持续监听的业务
HTTP 定时轮询	HTTPS	客户端主动拉取	无长连接能力，或可接受秒级延迟的场景

3.1.1. 业务与队列

- 业务消息与队列绑定。可订阅哪些队列，取决于当前 OAuth2 账户是否已开通 MQ 权限。
- 不同队列的消息体结构不同，请按具体业务场景使用。
- 队列名称 / 订阅 destination 由 HeyGears 按业务下发，请勿自行猜测。

3.1.2. 使用前提

- 已完成 OAuth2 授权，拿到有效 access_token（见 2.2）。
- 组织已开通 RabbitMQ 虚拟主机（未开通时 WebSocket 会拒绝连接，HTTP 接口会返回错误）。
- 已获知目标业务对应的队列名（及 WebSocket 场景下的 destination）。

3.1.3. 两种方式对比

对比项	WebSocket	HTTP
实时性	高（推送）	取决于轮询间隔
鉴权位置	连接 URL Query: token	Header : Authorization: Bearer {token}
发送消息	不支持（SEND 会被拒绝）	不支持
推荐轮询间隔（HTTP）	—	建议 3 ~ 10 秒

3.2. 基于 WebSocket 的消息订阅（长连接）

开放平台 WebSocket 通道采用 STOMP 协议。客户端连接开放平台后，由开放平台代理转发至 RabbitMQ Web-STOMP；客户端只需按 STOMP 规范发送帧即可。

说明：客户端 CONNECT 帧中的 login / passcode / host 无需自行填写正确值。开放平台会根据 access_token 自动注入 MQ 凭据与虚拟主机信息。

3.2.1. 接入准备

目前 HeyGears Cloud OpenApi 服务对应的 WebSocket 服务地址如下表格所示。

环境	服务地址
中国大陆	<code>wss://cnopenapi.heygears.cloud/open-platform/mq/ws</code>
日本	<code>wss://tkyopenapi.heygears.cloud/open-platform/mq/ws</code>
欧洲	<code>wss://paropenapi.heygears.cloud/open-platform/mq/ws</code>
美国	<code>wss://usopenapi.heygears.cloud/open-platform/mq/ws</code>

接入 WebSocket 服务需准备好授权认证成功后的 AccessToken，可参考 2.2 授权方式获取 AccessToken。

完整连接 URL：

`{服务地址}?token={access_token}`

示例（中国大陆）：

`wss://cnopenapi.heygears.cloud/open-platform/mq/ws?token=your_access_token`

常见连接拒绝原因

原因	说明
AccessToken 为空	未传 token
AccessToken 无效或过期	需重新授权获取
组织编号为空	Token 关联用户信息异常，需重新授权
虚拟主机未开通	联系 HeyGears 技术支持开通 MQ

3.2.2. 消息确认机制

订阅时通过 STOMP 头 ack 指定确认模式：

确认模式	工作原理	适用场景
auto	消息投递后由 Broker 自动确认	可容忍少量丢失的场景（如日志、监控）
client	确认一条消息等同确认此前所有未确认消息	严格顺序处理的任务队列
client-individual	每条消息需单独 ACK / NACK	并行处理、需精确确认的业务消息

使用 client 或 client-individual 时，业务处理成功后发送 ACK，失败可发送 NACK。

3.2.3. 连接流程

步骤 1: 建立 WebSocket 连接（携带 token，协商 STOMP 子协议）

↓

步骤 2: 发送 STOMP CONNECT → 收到 CONNECTED

↓

步骤 3: 发送 SUBSCRIBE 订阅队列

↓

步骤 4: 接收 MESSAGE; 按 ack 模式决定是否 ACK/NACK

3.2.4. 前端接入 (JavaScript)

基础连接示例: (服务地址以中国大陆为例)

```
const token = "your_access_token";
const wsUrl =
`wss://cnopenapi.heygears.cloud/open-platform/mq/ws?token=${encodeURIComponent(token)}`;
// 协商 STOMP 子协议
const ws = new WebSocket(wsUrl, ["v12.stomp", "v11.stomp", "v10.stomp"]);
let heartbeatTimer = null;
ws.onopen = function () {
  console.log("WebSocket 连接成功");
  // CONNECT: login/passcode/host 由开放平台注入, 客户端无需填写真实值
  const connectFrame =
    "CONNECT\n" +
    "accept-version:1.2\n" +
    "host: /\n" +
    "heart-beat:10000,10000\n" +
    "\n" +
    "\0";
  ws.send(connectFrame);
  // 心跳保活 (建议 10 秒)
  heartbeatTimer = setInterval(() => {
    if (ws.readyState === WebSocket.OPEN) {
      ws.send("\n");
    }
  }, 10000);
};

ws.onmessage = function (event) {
  const frame = event.data;
```

```
// Broker / 代理心跳
if (frame === "\n" || frame === "\r\n" || frame === "") {
    return;
}
if (frame.startsWith("CONNECTED")) {
    console.log("STOMP 握手成功, 开始订阅");
    // destination 以 HeyGears 下发为准
    subscribeAuto("/queue/yourQueueName");
    return;
}
if (frame.startsWith("MESSAGE")) {
    const { headers, body } = parseStompFrame(frame);
    console.log("收到消息:", body);
    // 若订阅时 ack 为 client / client-individual, 处理成功后需 ACK
    // ackMessage(headers["ack"]);
    return;
}
if (frame.startsWith("ERROR")) {
    console.error("STOMP 错误:", frame);
}
};

ws.onclose = function (event) {
    console.warn("连接关闭:", event.code, event.reason);
    if (heartbeatTimer) {
        clearInterval(heartbeatTimer);
    }
    // 建议在此实现重连
};

ws.onerror = function (err) {
    console.error("WebSocket 错误:", err);
};

/** 自动确认订阅 */
```

```
function subscribeAuto(destination) {
    const subscribeFrame =
        "SUBSCRIBE\n" +
        "id:sub-" + Date.now() + "\n" +
        "destination:" + destination + "\n" +
        "ack:auto\n" +
        "\n" +
        "\0";
    ws.send(subscribeFrame);
}

/** 手动确认订阅: mode = client | client-individual */
function subscribeManual(destination, mode = "client-individual") {
    const subscribeFrame =
        "SUBSCRIBE\n" +
        "id:sub-" + Date.now() + "\n" +
        "destination:" + destination + "\n" +
        "ack:" + mode + "\n" +
        "\n" +
        "\0";
    ws.send(subscribeFrame);
}

function ackMessage(ackId) {
    const ackFrame =
        "ACK\n" +
        "id:" + ackId + "\n" +
        "\n" +
        "\0";
    ws.send(ackFrame);
}

function nackMessage(ackId) {
    const nackFrame =
        "NACK\n" +
```

```
        "id:" + ackId + "\n" +
        "\n" +
        "\0";
        ws.send(nackFrame);
    }
    /** 简易 STOMP 文本帧解析 */
    function parseStompFrame(frame) {
        const nullIdx = frame.indexOf("\0");
        const text = nullIdx >= 0 ? frame.substring(0, nullIdx) : frame;
        const headerBodySplit = text.indexOf("\n\n");
        const headerPart = headerBodySplit >= 0 ? text.substring(0,
headerBodySplit) : text;
        const body = headerBodySplit >= 0 ? text.substring(headerBodySplit + 2) :
"";

        const lines = headerPart.split("\n");
        const headers = {};
        for (let i = 1; i < lines.length; i++) {
            const idx = lines[i].indexOf(":");
            if (idx > 0) {
                headers[lines[i].substring(0, idx)] = lines[i].substring(idx + 1);
            }
        }
        return { command: lines[0], headers, body };
    }
}
```

3.2.5. 后端接入 (Java)

Maven 添加依赖

```
<dependency>
  <groupId>org.java-websocket</groupId>
  <artifactId>Java-WebSocket</artifactId>
  <version>1.5.3</version>
```

```
</dependency>
```

使用 Java-WebSocket 客户端

```
import org.java_websocket.client.WebSocketClient;
import org.java_websocket.drafts.Draft_6455;
import org.java_websocket.handshake.ServerHandshake;
import org.java_websocket.protocols.Protocol;
import java.net.URI;
import java.util.Collections;
import java.util.Map;
import java.util.concurrent.ConcurrentHashMap;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
/**
 * 开放平台 STOMP over WebSocket 订阅客户端示例。
 * 说明: CONNECT 凭据由开放平台根据 token 注入; 客户端禁止发送 SEND。
 */
public class StompClient extends WebSocketClient {

    private final Map<String, MessageHandler> handlers = new
ConcurrentHashMap<>();

    private final ScheduledExecutorService scheduler =
Executors.newSingleThreadScheduledExecutor();

    private volatile boolean heartbeatStarted = false;

    public StompClient(String serverUrl, String token) {
        super(
            URI.create(serverUrl + "?token=" + token),
            // 协商 STOMP 子协议, 与开放平台 getSubProtocols 对齐
            new Draft_6455(Collections.emptyList(),
Collections.singletonList(new Protocol("v12.stomp")))
        );
    }
}
```

第 23 页 共 134 页

```
}

@Override
public void onOpen(ServerHandshake handshake) {
    System.out.println("WebSocket 连接成功");
    sendConnect();
    startHeartbeat();
}

private void sendConnect() {
    String connectFrame =
        "CONNECT\n" +
        "accept-version:1.2\n" +
        "host: /\n" +
        "heart-beat:10000,10000\n" +
        "\n" +
        "\0";
    send(connectFrame);
}

private void startHeartbeat() {
    if (heartbeatStarted) {
        return;
    }
    heartbeatStarted = true;
    scheduler.scheduleAtFixedRate(() -> {
        if (isOpen()) {
            send("\n");
        }
    }, 10, 10, TimeUnit.SECONDS);
}

@Override
public void onMessage(String message) {
```

```
        if (message == null || message.isEmpty() || "\n".equals(message) ||
"\r\n".equals(message)) {
            return;
        }
        if (message.startsWith("CONNECTED")) {
            System.out.println("STOMP 连接成功");
            onStompConnected();
        } else if (message.startsWith("MESSAGE")) {
            handleMessage(message);
        } else if (message.startsWith("ERROR")) {
            System.err.println("STOMP 错误: " + message);
        }
    }

    private void handleMessage(String frame) {
        String ackId = null;
        String subscription = null;
        String[] lines = frame.split("\n", -1);
        int i = 1;
        for (; i < lines.length; i++) {
            String line = lines[i];
            if (line.isEmpty()) {
                i++;
                break;
            }
            if (line.startsWith("ack:")) {
                ackId = line.substring("ack:".length());
            } else if (line.startsWith("subscription:")) {
                subscription = line.substring("subscription:".length());
            }
        }
        StringBuilder body = new StringBuilder();
        for (; i < lines.length; i++) {
```

```
        if (lines[i].endsWith("\0")) {
            body.append(lines[i], 0, lines[i].length() - 1);
            break;
        }
        body.append(lines[i]);
        if (i + 1 < lines.length) {
            body.append('\n');
        }
    }

    MessageHandler handler = subscription == null ? null :
handlers.get(subscription);
    if (handler != null) {
        handler.onMessage(body.toString(), ackId, subscription);
    }
}

/**
 * @param destination HeyGears 下发的 destination, 例如 /queue/xxx
 * @param ackMode      auto | client | client-individual
 */
public void subscribe(String destination, String ackMode, MessageHandler
handler) {
    String subscriptionId = "sub-" + System.currentTimeMillis();
    String subscribeFrame =
        "SUBSCRIBE\n" +
        "id:" + subscriptionId + "\n" +
        "destination:" + destination + "\n" +
        "ack:" + ackMode + "\n" +
        "\n" +
        "\0";
    send(subscribeFrame);
    handlers.put(subscriptionId, handler);
}
```

```
}

public void ack(String ackId) {
    send("ACK\nid:" + ackId + "\n\n\0");
}

public void nack(String ackId) {
    send("NACK\nid:" + ackId + "\n\n\0");
}

public interface MessageHandler {
    void onMessage(String body, String ackId, String subscriptionId);
}

/** 子类可重写：在 STOMP CONNECTED 之后发起订阅 */
protected void onStompConnected() {

@Override
public void onClose(int code, String reason, boolean remote) {
    System.out.println("连接关闭: " + reason);
    scheduler.shutdownNow();
}

@Override
public void onError(Exception ex) {
    ex.printStackTrace();
}
}
```

使用示例

```
public class OrderConsumer {
    public static void main(String[] args) throws Exception {
```

第 27 页 共 134 页

```
String serverUrl =
"wss://cnopenapi.heygears.cloud/open-platform/mq/ws";
String token = "your_access_token";
StompClient client = new StompClient(serverUrl, token) {
    @Override
    protected void onStompConnected() {
        // 自动确认
        subscribe("/queue/orderState", "auto", (body, msgId, subId) ->
            System.out.println("收到订单状态: " + body));
        // 逐条手动确认
        subscribe("/queue/paymentResult", "client-individual", (body,
ackId, subId) -> {
            try {
                processPayment(body);
                ack(ackId);
            } catch (Exception e) {
                nack(ackId);
            }
        });
    }
};

client.connectBlocking();
}
private static void processPayment(String message) {
    // 业务处理
}
}
```

3.3. 基于 HTTP 的消息订阅 (定时轮询)

3.3.1. 接入准备

接口列表

接口	方法	路径	功能
获取消息 (预览)	GET	/mq/http/peekMessage	从队列取 1 条消息, 不确认 (消息仍留在队列)
消费消息	GET	/mq/http/consumeMessage	从队列取 1 条消息并自动确认 (消息出队)

公共约定

项	说明
鉴权	Header: Authorization: Bearer {access_token}
队列参数	Query: queue={队列名称} (HeyGears 下发的队列名, 不含 /queue/ 前缀)
单次条数	每次固定获取 1 条
队列为空	响应体为 []
建议轮询	3 ~ 10 秒一次; 无消息时勿高频空转

3.3.2. 获取消息接口

接口说明

从指定队列获取一条消息, **不会确认**。消息仍保留在队列中, 适合先预览内容再决定后续处理。若需真正消费, 请调用 consumeMessage, 或在确认业务可处理后改用消费接口。

请求说明

GET /mq/http/peekMessage?queue={队列名称}

Authorization: Bearer {access_token}

完整示例（中国大陆）：

GET https://cnopenapi.heygears.cloud/open-platform/mq/http/peekMessage?queue=orderState

Authorization: Bearer your_access_token

请求参数

参数	位置	必填	类型	说明
queue	Query	是	String	队列名称
Authorization	Header	是	String	Bearer {access_token}

响应示例（成功）

```
{
  "payload": "{\"orderId\":\"123456\",\"status\":\"PAID\",\"amount\":99.90}",
  "routing_key": "order.paid"
}
```

响应示例（队列为空）

```
[]
```

响应字段说明

字段	类型	说明
payload	String	解码后的消息内容（通常为 UTF-8 JSON 字符串）
routing_key	String	消息 routing key（可能与队列名不同，以实际返回为准）

3.3.3. 消费消息接口

接口说明

从指定队列获取一条消息并**自动确认 (ACK)**。消息被消费后将从队列移除，请确保调用方具备幂等处理能力。

请求说明

GET /mq/http/consumeMessage?queue={队列名称}

Authorization: Bearer {access_token}

完整示例（中国大陆）：

GET https://cnopenapi.heygears.cloud/open-platform/mq/http/consumeMessage?queue=orderState

Authorization: Bearer your_access_token

请求参数

参数	位置	必填	类型	说明
queue	Query	是	String	队列名称
Authorization	Header	是	String	Bearer {access_token}

响应示例（成功）

```
{  
  "payload": "{\"orderId\":\"123456\",\"status\":\"PAID\",\"amount\":99.90}",  
  "routing_key": "order.paid"  
}
```

响应示例（队列为空）

```
[]
```

响应字段说明

与 3.3.2 相同。

4. 基础数据接口

4.1. 用户中心

4.1.1. 组织下所有用户查询

查询某个客户下的所有用户信息

- 接口地址

请求地址	请求类型
/user-basic/external/v1/getOrgUsers	GET

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

无

- 响应字段

参数	类型	说明
code	Integer	响应码

message	String	成功/失败信息
detailMessage	String	详细信息
data	Array of UserInfo	用户信息列表。
UserInfo		
userCode	Integer	用户编号
realName	String	用户名称
mobilePrefix	String	手机区号
mobile	String	手机号
email	String	邮箱

- 请求示例如下

```
GET /user-basic/external/v1/getOrgUsers
Authorization: Bearer {access_token}
Content-Type: application/json

{
  "code": 200,
  "data": [
    {
      "userCode": 100119,
      "realName": "lab 用户端用户",
      "mobilePrefix": "+86",
      "mobile": "",
      "email": "labuser@qq.com"
    },
    {
      "userCode": 100126,
      "realName": "lab 管理员",
      "mobilePrefix": "+86",
```

```

        "mobile": "",
        "email": "labuserkang@dev.com"
    },
    "message": "成功",
    "detailMessage": null,
    "success": true,
    "error": false
}

```

5. 生产与设备接口

5.1. 生产与切片

5.1.1. 零件信息批量查询

- 功能描述：根据入参查询零件信息。查询间隔为 10 秒。
- 接口地址

请求地址	请求类型
/production-lab-api-service/modelQuery	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
startTime	是	String	开始时间，举例： 2024-04-01 13:12:21
endTime	是	String	截止时间，举例： 2024-04-01 14:02:00
pageNum	否	Integer	页码，默认为 1，举例： 3
pageSize	否	Integer	每页记录数，默认为 10，举例： 20 最大值不能超过 50

• 响应字段

参数	类型	说明
caseName	String	代型应用的 case 名称
partId	String	零件 Id
modelName	String	零件名称
From	String	来源
modelType	String	零件类型
status	String	零件状态
sender	Integer	发送人
receiver	String	接收人

handler	String	处理人
sliceName	String	关联切片
deviceName	String	实际打印设备名称
deviceSn	String	实际打印设备 SN 码
layerThickness	Integer	实际打印层厚
estimatedTime	Integer	预估用时, 单位秒
estimatedMaterialCost	String	切片预估材料用量, 单位克
printStartTime	String	实际打印开始时间
PrintEndTime	String	实际打印结束时间
totalCount	Integer	总数据量
totalPage	Integer	总页数
currentPage	Integer	当前页码
pageSize	Integer	页大小

- 请求示例如下

POST /production-lab-api-service/modelQuery HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json { "startTime": "2024-04-01 13:12:21", "endTime": "2024-04-02 13:12:21" }
{ "code": 200, }

```
"data": {
  "data": [
    {
      "sliceName": "19_117019_20240801001.ult",
      "sliceId": "19100302122093202407050001",
      "status": "01",
      "appType": "代型牙模",
      "printerName": "UltraCraft DS",
      "materialName": "Model HP 2.0 UV Grey XY",
      "layerThickness": 50,
      "deviceName": "UltraCraft DS",
      "deviceSn": "25A20211041A",
      "estimatedTime": "100",
      "estimatedMaterialCost": 200,
      "printStartTime": "2024-04-01 13:12:21",
      "PrintEndTime": "2024-04-01 13:13:51",
      "modelNameList": [
        "模型 1.stl",
        "模型 2.stl",
        "模型 3.stl"
      ]
    },
    {
      "sliceName": "19_117019_20240801002.ult",
      "sliceId": "19100302122093202407050002",
      "status": "01",
      "appType": "代型牙模",
      "printerName": "UltraCraft DS",
      "materialName": "Model HP 2.0 UV Grey XY",
      "layerThickness": 50,
      "deviceName": "UltraCraft DS",
      "deviceSn": "25A20211041A",
      "estimatedTime": "100",
      "estimatedMaterialCost": 200,
      "printStartTime": "2024-04-01 13:12:26",
      "PrintEndTime": "2024-04-01 13:13:56",
      "modelNameList": [
```

```

        "模型 1.stl",
        "模型 2.stl",
        "模型 3.stl"
    ]
}
],
"totalCount": 10,
"totalPage": 1,
"currentPage": 1,
"pageSize": 1
},
"message": "成功",
"detailMessage": null,
"success": true,
"error": false
}

```

5.1.2. 零件生产记录查询

暂时不提供，有需要可以洽谈。

5.1.3. 切片信息批量查询

- 功能描述：根据入参查询切片信息。查询间隔为 10 秒。
- 接口地址

请求地址	请求类型
/production-lab-api-service/sliceQuery	POST

- 请求头

参数	内容	说明	示例
----	----	----	----

Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq
---------------	----------------------	--	--------------------------------

• 请求参数

参数	是否必填	参数类型	说明
startTime	是	String	开始时间，举例： 2024-04-01 13:12:21
endTime	是	String	截止时间，举例： 2024-04-01 14:02:00
pageNum	否	Integer	页码，默认为 1，举例： 3
pageSize	否	Integer	每页记录数，默认为 10，举例：20 最大值不能超过 50

• 响应字段

参数	类型	说明
sliceName	String	切片名称
sliceId	String	切片 Id
status	String	切片状态：待切片（01）、切片中（02）、已切片且尚

		未下发 (03) 、已下发但尚未打印完成 (04) 、已打印 (05)
appType	String	应用类型
printerName	String	切片适配机型
materialName	String	切片适配材料
layerThickness	Integer	切片适配层厚
deviceName	String	实际打印设备名称
deviceSn	String	实际打印设备 SN 码
estimatedTime	Integer	预估用时, 单位秒
estimatedMaterialCost	String	预估材料用量, 单位克
printStartTime	String	实际打印开始时间
PrintEndTime	String	实际打印结束时间
modelNameList	List	关联零件
totalCount	Integer	总数据量
totalPage	Integer	总页数
currentPage	Integer	当前页码
pageSize	Integer	页大小

- 请求示例如下

第 40 页 共 134 页

```
POST /production-lab-api-service/sliceQuery HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{
  "startTime": "2024-04-01 13:12:21",
  "endTime": "2024-04-02 13:12:21"
}

{
  "code": 200,
  "data": {
    "data": [
      {
        "sliceName": "19_117019_20240801001.ult",
        "sliceId": "19100302122093202407050001",
        "status": "01",
        "appType": "代型牙模",
        "printerName": "UltraCraft DS",
        "materialName": "Model HP 2.0 UV Grey XY",
        "layerThickness": 50,
        "deviceName": "UltraCraft DS",
        "deviceSn": "25A20211041A",
        "estimatedTime": "100",
        "estimatedMaterialCost": 200,
        "printStartTime": "2024-04-01 13:12:21",
        "PrintEndTime": "2024-04-01 13:13:51",
        "modelNameList": ["模型 1.stl", "模型 2.stl", "模型 3.stl"]
      },
      {
        "sliceName": "19_117019_20240801002.ult",
        "sliceId": "19100302122093202407050002",
        "status": "01",
        "appType": "代型牙模",
        "printerName": "UltraCraft DS",
        "materialName": "Model HP 2.0 UV Grey XY",
        "layerThickness": 50,
        "deviceName": "UltraCraft DS",
        "deviceSn": "25A20211041A",
        "estimatedTime": "100",
        "estimatedMaterialCost": 200,
        "printStartTime": "2024-04-01 13:12:26",
        "PrintEndTime": "2024-04-01 13:13:56",
        "modelNameList": ["模型 1.stl", "模型 2.stl", "模型 3.stl"]
      }
    ],
    "totalCount": 10,
    "totalPage": 1,
  }
}
```

```
{
  "currentPage": 1,
  "pageSize": 1
},

{
  "message": "成功",
  "detailMessage": null,
  "success": true,
  "error": false
}
```

5.1.4. 前处理方案查询

暂时不提供，有需要可以洽谈。

5.2. 设备与工艺包

5.2.1. 关联设备查询

查询某个客户下的所有设备信息(接口信息缓存 1 分钟)。

- 接口地址

请求地址	请求类型
/device/openApi/v1/queryDeviceList	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	类型	说明
pageNo	是	Integer	指定显示返回结果中的第几页。
pageSize	是	Integer	指定返回结果中每页显示的设备数量

- 响应字段

参数	类型	说明
code	Integer	响应码
pageNo	Integer	返回结果中的第几页
pageSize	Integer	返回结果中每页显示的设备数量
totalSize	Integer	总条数
totalPages	Integer	总页数
message	String	成功/失败信息
detailMessage	String	详细信息
data	Array of DeviceInfo	设备信息列表。 说明 返回的设备信息按照设备创建时间倒序排列。
DeviceInfo		

deviceSn	String	设备编号
deviceName	String	设备名称
deviceTypeName	String	设备类型名称
deviceState	Integer	设备状态, 0 离线, 1 空闲, 2 工作中, 3 维护中, 4 设备异常
softwareVer	String	软件版本号, 如:1.1.1.1
printingTaskName	String	在打印任务名称(如果设备正在打印中, 则展示)
totalWorkTimes	Long	累计工作时长(单位为秒)。从绑定时间开始, 累计的工作时长 (打印开始时间-打印结束时间的累加), 解绑后清空
totalWorkDays	Integer	累计工作天数。从绑定时间开始, 累计的工作天数, 工作天数定义为: 当天有任务打印成功, 计数加一, 没有任务打印或没有任务打印成功则不计数
recentlyEleclightTime	Long	最近光机校准时间, 返回时间戳
insideHumidity	String	打印机舱内湿度, 单位为%(保留两位小数)
insideTemperature	String	打印机舱内温度, 单位为 °C (保

		留两位小数)
printingTaskProgress	String	在打印任务进度 (不带百分号) , 例: 57
printingTaskEstimatedRemainTime	String	在打印任务预估剩余时间,秒
ReadyTaskTotalCount	Integer	七天内待打印任务总数
ReadyTaskTotalModelNumCount	Integer	七天内待打印任务总零件数
ReadyTaskEstimatedLeftTimeInSecondSum	Long	七天内待打印任务预估用时之和, 不含正在打印的剩余时间, 单位: 秒
usingMaterial1Name	String	在用材料 1 名称
usingMaterial1BottleRemainWeight	Float	在用材料 1 料瓶余量,单位: g
isUsingMaterial1ResinTankRemainEnough	Boolean	在用材料 1 料盘余量是否充足
usingMaterial2Name	String	在用材料 2 名称
usingMaterial2BottleRemainWeight	Float	在用材料 2 料瓶余量,单位: g
isUsingMaterial2ResinTankRemainEnough	Boolean	在用材料 2 料盘余量是否充足
usingMaterial3Name	String	在用材料 3 名称
usingMaterial3BottleRemainWeight	Float	在用材料 3 料瓶余量,单位: g

isUsingMaterial3ResinTank RemainEnough	Boolean	在用材料 3 料盘余量是否充足
collectboxWatermark	Float	成品仓堆件程度,最小为 0, 允许大于 1, 精确到小数点后两位
taskQueue	List<TaskQueueInfo>	未打印任务队列,TaskQueueInfo 包含每个未打印任务的名称和对应任务的预估打印用时。 taskName: 任务名称 estimatedTimeInSecond: 剩余打印时间, 单位: 秒

- 请求示例如下

POST /device/openApi/v1/queryDeviceList HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json { "pageNo": 1, "pageSize":20 }	{ "code": 200, "pageNo": 1, "pageSize": 20, "totalSize": 1, "totalPages": 1, "data": [{ "deviceSn": "abc123456dafaf", "deviceName": "A3D NO.1", "deviceTypeName": "A3D",
---	---

```

"deviceState": 1,
"softwareVer": "1.1.2.3",
"printingTaskName": "2024070501 任务",
"totalWorkTimes": 36000,
"totalWorkDays": 180,
"recentlyEleclightTime": 1720160977000,
"insideHumidity": "56.56",
"insideTemperature": "22.80",
"printingTaskProgress": "57",
"printingTaskEstimatedRemainTime": 120,
"TaskCountTotalReady": 10,
"ReadyTaskTotalModelNumCount": 100,
"ReadyTaskEstimatedLeftTimeInSecondSum": 13000,
"isUsingMaterial1ResinTankRemainEnough": false,
"isUsingMaterial2ResinTankRemainEnough": null,
"isUsingMaterial3ResinTankRemainEnough": null,
"usingMaterial1BottleRemainWeight": 2.33,
"usingMaterial1Name": "Model HP 2.0 UV Grey",
"usingMaterial2BottleRemainWeight": null,
"usingMaterial2Name": null,
"usingMaterial3BottleRemainWeight": null,
"usingMaterial3Name": null,
"collectboxWatermark": 0.6,
"taskQueue": [
  {
    "estimatedTimeInSecond": 114,
    "taskName": "task2"
  },
  {
    "estimatedTimeInSecond": 514,
    "taskName": "task3"
  }
]
},
{
  "deviceSn": "abc123456dafaf",
  "deviceName": "A3D NO.1",

```

```

"deviceTypeName": "A3D",
"deviceState": 1,
"softwareVer": "1.1.2.3",
"printingTaskName": "2024070501 任务",
"totalWorkTimes": 36000,
"totalWorkDays": 180,
"recentlyEleclightTime": 1720160977000,
"insideHumidity": "56.56",
"insideTemperature": "22.80",
"printingTaskProgress": "57",
"printingTaskEstimatedRemainTime": 120,
"TaskCountTotalReady": 10,
"ReadyTaskTotalModelNumCount": 100,
"ReadyTaskEstimatedLeftTimeInSecondSum": 13000,
"isUsingMaterial1ResinTankRemainEnough": false,
"isUsingMaterial2ResinTankRemainEnough": null,
"isUsingMaterial3ResinTankRemainEnough": null,
"usingMaterial1BottleRemainWeight": 2.33,
"usingMaterial1Name": "Model HP 2.0 UV Grey",
"usingMaterial2BottleRemainWeight": null,
"usingMaterial2Name": null,
"usingMaterial3BottleRemainWeight": null,
"usingMaterial3Name": null,
"collectboxWatermark": 0.6,
"taskQueue": [
  {
    "estimatedTimeInSecond": 114,
    "taskName": "task2"
  },
  {
    "estimatedTimeInSecond": 514,
    "taskName": "task3"
  }
]
},
"message": "成功",

```

```

{
  "detailMessage": null,
  "error": false,
  "success": true
}

```

5.2.2. 五分钟内的设备异常信息查询

查询五分钟内的设备异常信息。

- 接口地址

请求地址	请求类型
/device/openApi/v1/queryDeviceErrorInfo	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	类型	说明
deviceSnList	是	Array of String	设备列表

- 响应字段

参数	类型	说明
----	----	----

code	Integer	响应码
message	String	成功/失败信息
detailMessage	String	详细信息
data	Array of ErrorInfo	设备异常信息列表。 说明 返回的异常信息按照设备编号及异常信息发生时间排列。
ErrorInfo		
deviceSn	String	设备编号
taskId	String	任务编号
errorMsg	String	错误信息-中文
errorMsgEn	String	错误信息-英文
eventTime	Long	异常发生时间，返回时间戳
materialName	String	打印任务的材料名

- 请求示例如下

POST /device/openApi/v1/queryDeviceErrorInfo HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json { "deviceSnList": ["deviceSn1","deviceSn2"] }
{ "code": 200,

```
"data": [  
  {  
    "deviceSn": "deviceSn1",  
    "taskId": "D-aabbcc-ddee-ff-gg-hh",  
    "errorMsg": "打印错误",  
    "errorMsgEn": "Print error",  
    "eventTime": 1733189362000,  
    "materialName": "Model HP 2.0 UV Grey"  
  },  
  {  
    "deviceSn": "deviceSn1",  
    "taskId": "D-aabbcc-ddee-ff-gg-ll",  
    "errorMsg": "液位过低",  
    "errorMsgEn": "the liquid level is too low",  
    "eventTime": 1733189422000,  
    "materialName": "Model HP 2.0 UV Grey"  
  },  
  {  
    "deviceSn": "deviceSn2",  
    "taskId": "D-aabbcc-ddee-ff-gg-ss",  
    "errorMsg": "打印错误",  
    "errorMsgEn": "Print error",  
    "eventTime": 1733189362000,  
    "materialName": "Model HP 2.0 UV Grey"  
  }  
],  
"message": "成功",  
"detailMessage": null,  
"error": false,  
"success": true  
}
```

6. 自动前处理接口

6.1. 查询策略列表

6.1.1. 查询策略列表

- 功能：获取用户在云端保存的策略信息
- 接口地址

请求地址	请求类型
/production-lab-api-service/listStrategy	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8i bdq

- 请求参数

参数	是否必填	参数类型	说明
userId	是	Integer	用户 id, 举例: 100901

- 响应字段

参数	类型	说明
----	----	----

strategyId	String	策略 Id
strategyName	String	策略名称
applicationId	Integer	应用 Id
printerId	Integer	打印机 Id
materialId	Integer	材料 id
layerThickness	String	层厚
caseSplitStart	String	按 case 排版文件分隔符开始值（生产中心内部使用）
caseSplitEnd	String	按 case 排版文件分隔符结束值（生产中心内部使用）
folderSwitch	String	开启文件夹多层上传的开关。按 case 排版默认为 true.关-false,开-true
folderDepth	String	文件夹的层数（生产中心内部使用）
unavailable	Boolean	如果工艺包不存在时，unavailable=true，表示此策略不可用；unavailable=false，表示此策略可用
userId	Integer	用户 id

- 请求示例如下

```
POST /production-lab-api-service/listStrategy HTTP/1.1
```

```
Authorization: Bearer {access_token}
```

```
Content-Type: application/json
```

```
{  
  "userId": 1009810  
}
```

```
{  
  "code": 200,  
  "data": [  
    {  
      "strategyId": "191009810250313001",  
      "strategyName": "策略 1",  
      "applicationId": 10007,  
      "printerId": 10022,  
      "materialId": 10135,  
      "layerThickness": 100,  
      "caseSplitStart": "",  
      "caseSplitEnd": "_",  
      "folderSwitch": "false",  
      "folderDepth": "3",  
      "unavailable": false,  
      "userId": 1009810  
    },  
    {  
      "strategyId": "191009810250313002",  
      "strategyName": "新策略 2",  
      "applicationId": 10007,  
      "printerId": 10012,  
      "materialId": 10072,  
      "layerThickness": 100,  
      "caseSplitStart": "",  
      "caseSplitEnd": "_",  
      "folderSwitch": "false",  
      "folderDepth": "3",  
      "unavailable": true,  
      "userId": 1009810  
    }  
  ]  
}
```

```

    }
  ],
  "message": "成功",
  "detailMessage": null,
  "success": true,
  "error": false
}

```

6.1.2. 查询策略列表 (管理员)

- 功能：获取管理员所在集团(公司)下所有用户在云端保存的策略信息
- 接口地址

请求地址	请求类型
/production-lab-api-service/listStrategyAdmin	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8i bdq

- 请求参数

无

- 响应字段

参数	类型	说明
----	----	----

strategyId	String	策略 Id
strategyName	String	策略名称
applicationId	Integer	应用 Id
printerId	Integer	打印机 Id
materialId	Integer	材料 id
layerThickness	String	层厚
caseSplitStart	String	按 case 排版文件分隔符开始值（生产中心内部使用）
caseSplitEnd	String	按 case 排版文件分隔符结束值（生产中心内部使用）
folderSwitch	String	开启文件夹多层上传的开关。按 case 排版默认为 true.关-false,开-true
folderDepth	String	文件夹的层数（生产中心内部使用）
unavailable	Boolean	如果工艺包不存在时，unavailable=true，表示此策略不可用；unavailable=false，表示此策略可用
userId	Integer	用户 id

- 请求示例如下

```
POST /production-lab-api-service/listStrategyAdmin HTTP/1.1
```

```
Authorization: Bearer {access_token}
```

```
Content-Type: application/json
```

```
{  
}
```

```
{  
  "code": 200,  
  "data": [  
    {  
      "strategyId": "191009810250313001",  
      "strategyName": "策略 1",  
      "applicationId": 10007,  
      "printerId": 10022,  
      "materialId": 10135,  
      "layerThickness": 100,  
      "caseSplitStart": "",  
      "caseSplitEnd": "_",  
      "folderSwitch": "false",  
      "folderDepth": "3",  
      "unavailable": false,  
      "userId": 100523  
    },  
    {  
      "strategyId": "191009810250313002",  
      "strategyName": "新策略 2",  
      "applicationId": 10007,  
      "printerId": 10012,  
      "materialId": 10072,  
      "layerThickness": 100,  
      "caseSplitStart": "",  
      "caseSplitEnd": "_",  
      "folderSwitch": "false",  
      "folderDepth": "3",  
      "unavailable": true,  
      "userId": 100523  
    }  
  ]  
}
```

```

    ],
    "message": "成功",
    "detailMessage": null,
    "success": true,
    "error": false
  }

```

6.2. stl 文件上传

文件上传分多个步骤，需要按步骤进行接口调用实现文件上传。

6.2.1. 获取预授权上传地址

- 功能：获取上传地址
- 接口地址

请求地址	请求类型
/production-lab-api-service/getUploadUrl	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
----	------	------	----

ultMd5	是	String	文件的 md5 值
appld	是	Integer	应用 id, 举例: 10007
userId	否	Integer	用户 id, 举例: 100901

备注:

Java 可以使用 hutool 工具生成文件的 md5 值:

```
public String contentMd5(String filePath) {
    // 计算文件的 MD5 哈希值
    byte[] md5Bytes = DigestUtil.md5(FileUtil.getInputStream(filePath));
    // 对 MD5 字节数组进行转换: xxd -p -r | base64
    return Base64.encode(md5Bytes);
}
```

JavaScript 需要 aws 的 sdk 生成文件的 md5 值:

```
self.importScripts("./aws-sdk-2.828.0.min.js");

// 生成文件 hash
self.onmessage = e => {
    const { fileChunkList } = e.data;
    const reader = new FileReader();
    reader.readAsArrayBuffer(fileChunkList);
    reader.onload = e => {
        var md5 = AWS.util.crypto.md5(e.target.result, "base64")
        self.postMessage({
            percentage: 100,
            hash: md5
        });
    };
};
```

- 响应字段

参数	类型	说明
url	String	预授权的上传地址
urlUuid	String	预授权的流水号
partId	String	模型 id

- 请求示例如下

```
POST /production-lab-api-service/getUploadUrl HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{
  "ultMd5": "3pyrsXLJMqTK+Rn2W9gXPg==",
  "appId": 10007
}

{
  "code": 200,
  "data": {
    "url":
"https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/archive_15/2025/03/19/b5d7796f9e934f61afffc706db7878f8.stl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250319T034225Z&X-Amz-SignedHeaders=content-md5%3Bhost&X-Amz-Expires=28799&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%2F20250319%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=617cd4c04713b3fbf0cc6fed2fc9b24ee185bad1c43535fbf4c2fb481222041c",
    "urlUuid": "202503190342256503549",
    "partId": "1810098102025031900001"
  },
  "message": "成功",
  "detailMessage": null,
  "success": true,
  "error": false
}
```

6.2.2. 文件上传

- 功能：将文件二进制内容上传至上一步返回的预授权 URL。
- 接口地址

请求地址	请求类型
上一步返回的 url	PUT

- 请求头

参数	内容	说明	示例
Content-Type	application/octet-stream	文件 MIME 类型	application/octet-stream
Content-MD5	Base64 编码的 MD5 值	文件内容的 MD5 值 (Base64 编码)	3pyrsXLJMqTK+Rn2W9gxPg==

- 请求参数

文件二进制内容（直接作为 PUT 请求体，不使用 multipart/form-data 封装）

- 响应说明

参数	类型	说明
HTTP 状态码	Integer	上传成功返回 200；响应头包含 ETag，响应体通常为空白

- 请求示例如下

```
PUT
https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/archive_15/2025/03/19/b5d7796f9e934f61afffc706db7878f8.stl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250319T034225Z&X-Amz-SignedHeaders=content-md5%3Bhost&X-Amz-Expires=28799&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%2F20250319%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=617cd4c04713b3fbf0cc6fed2fc9b24ee185bad1c43535fbf4c2fb481222041c HTTP/1.1
```

```
Content-Type: application/octet-stream
Content-MD5: 3pyrsXLJMqTK+Rn2W9gxPg==
【需要上传的文件】
HTTP/1.1 200 OK
ETag: "{etag}"
```

6.2.3. 模型文件下载地址与 partId、batchNo 映射

- 功能：将上传的模型文件 S3FileId，与 partId、batchNo 进行映射
- 接口地址

请求地址	请求类型
/production-lab-api-service/uploadModelFromS3	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。 需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
partMd5	是	String	文件的 md5 值
urlUuid	是	String	S3 文件的 urlUuid
oriFileName	是	String	模型原文件名

partId	是	String	模型 id
userId	是	Integer	用户 Id
userOrgCode	是	Integer	组织 id
fullFilePath	是	String	模型的完整路径
strategyId	是	String	策略 id
importFolder	是	Boolean	是否是文件夹拖拽导入,同一个文件夹默认是同一个 case 的模型。默认传 true
caseId	是	String	文件的 case id/order id 或者文件所在文件夹名称; 无需按 case 排版可传文件名
batchNo	否	String	批次号,不传或者传入空字符串时将会按原逻辑进行处理,不为空时,同批次零件一起排版切片,不会混入其他批次的零件。

- 响应字段

参数	类型	说明
data	Boolean	布尔值表达调用是否成功

- 请求示例如下

```
POST /production-lab-api-service/uploadModelFromS3 HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{
  "caseId": "【1】20201123",
  "fullFilePath": "IT04-2021030009:E:\\文档\\桌面\\user1\\file\\【1】20201123_Salvador_martinez_Tissue_4.stl",
  "importFolder": true,
  "oriFileName": "【1】20201123_Salvador_martinez_Tissue_4.stl",
  "partId": "191221352025031800001",
  "partMd5": "7nu1Ttl/v87uR9TgC7gCFA==",
  "strategyId": "19122135250224001",
  "urlUuid": "202503180637383688609",
  "userId": 122135,
  "userOrgCode": 100302,
  "batchNo": "PN1"
}
{"code":200,"data":true,"message":"成功"}
```

6.2.4. case 完整上传完成确认

- 功能：如果策略的【folderSwitch】参数值为 true 时，上传完了所有模型，需要调用此接口确认。
- 接口地址

请求地址	请求类型
/production-lab-api-service/uploadCompleteConfirm	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意	Bearer w6iupf sv0xbb 4ix8ibd

		Bearer 与 AccessToken 之间存在一个空白字符	q
--	--	---------------------------------	---

• 请求参数

参数	是否必填	参数类型	说明
userId	是	Integer	用户 Id
strategyId	是	String	策略 id
partIdList	是	List<String>	partId 列表
batchConfirm	否	Boolean	批次确认标识。如果需要批次号的业务, 该参数传 true; 不需要批次号的业务, 该参数传 false。该参数传 true 时, batchNo 批次号参数必填。
batchNo	否	String	批次号。如果需要批次号的业务, 该参数传真实的批次号; batchConfirm=true 时, 该参数必填。

• 响应字段

参数	类型	说明
data	Boolean	布尔值表达调用是否成功

• 请求示例如下

```
POST /production-lab-api-service/uploadCompleteConfirm HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{
  "partIdList": [
    "191221352025031800001"
  ],
  "strategyId": "19122135250224001",
  "userId": 122135,
  "batchConfirm": true,
  "batchNo": "1234"
}
{"code":200,"data":true,"message":"成功"}
```

6.3. stl 文件上传（管理员）

文件上传分多个步骤，需要按步骤进行接口调用实现文件上传。

6.3.1. 获取预授权上传地址（管理员）

- 功能：获取入参用户 id 对应的模型文件上传地址
- 接口地址

请求地址	请求类型
/production-lab-api-service/getUploadUrlAdmin	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与	Bearer w6iupfsv0xbb4ix8ibdq

		AccessToken 之间存在一个空白字符	
--	--	------------------------	--

• 请求参数

参数	是否必填	参数类型	说明
ultMd5	是	String	文件的 md5 值
appld	是	Integer	应用 id, 举例: 10007
userId	是	Integer	用户 id, 举例: 100523

备注:

Java 可以使用 hutool 工具生成文件的 md5 值:

```
public String contentMd5(String filePath) {
    // 计算文件的 MD5 哈希值
    byte[] md5Bytes = DigestUtil.md5(FileUtil.getInputStream(filePath));
    // 对 MD5 字节数组进行转换: xxd -p -r | base64
    return Base64.encode(md5Bytes);
}
```

JavaScript 需要 aws 的 sdk 生成文件的 md5 值:

```
self.importScripts("./aws-sdk-2.828.0.min.js");

// 生成文件 hash
self.onmessage = e => {
    const { fileChunkList } = e.data;
    const reader = new FileReader();
    reader.readAsArrayBuffer(fileChunkList);
    reader.onload = e => {
        var md5 = AWS.util.crypto.md5(e.target.result, "base64")
        self.postMessage({
            percentage: 100,
```

```

        hash: md5
    });
};
};

```

- 响应字段

参数	类型	说明
url	String	预授权的上传地址
urlUuid	String	预授权的流水号
partId	String	模型 id

- 请求示例如下

```

POST /production-lab-api-service/getUploadUrlAdmin HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{
  "ultMd5": "3pyrsXLJMqTK+Rn2W9gxPg==",
  "appId": 10007,
  "userId": 122135
}

{
  "code": 200,
  "data": {
    "url":
"https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/archive_15/2025/03/19/b5d7796f9e934f61afffc706db7878f8.stl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250319T034225Z&X-Amz-SignedHeaders=content-md5%3Bhost&X-Amz-Expires=28799&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%2F20250319%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=617cd4c04713b3fbf0cc6fed2fc9b24ee185bad1c43535fbf4c2fb481222041c",
    "urlUuid": "202503190342256503549",
    "partId": "1810098102025031900001"
  },
  "message": "成功",
  "detailMessage": null,

```

```
"success": true,
"error": false
}
```

6.3.2. 文件上传（管理员）

- 功能：将文件二进制内容上传至上一步返回的预授权 URL。
- 接口地址

请求地址	请求类型
上一步返回的 url	PUT

- 请求头

参数	内容	说明	示例
Content-Type	application/octet-stream	文件 MIME 类型	application/octet-stream
Content-MD5	Base64 编码的 MD5 值	文件内容的 MD5 值 (Base64 编码)	3pyrsXLJMqTK+Rn2W9gxPg==

- 请求参数

文件二进制内容（直接作为 PUT 请求体，不使用 multipart/form-data 封装）

- 响应说明

参数	类型	说明
HTTP 状态码	Integer	上传成功返回 200；响应头包含 ETag，响应体通常为空白

- 请求示例如下

PUT https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/archive_15/2025/03/19/b5d7796f9e934f61afffc706db7878f8.stl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250319T034225Z&X-Amz-SignedHeaders=content-md5%3Bhost&X-Amz-Expires=28799&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLWn%2F20250319%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=617cd4c04713b3fbf0cc6fed2fc9b24ee185bad1c43535fbf4c2fb481222041c HTTP/1.1 Content-Type: application/octet-stream Content-MD5: 3pyrsXLJMqTK+Rn2W9gxPg== 【需要上传的文件】
HTTP/1.1 200 OK ETag: "{etag}"

6.3.3. 模型文件下载地址与 partId、batchNo 映射（管理员）

- 功能：将入参用户上传的模型文件 S3FileId，与 partId、batchNo 进行映射
- 接口地址

请求地址	请求类型
/production-lab-api-service/uploadModelFromS3Admin	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。 需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
partMd5	是	String	文件的 md5 值
urlUuid	是	String	S3 文件的 urlUuid
oriFileName	是	String	模型原文件名
partId	是	String	模型 id
userId	是	Integer	用户 id
userOrgCode	是	Integer	组织 id
fullFilePath	是	String	模型的完整路径
strategyId	是	String	策略 id
importFolder	是	Boolean	是否是文件夹拖拽导入,同一个文件夹默认是同一个 case 的模型。默认传 true
caseId	是	String	文件的 case id/order id 或者文件所在文件夹名称; 无需按 case 排版可传文件名
batchNo	否	String	批次号,不传或者传入空字符串时将会按原逻辑进行处理,不为空时,同批次零件一起排版切片,不会混入其他批次的零件。

- 响应字段

参数	类型	说明
data	Boolean	布尔值表达调用是否成功

- 请求示例如下

POST /production-lab-api-service/uploadModelFromS3Admin HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json { "caseId": " 【1】 20201123", "fullFilePath": "IT04-2021030009:E:\\文档\\桌面\\user1\\file\\ 【1】 20201123_Salvador_martinez_Tissue_4.stl", "importFolder": true, "oriFileName": " 【1】 20201123_Salvador_martinez_Tissue_4.stl", "partId": "191221352025031800001", "partMd5": "7nu1Ttl/v87uR9TgC7gCFA==", "strategyId": "19122135250224001", "urlUuid": "202503180637383688609", "userId": 122135, "userOrgCode": 100302, "batchNo": "PN1" } {"code":200,"data":true,"message":"成功"}
--

6.3.4. case 完整上传完成确认（管理员）

- 功能：如果策略的【folderSwitch】参数值为 true 时，上传完了所有模型，需要调用此接口确认。
- 接口地址

请求地址	请求类型
/production-lab-api-service/uploadCompleteConfirmAdmin	POST

- 请求头

参数	内容	说明	示例
----	----	----	----

Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Beare r w6iupf sv0xbb 4ix8ibd q
---------------	-------------------------	--	--

• 请求参数

参数	是否必填	参数类型	说明
userId	是	Integer	用户 Id
strategyId	是	String	策略 id
partIdList	是	List<String>	partId 列表
batchConfirm	否	Boolean	批次确认标识。如果需要批次号的业务，该参数传 true；不需要批次号的业务，该参数传 false。该参数传 true 时，batchNo 批次号参数必填。
batchNo	否	String	批次号。如果需要批次号的业务，该参数传真实的批次号；batchConfirm=true 时，该参数必填。

• 响应字段

参数	类型	说明
data	Boolean	布尔值表达调用是否

		成功
--	--	----

- 请求示例如下

POST /production-lab-api-service/uploadCompleteConfirmAdmin HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json { "partIdList": ["191221352025031800001"], "strategyId": "19122135250224001", "userId": 122135, "batchConfirm": true, "batchNo": "1234" }
{ "code": 200, "data": true, "message": "成功" }

6.4. 零件库模型上传

文件上传分 3 个步骤，需要按步骤进行接口调用实现文件上传。

6.4.1. 获取预授权上传地址

- 功能：获取上传地址
- 接口地址

请求地址	请求类型
/production-lab-partlib/getUploadUrl	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与	Bearer w6iupfsv0xbb4ix8ibdq

		AccessToken 之间存在一个空白字符; 需要开通【PARTLIB】的请求权限: scope=PARTLIB (详细内容请参考文档 2.2.1)	
--	--	---	--

• 请求参数

参数	是否必填	参数类型	说明
libPartMd5	是	String	模型文件的 md5 值

备注:

Java 可以使用 hutool 工具生成文件的 md5 值:

```
public String contentMd5(String filePath) {
    // 计算文件的 MD5 哈希值
    byte[] md5Bytes = DigestUtil.md5(FileUtil.getInputStream(filePath));
    // 对 MD5 字节数组进行转换: xxd -p -r | base64
    return Base64.encode(md5Bytes);
}
```

JavaScript 需要 aws 的 sdk 生成文件的 md5 值:

```
self.importScripts("./aws-sdk-2.828.0.min.js");

// 生成文件 hash
self.onmessage = e => {
    const { fileChunkList } = e.data;
    const reader = new FileReader();
    reader.readAsArrayBuffer(fileChunkList);
    reader.onload = e => {
        var md5 = AWS.util.crypto.md5(e.target.result, "base64")
        self.postMessage({
```

```
percentage: 100,
hash: md5
});
};
};
```

• 响应字段

参数	类型	说明
url	String	预授权的上传地址
urlUuid	String	预授权的流水号
libPartId	String	库模型 id

• 请求示例如下

```
POST /production-lab-partlib/getUploadUrl HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json

{
  "libPartMd5": "83rmQunnNdam580o0I214Q=="
}

{
  "code": 200,
  "data": {
    "url":
      "https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/archive_15/2025/11/24/983da860492d4def83d16d763363fde2.stl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20251124T082715Z&X-Amz-SignedHeaders=content-md5%3Bhost&X-Amz-Expires=28799&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%2F20251124%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=3173f663294e284983f9c757728e5e210d869bc2d8cd184c73e049a1eed243c5",
    "urlUuid": "202511240827155799720",
    "libPartId": "10020625112400002"
  },
  "message": "成功",
```

```

"detailMessage": null,
"success": true,
"error": false
}

```

6.4.2. 文件上传

- 功能：将文件二进制内容上传至上一步返回的预授权 URL。
- 接口地址

请求地址	请求类型
上一步返回的 url	PUT

- 请求头

参数	内容	说明	示例
Content-Type	application/octet-stream	文件 MIME 类型	application/octet-stream
Content-MD5	Base64 编码的 MD5 值	文件内容的 MD5 值 (Base64 编码)	3pyrsXLJMqTK+Rn2W9gxPg==

- 请求参数

文件二进制内容（直接作为 PUT 请求体，不使用 multipart/form-data 封装）

- 响应说明

参数	类型	说明
HTTP 状态码	Integer	上传成功返回 200；响应头包含 ETag，响应体通常

		为空
--	--	----

- 请求示例如下

PUT https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/archive_15/2025/03/19/b5d7796f9e934f61afffc706db7878f8.stl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250319T034225Z&X-Amz-SignedHeaders=content-md5%3Bhost&X-Amz-Expires=28799&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%2F20250319%2Foss-cn-shenzhen%2Faws4_request&X-Amz-Signature=617cd4c04713b3fbf0cc6fed2fc9b24ee185bad1c43535fbf4c2fb481222041c HTTP/1.1 Content-Type: application/octet-stream Content-MD5: 3pyrsXLJMqTK+Rn2W9gxPg== 【需要上传的文件】
HTTP/1.1 200 OK ETag: "{etag}"

6.4.3. 模型文件完成上传

- 功能：将上传的模型文件 S3FileId，与 libPartId 进行映射
- 接口地址

请求地址	请求类型
/production-lab-partlib/completeUpload	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。 需注意 Bearer 与 AccessToken 之间存在一个空白字符； 需要开通【PARTLIB】的请求权限：scope=PARTLIB（详细内容请	Bearer w6iupfsv0xbb4i x8ibdq

		参考文档 2.2.1)	
--	--	-------------	--

• 请求参数

参数	是否必填	参数类型	说明
libPartMd5	是	String	文件的 md5 值
urlUuid	是	String	S3 文件的 urlUuid
libPartName	是	String	模型原文件名
libPartId	是	String	库模型的 id
appType	否	String	设计应用类型, 指: 代型牙模、种植牙模、正畸牙模等。例如: 代型牙模 传 10007 种植牙模 传 10030 复合牙模 传 1000701 正畸牙模 传 1000901 正畸+CNC 传 1000902 临时冠桥 传 10002 颌垫 传 10011 活动义齿基托 传 10000 弹性义齿基托 传 10038 试戴义齿 传 10014 活动义齿上部牙 传 10021 种植导板 传 10010

			牙龈胶 传 10015 蜡冠 传 10031 蜡冠支架 传 10001 托盘 传 10013 透明基托 传 10023 给药托盘牙模 传 1000903 通用应用 传 10024 MMF 通用 传 19999
--	--	--	---

- 响应字段

参数	类型	说明
data	Boolean	布尔值表达调用是否成功

- 请求示例如下

POST /production-lab-partlib/completeUpload HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json { "uuid": "202511240827155799720", "libPartName": "221_Tooth_31.stl", "libPartId": "10020625112400002", "libPartMd5": "dBnV/Iy9xymu/UgLpbSVGw==", "appType": "10007" }
{ "code": 200, "data": true, }

```

"message": "成功",
"detailMessage": null,
"success": true,
"error": false
}

```

7. 设计中心接口

7.1. 查询订单列表(管理员--组织下全量数据)

- 功能：查询订单列表信息
- 接口地址

请求地址	请求类型
/design-basic-service/openApi/v1/getOrderPageList	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。 需注意 Bearer 与 AccessToken 之间存在一个空白字符； 需要开通【DESIGN】的请求权限：scope=DESIGN（详细内容请参考文档 2.2.1）	Bearer w6iupfsv0xbb4i x8ibdq

- 请求参数

参数	是否必填	参数类型	说明
----	------	------	----

startTime	是	Integer(int64)	创建时间-开始时间戳
endTime	是	Integer(int64)	创建时间-结束时间戳
pageNo	是	Integer	当前页码 (默认第 1 页)
pageSize	是	Integer	每页数量 (默认 20 条, 最大 100 条)
orderStatus	否	Integer	订单状态(1:新建 2:设计中 3:待确认 4:已退回 5:已完成 6:已取消 7:审核中 8:已买单)

- 响应字段

参数	类型	说明
createdTime	Integer(int64)	订单创建时间戳
currency	Integer	0: 美元; 1: 人民币; 2: 欧元; 3: 日元; 4: 澳元
prices	Number	预计费用
deliveryTime	Integer(int64)	期望完成时间戳
completedTime	Integer(int64)	订单实际完成时间戳
designCategory	String	设计大类, 多个类, 逗号隔开 21000:固定修复 22000:活动修复 23000:种植修复

		24000: 正畸&其他
isUnion	Boolean	是否联合修复类型: true 是; false 否
orderCode	Integer(int64)	订单编号唯一标识
orderFileName	String	原始订单文件名称
orderNo	String	原订单号
orderStatus	Integer	订单状态 (1:新建 2: 设计中 3:待确认 4: 已退回 5:已完成 6: 已取消 7:审核中 8: 已免单)
toothDesign	String	牙位信息存储设计类 型和牙位
pageNo	Integer	返回结果中的第几页
pageSize	Integer	返回结果中每页显示的订 单数量
totalSize	Integer	总条数
totalPages	Integer	总页数

- 请求示例如下

```
POST /design-basic-service/openApi/v1/getOrderPageList HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{
  "endTime": 1769788799000,
```

<pre> "pageNo": 1, "pageSize": 10, "startTime": 1767196800000 } </pre>
<pre> { "code": 200, "data": { "pageNo": 1, "pageSize": 10, "totalSize": 97, "totalPages": 10, "data": [{ "orderCode": "249343841152416768", "orderNo": "Q-100765-260130-9494", "createdTime": 1769764470744, "deliveryTime": 1769850870744, "orderFileName": "16533159.zip", "orderStatus": 1, "completedTime": 0, "prices": 2.8, "designCategory": "21000", "isUnion": false, "toothDesign": "[\{"code":21104,\"enName\":\"Inlay/Onlay\", \"pidCode\":21100, \"pidEnName\":\"Anatomy\", \"pidZhName\":\"解剖冠\", \"tooth\":[16], \"zhName\":\"嵌体/高嵌体\"]]", "currency": 0 }, { "orderCode": "249319189449423872", "orderNo": "Q-100765-260130-7443", "createdTime": 1769740961053, "deliveryTime": 1769827361053, "orderFileName": "18160065.zip", "orderStatus": 1, "completedTime": 0, "prices": 2.8, "designCategory": "21000", </pre>

```

    "isUnion": false,
    "toothDesign": "[{"code":21101,"enName":"Full
Crown","pidCode":21100,"pidEnName":"Anatomy","pidZhName":"解剖冠
","tooth":[26],"zhName":"全冠"}]",
    "currency": 0
  },
  {
    "orderCode": "249239754831454208",
    "orderNo": "10076532480431",
    "createdTime": 1769665206270,
    "deliveryTime": 1769821200000,
    "orderFileName": "A3-LaFRANCE_FREMA.AA.zip",
    "orderStatus": 2,
    "completedTime": 0,
    "prices": 5.6,
    "designCategory": "21000",
    "isUnion": false,
    "toothDesign": "[{"code":21101,"enName":"Full
Crown","pidCode":21100,"pidEnName":"","pidZhName":"","tooth":[27,28],"
zhName":"全冠"}]",
    "currency": 0
  },
  {
    "orderCode": "249239437058400256",
    "orderNo": "10076574085431",
    "createdTime": 1769664900705,
    "deliveryTime": 1769821200000,
    "orderFileName": "2749929-mmp_Smith.zip",
    "orderStatus": 2,
    "completedTime": 0,
    "prices": 22.4,
    "designCategory": "21000",
    "isUnion": false,
    "toothDesign": "[{"code":21101,"enName":"Full
Crown","pidCode":21100,"pidEnName":"Anatomy","pidZhName":"解剖冠
","tooth":[33,21,43,28],"zhName":"全冠
"}, {"code":21501,"enName":"Connector","pidCode":21500,"pidEnName":"Bridg
e","pidZhName":"牙桥","tooth":[32,33,41,42,43,31],"zhName":"连接体
"}, {"code":21106,"enName":"Crown

```

```
Pontic\", \"pidCode\": 21100, \"pidEnName\": \"Anatomy\", \"pidZhName\": \"解剖冠\", \"tooth\": [32, 41, 42, 31], \"zhName\": \"牙冠缺失位\"}],
  \"currency\": 0
},
{
  \"orderCode\": \"249159489205460992\",
  \"orderNo\": \"10076588965281\",
  \"createdTime\": 1769588658998,
  \"deliveryTime\": 1769754258998,
  \"orderFileName\": \"2750343-RV_Works (1).zip\",
  \"orderStatus\": 1,
  \"completedTime\": 0,
  \"prices\": 22.1,
  \"designCategory\": \"22000\",
  \"isUnion\": false,
  \"toothDesign\": \"[{\"code\": 22201, \"enName\": \"Denture≥7(Artificial Tooth)\", \"pidCode\": 22200, \"pidEnName\": \"\", \"pidZhName\": \"\", \"tooth\": [21, 22, 23, 24, 25, 26, 27, 28], \"zhName\": \"活动义齿≥7(成品牙)\"]\",
  \"currency\": 0
},
{
  \"orderCode\": \"249159416886222848\",
  \"orderNo\": \"10076500065491\",
  \"createdTime\": 1769588589572,
  \"deliveryTime\": 1769754189572,
  \"orderFileName\": \"2750343-RV_Works (1).zip\",
  \"orderStatus\": 5,
  \"completedTime\": 1769669395061,
  \"prices\": 3,
  \"designCategory\": \"22000\",
  \"isUnion\": false,
  \"toothDesign\": \"[{\"code\": 22205, \"enName\": \"Full Denture(MMF)\", \"pidCode\": 22200, \"pidEnName\": \"\", \"pidZhName\": \"\", \"tooth\": [21, 22, 23, 24, 25, 26, 27, 28], \"zhName\": \"全口义齿 (MMF)\"]\",
  \"currency\": 0
},
{
  \"orderCode\": \"24869268886664448\",
```

```

"orderNo": "1649498",
"createdTime": 1769143483552,
"deliveryTime": 1769229883552,
"orderFileName": "CARDONA_TULIA Pur514_Tooth_36 (1) (1) (1).stl",
"orderStatus": 4,
"completedTime": 0,
"prices": 0,
"designCategory": "21000",
"isUnion": false,
"toothDesign":
"[{"code":25001,"enName":"Unknown","tooth":[],"zhName":"未知"}]",
"currency": 0
},
{
"orderCode": "248688954825662464",
"orderNo": "6736339",
"createdTime": 1769139922476,
"deliveryTime": 1769226322476,
"orderFileName": "CARDONA_TULIA Pur514_Tooth_36 (1) (2).stl",
"orderStatus": 1,
"completedTime": 0,
"prices": 0,
"designCategory": "21000",
"isUnion": false,
"toothDesign":
"[{"code":25001,"enName":"Unknown","tooth":[],"zhName":"未知"}]",
"currency": 0
},
{
"orderCode": "248687362871287808",
"orderNo": "10076573699341",
"createdTime": 1769138404279,
"deliveryTime": 1769224804279,
"orderFileName": "CARDONA_TULIA Pur514_Tooth_36 (1) (1) (1).stl",
"orderStatus": 2,
"completedTime": 0,
"prices": 2.48,
"designCategory": "21000",

```

```

    "isUnion": false,
    "toothDesign":
    "[{"code":21201,"enName":"Coping","pidCode":21200,"pidEnName":"","pidZhName":"","tooth":28,"zhName":"内冠"}]",
    "currency": 0
  },
  {
    "orderCode": "248679232766631936",
    "orderNo": "10076569229191",
    "createdTime": 1769130650766,
    "deliveryTime": 1769217050766,
    "orderFileName": "01b73d0d42c74c2c96f06ef594f5d2d9.stl",
    "orderStatus": 1,
    "completedTime": 0,
    "prices": 2.8,
    "designCategory": "21000",
    "isUnion": false,
    "toothDesign": "[{"code":21101,"enName":"Full Crown","pidCode":21100,"pidEnName":"","pidZhName":"","tooth":27,"zhName":"全冠"}]",
    "currency": 0
  }
]
},
"message": "成功",
"detailMessage": null,
"error": false,
"success": true
}

```

7.2. 查看订单详情 (通过订单列表 orderCode 查询)

- 接口地址

请求地址	请求类型
------	------

/design-basic-service/openApi/v1/orderInfo? orderCode={订单编号}	GET
---	-----

• 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。 需注意 Bearer 与 AccessToken 之间存在一个空白字符; 需要开通【DESIGN】的请求权限: scope=DESIGN (详细内容请参考文档 2.2.1)	Bearer w6iupfsv0xbb4i x8ibdq

• 请求参数

参数	是否必填	参数类型	说明
orderCode	是	Integer(int64)	订单编号唯一标识

• 响应字段

参数	类型	说明
orderFiles	Array of orderFiles	订单文件集合
orderFiles.fileName	String	文件名
orderFiles.fileType	Integer	文件类型 1: 原始文件 2: 设计文件 3: 设计 STL 4: 设计截图 5: 其它

		6: html
orderFiles.s3FileId	String	s3FileId
orderStatus	Integer	订单状态(1:新建 2:设计中 3:待确认 4:已退回 5:已完成 6:已取消 7:审核中 8:已免单)
orderNo	String	原订单号
toothImage	String	牙位图 s3FileId
toothDesign	String	牙位信息存储设计类型和牙位
completedTime	Integer(int64)	实际完成时间戳
createdTime	Integer(int64)	创建时间戳
designCompletedTime	Integer(int64)	设计完成时间戳
currency	Integer	币种 0: 美元; 1: 人民币; 2: 欧元; 3: 日元; 4: 澳元
deliveryTime	Integer(int64)	期望完成时间戳
designCategory	String	设计大类, 多个使用逗号隔开 21000:固定修复 22000:活动修复 23000:种植修复

		24000: 正畸&其他
designSoftware	Integer	设计软件 1001: 3Shape 1002: Exocad
softwareVersion	Integer	软件版本 3Shape 对应的版本 100101: 2019 100102: 2020 100103: 2021 100104: 2022 100105: 2023 100106: 2024 Exocad 对应的版本 100201: 3.0 100202: 3.1
prices	Number	预计费用
remark	String	其他约束
isFast	Boolean	是否极速建单: true 是; false 否
isUnion	Boolean	是否联合修复类型: true 是; false 否

- 请求示例如下

```
GET /design-basic-service/openApi/v1/orderInfo?orderCode=247984383479405568 HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
```

```
{
  "code": 200,
  "data": {
    "orderCode": "247984383479405568",
    "orderNo": "10076539544931",
    "orderStatus": 5,
    "remark": "",
    "toothDesign": "[{\"code\":\"21404\",\"enName\":\"Contact Model\",\"pidCode\":\"21400\",\"pidEnName\":\"\",\"pidZhName\":\"\",\"tooth\":[41,11],\"zhName\":\"接触模\"},{\"code\":\"21401\",\"enName\":\"Sectioned Model\",\"pidCode\":\"21400\",\"pidEnName\":\"\",\"pidZhName\":\"\",\"tooth\":[27,28],\"zhName\":\"分割代型牙模\"}]",
    "toothImage": "{\"21000\":\"808fe6271cc44b3a9e59c321d3089cd3\"}",
    "orderFiles": [
      {
        "s3FileId": "26b0e8a834a646deb913932cb91a9813",
        "fileName": "CARDONA_TULIA Pur514_Tooth_36 (1) (2).stl",
        "fileType": 1
      },
      {
        "s3FileId": "f4ef867a26f846a681e516d76849ba07",
        "fileName": "25953_20240325_0738_Zartman.zip",
        "fileType": 2
      },
      {
        "s3FileId": "ea7993bbe54d48528f76bb090e6286dc",
        "fileName": "0277152a344f4b7c96f491dcc9297ee1.stl",
        "fileType": 3
      },
      {
        "s3FileId": "a409947bf26344e69d9226c5331001b6",
        "fileName": "0023407b2d584a53a84ffdbbeb368f7f.png",
        "fileType": 4
      }
    ],
    "prices": 2.08,
    "deliveryTime": 1768611600000,
    "completedTime": 1768468283624,
  }
}
```

```

    "designCompletedTime": 1768468283624,
    "createdTime": 1768467990830,
    "isFast": false,
    "isUnion": false,
    "designSoftware": 1001,
    "softwareVersion": 100101,
    "designCategory": "21000",
    "currency": 0
  },
  "message": "成功",
  "detailMessage": null,
  "error": false,
  "success": true
}

```

7.3. 获取原始文件预授权上传地址

- 功能：获取上传地址
- 接口地址

请求地址	请求类型
/design-file-service/platformS3/v1/getUploadUrl	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
contentMd5	是	String	文件的 md5 值
expirationInDays	是	Integer	文件存储的过期时间，默认是 90 天举例：90
isAccelerate	否	Boolean	是否使用全球加速模式: true 是 false 否

备注:

Java 可以使用 hutool 工具生成文件的 md5 值:

```
public String contentMd5(String filePath) {
    // 计算文件的 MD5 哈希值
    byte[] md5Bytes = DigestUtil.md5(FileUtil.getInputStream(filePath));
    // 对 MD5 字节数组进行转换: xxd -p -r | base64
    return Base64.encode(md5Bytes);
}
```

JavaScript 需要 aws 的 sdk 生成文件的 md5 值:

```
self.importScripts("./aws-sdk-2.828.0.min.js");

// 生成文件 hash
self.onmessage = e => {
    const { fileChunkList } = e.data;
    const reader = new FileReader();
    reader.readAsArrayBuffer(fileChunkList);
    reader.onload = e => {
        var md5 = AWS.util.crypto.md5(e.target.result, "base64")
        self.postMessage({
            percentage: 100,
            hash: md5
        })
    }
}
```

```
});
};
};
```

- 响应字段

参数	类型	说明
url	String	预授权的上传地址
urlUuid	String	预授权的流水号
contentMd5	String	文件的 MD5 值

- 请求示例如下

```
POST /design-file-service/platformS3/v1/getUploadUrl HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{"contentMd5":"rpYXP8awAmfHy+CcctqwsA==","objectSuffix":"png","isAccelerate":false,"expirationInDays":90}

{
  "code": 200,
  "data": {
    "url":
"https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/normal/2026/01/23/0272226545374afdb48b709c35501902.png?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20260123T032003Z&X-Amz-SignedHeaders=content-md5%3Bhost&X-Amz-Expires=3600&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%2F20260123%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=bd1618847cf3588f10c1f344c14a2b355d35eb058fc18bd6f4e04ac1edf2cab8",
    "contentMd5": "rpYXP8awAmfHy+CcctqwsA==",
    "urlUuid": "202601230320039612493"
  },
  "message": "成功",
  "detailMessage": null,
  "success": true,
  "error": false
}
```

7.4. 文件上传

- 功能：将文件二进制内容上传至上一步返回的预授权 URL。
- 接口地址

请求地址	请求类型
上一步返回的 url	PUT

- 请求头

参数	内容	说明	示例
Content-Type	application/octet-stream	文件 MIME 类型	application/octet-stream
Content-MD5	Base64 编码的 MD5 值	文件内容的 MD5 值 (Base64 编码)	3pyrsXLJMqTK+Rn2W9gxPg==

- 请求参数

文件二进制内容（直接作为 PUT 请求体，不使用 multipart/form-data 封装）

- 响应说明

参数	类型	说明
HTTP 状态码	Integer	上传成功返回 200；响应头包含 ETag，响应体通常为空白

- 请求示例如下

PUT

```
https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com
/archive_15/2025/03/19/b5d7796f9e934f61afffc706db7878f8.stl?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250319T034225Z&X-Amz-SignedHeaders=content
-md5%3Bhost&X-Amz-Expires=28799&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%
2F20250319%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=617cd4c04
713b3fbf0cc6fed2fc9b24ee185bad1c43535fbf4c2fb481222041c HTTP/1.1
Content-Type: application/octet-stream
Content-MD5: 3pyrsXLJMqTK+Rn2W9gxPg==
【需要上传的文件】
HTTP/1.1 200 OK
ETag: "{etag}"
```

7.5. 文件完成上传

- 功能：将上传的模型文件返回 S3FileId
- 接口地址

请求地址	请求类型
/design-file-service/platformS3/v1/completeUpload	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
----	------	------	----

urlUuid	是	String	预授权的流水号
isAccelerate	否	Boolean	是否使用全球加速模式: true 是 false 否

• 响应字段

参数	类型	说明
s3FileId	String	文件 s3FileId
url	String	文件下载 url
contentMd5	String	文件的 MD5 值

• 请求示例如下

POST /design-file-service/platformS3/v1/completeUpload HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json {"urlUuid":"202601230320039612493","isAccelerate":false}
<pre>{ "code": 200, "data": { "s3FileId": "b686147204474ff7b5ede55b6cd94904", "url": "https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/normal/2026/01/23/0272226545374afdb48b709c35501902.png?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20260123T032004Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%2F20260123%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=823f1fcc5b2e5fea1b9c2578db3f04fcb1b538416a5c3d19ae3874de2d3226d0", "contentMd5": "rpYXP8awAmfHy+CcctqwsA==" }, "message": "成功", "detailMessage": null, "success": true, "error": false }</pre>

}

7.6. 获取 S3 的下载预授权

- 功能：根据 S3FileId 下载文件
- 接口地址

请求地址	请求类型
/design-file-service/platformS3/v3/getDownloadUrl	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
s3FileId	是	String	S3 文件 ID
filename	否	String	自定义下载文件名,不包含后缀,可以为空
isAccelerate	否	Boolean	是否使用全球加速模式: true 是 false 否

- 响应字段

参数	类型	说明
url	String	文件下载 url
contentMd5	String	文件 md5 值
suffix	String	文件后缀

- 请求示例如下

```
POST /design-file-service/platformS3/v3/getDownloadUrl HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{"s3FileId":"3f26bbb403c640bd8203661ac16ecaee","isAccelerate":false}

{
  "code": 200,
  "data": {
    "url":
"https://dev-aly-design-center-bucket-shenzhen.oss-cn-shenzhen.aliyuncs.com/normal/2025/11/12/9236732bd4ff489b817174ec5bd64a09.png?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20260123T034123Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=LTAI5tFUY76jKm8tCe64LLwN%2F20260123%2Foss-cn-shenzhen%2Fs3%2Faws4_request&X-Amz-Signature=d490a3011a906261e8b576aebb61339bc3e84bf1cdbe3f9383dfa3dac2a37873",
    "contentMd5": "bAApuPwlpm0KimXgbLDN7w==",
    "suffix": "png"
  },
  "message": "成功",
  "detailMessage": null,
  "success": true,
  "error": false
}
```

7.7. 极速建单

- 功能：上传原始文件快速创建订单

第 100 页 共 134 页

- 接口地址

请求地址	请求类型
/design-basic-service/openApi/v1/addOrderByFast	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
designCategory	是	String	设计大类编码 21000:固定修复 22000:活动修复 23000:种植修复 24000:正畸&其他
designSoftware	是	Integer	设计软件 1001: 3Shape 1002: Exocad
designVersion	是	Integer	软件版本 3Shape 对应的版本 100101:2019

			100102:2020 100103:2021 100104:2022 100105:2023 100106:2024 Exocad 对应的版本 100201:3.0 100202:3.1
fileName	是	String	原始文件名称
s3FileId	是	String	文件上传返回的 s3FileId
orderNo	是	String	原订单号
delivery	是	Integer	交期: 24 或 2

• 响应字段

参数	类型	说明
data	Boolean	布尔值表达调用是否成功

• 请求示例如下

POST /design-basic-service/openApi/v1/addOrderByFast HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json {"delivery":24,"designCategory":21000,"fileName":"2750739_Z_Postell.zip","s3FileId": "63b7f5abe1c5443cb8396ca51e310f5c","orderNo":"06289111","designSoftware":1001,"desig nVersion":100101}
{ "code": 200, "data": true, "message": "成功", }

```

    "detailMessage": null,
    "success": true,
    "error": false
  }

```

7.8. 设计中心订单消息(消息订阅使用)

- 接口说明

当订单状态变更为“设计中”、“待确认”、“已退回”或“已免单”时，发送订阅订单消息。

- 订单状态说明

状态名称	状态含义	触发动作/前置条件	后续状态
新建	客户已创建订单,但尚未指定设计师或未开始设计工作。	客户提交订单	设计师接手订单 → 设计中
设计中	设计师正在对订单进行方案设计	设计师“开始设计”订单。	设计完成 → 待确认 客户要求修改/不符合要求 → 已退回
待确认	设计已完成,等待客户最终验收确认。	设计师点击“完成设计”。	客户确认无误 → 已完成 客户返单 → 设计中 客户申请免单 →

状态名称	状态含义	触发动作/前置条件	后续状态
			审核中
已退回	设计师认为设计无法继续或需客户重新沟通，将订单退回给客户。	设计师在“设计中”状态点击“退回”。	客户重新考虑 → 设计中 (通常需重新编辑) 客户放弃 → 已取消
已取消	订单在未完成前被客户主动终止。	客户在“已退回”状态“取消订单”。	流程终止（终态）
已完成	客户对设计满意，确认订单完成。	客户在“待确认”状态点击“确认订单”。	流程结束（终态）
审核中	客户发起了免单申请，订单进入后台审核流程。	客户在“待确认”状态点击“申请免单”。	审核通过 → 已免单 审核驳回 → 待确认
已免单	免单申请已通过审核，订单金额减免。	后台管理员在“审核中”状态点击“免单通过”。	流程结束（特殊终态）

- 请求说明

GET /mq/http/peekMessage?queue=design.order.info

Authorization: Bearer {access_token}

- 请求参数

参数	位置	必填	类型	说明
queue	Query	是	String	队列名称 design.order.info
Authorization	Header	是	String	Bearer token, 格式: Bearer {token}

- payload 响应字段

参数	类型	说明
orderFiles	Array of orderFiles	订单文件集合
orderFiles.fileName	String	文件名
orderFiles.fileType	Integer	文件类型 1: 原始文件 2: 设计文件 3: 设计 STL 4: 设计截图 5: 其它 6: html
orderFiles.s3FileId	String	s3FileId
orderStatus	Integer	订单状态 (1:新建 2:设计中 3:待确认 4:已退回 5:已完成 6:已取消 7:审核中 8:已免单)
orderNo	String	原订单号
toothImage	String	牙位图 s3FileId
toothDesign	String	牙位信息存储设计类型和牙位

completedTime	Integer(int64)	实际完成时间戳
designCompletedTime	Integer(int64)	设计完成时间戳
createdTime	Integer(int64)	创建时间戳
currency	Integer	币种 0: 美元; 1: 人民币; 2: 欧元; 3: 日元; 4: 澳元
deliveryTime	Integer(int64)	期望完成时间戳
designCategory	String	设计大类，多个使用逗号隔开 21000:固定修复 22000:活动修复 23000:种植修复 24000:正畸&其他
designSoftware	Integer	设计软件 1001: 3Shape 1002: Exocad
softwareVersion	Integer	软件版本 3Shape 对应的版本 100101: 2019 100102: 2020 100103: 2021 100104: 2022 100105: 2023 100106: 2024 Exocad 对应的版本 100201: 3.0 100202: 3.1

prices	Number	预计费用
remark	String	其他约束
isFast	Boolean	是否极速建单: true 是; false 否
isUnion	Boolean	是否联合修复类型: true 是; false 否

- 响应示例（成功）

GET /mq/http/peekMessage?queue=design.order.info HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json
<pre>{ "payload": { "orderCode": "247984383479405568", "orderNo": "10076539544931", "orderStatus": 5, "remark": "", "toothDesign": "[{\"code\":\"21404\",\"enName\":\"Contact Model\",\"pidCode\":\"21400\",\"pidEnName\":\"\",\"pidZhName\":\"\",\"tooth\":[41,11],\"zhName\":\"接触模\"},{\"code\":\"21401\",\"enName\":\"Sectioned Model\",\"pidCode\":\"21400\",\"pidEnName\":\"\",\"pidZhName\":\"\",\"tooth\":[27,28],\"zhName\":\"分割代型牙模\"}]", "toothImage": "{\"21000\":\"808fe6271cc44b3a9e59c321d3089cd3\"}", "orderFiles": [{ "s3FileId": "26b0e8a834a646deb913932cb91a9813", "fileName": "CARDONA_TULIA Pur514_Tooth_36 (1) (2).stl", "fileType": 1 }, { </pre>

7.9

```
"s3FileId": "f4ef867a26f846a681e516d76849ba07",
"fileName": "25953_20240325_0738_Zartman.zip",
"fileType": 2
},
{
  "s3FileId": "ea7993bbe54d48528f76bb090e6286dc",
  "fileName": "0277152a344f4b7c96f491dcc9297ee1.stl",
  "fileType": 3
},
{
  "s3FileId": "a409947bf26344e69d9226c5331001b6",
  "fileName": "0023407b2d584a53a84ffdbbeb368f7f.png",
  "fileType": 4
}
],
"prices": 2.08,
"deliveryTime": 1768611600000,
"completedTime": 1768468283624,
"designCompletedTime": 1768468283624,
"createdTime": 1768467990830,
"isFast": false,
"isUnion": false,
"designSoftware": 1001,
"softwareVersion": 100101,
"designCategory": "21000",
"currency": 0
},
"routing_key": "design.order.info"
}
```

息

- 接口地址

请求地址	请求类型
/design-basic-service/openApi/v1/getDesignTypes	GET

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数
无
- 响应字段

参数	类型	说明
designCode	Integer	设计类型编码
parentCode	Integer	设计类型父级编码
zhName	String	设计类型中文名称
enName	String	设计类型英文名称
level	Integer	设计类型等级 1/2/3 级
children	List<DesignType>	设计类型子集

- 请求示例如下

GET /design-basic-service/openApi/v1/getDesignTypes HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json
{ "code": 200, "data": [

```
{
  "designCode": 21000,
  "parentCode": 0,
  "zhName": "固定修复",
  "enName": "Fixed Restoration",
  "level": 1,
  "children": [
    {
      "designCode": 21100,
      "parentCode": 21000,
      "zhName": "解剖冠",
      "enName": "Anatomy",
      "level": 2,
      "children": [
        {
          "designCode": 21101,
          "parentCode": 21100,
          "zhName": "全冠",
          "enName": "Full Crown",
          "level": 3,
          "children": []
        },
        {
          "designCode": 21102,
          "parentCode": 21100,
          "zhName": "临时冠",
          "enName": "Temporary Crown",
          "level": 3,
          "children": []
        },
        {
          "designCode": 21104,
          "parentCode": 21100,
          "zhName": "嵌体/高嵌体",
          "enName": "Inlay/Onlay",
          "level": 3,
          "children": []
        }
      ]
    }
  ]
}
```

```

    },
    {
      "designCode": 21105,
      "parentCode": 21100,
      "zhName": "贴面",
      "enName": "Veneer",
      "level": 3,
      "children": []
    },
    {
      "designCode": 21106,
      "parentCode": 21100,
      "zhName": "牙冠缺失位",
      "enName": "Crown Pontic",
      "level": 3,
      "children": []
    },
    {
      "designCode": 21107,
      "parentCode": 21100,
      "zhName": "临时冠缺失位",
      "enName": "Temporary Crown Pontic",
      "level": 3,
      "children": []
    },
    {
      "designCode": 21103,
      "parentCode": 21100,
      "zhName": "马里兰桥",
      "enName": "Maryland Bridge",
      "level": 3,
      "children": []
    },
    {
      "designCode": 21108,
      "parentCode": 21100,
      "zhName": "诊断蜡型",

```

```
        "enName": "Diagnostic Wax-up",
        "level": 3,
        "children": []
    },
    {
        "designCode": 21111,
        "parentCode": 21100,
        "zhName": "超薄临时牙桥",
        "enName": "Snap-on smile",
        "level": 3,
        "children": []
    }
]
},
{
    "designCode": 21200,
    "parentCode": 21000,
    "zhName": "基底冠",
    "enName": "Frame",
    "level": 2,
    "children": [
        {
            "designCode": 21201,
            "parentCode": 21200,
            "zhName": "内冠",
            "enName": "Coping",
            "level": 3,
            "children": []
        },
        {
            "designCode": 21202,
            "parentCode": 21200,
            "zhName": "解剖型内冠",
            "enName": "Anatomic Coping",
            "level": 3,
            "children": []
        }
    ],
},
```

```
{
  "designCode": 21203,
  "parentCode": 21200,
  "zhName": "内冠缺失位",
  "enName": "Coping Pontic",
  "level": 3,
  "children": []
},
{
  "designCode": 21204,
  "parentCode": 21200,
  "zhName": "解剖型内冠缺失位",
  "enName": "Anatomical Coping Pontic",
  "level": 3,
  "children": []
},
{
  "designCode": 21205,
  "parentCode": 21200,
  "zhName": "内冠&全解剖冠",
  "enName": "Coping&Full Anatomic Crown",
  "level": 3,
  "children": []
}
]
}
]
}
```

7.10. 普通建单

- 功能：普通模式创建订单，选择对应的设计类型和牙位信息后提交订单。
- 接口地址

请求地址	请求类型
/design-basic-service/openApi/v1/addOrderByGeneral	POST

• 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

• 请求参数

1. 顶层请求实体

参数	是否必填	参数类型	说明
designSoftware	是	Integer	设计软件 1001: 3Shape 1002: Exocad
designVersion	是	Integer	软件版本 3Shape 对应的版本 100101:2019 100102:2020 100103:2021 100104:2022 100105:2023 100106:2024 Exocad 对应的版本 100201:3.0

			100202:3.1
orders	是	Array[order]	订单集合

2. 订单实体 order

参数	是否必填	参数类型	说明
designCategoryCode	是	String	设计大类编码，一级类多个编码使用逗号隔开 例： 固定修复:21000 活动修复:22000 种植修复:23000 正畸&其他:24000 联合修复: 21000,22000
fileName	是	String	原始文件名称
orderNo	是	String	原订单号
patient	否	String	患者名称
remark	否	String	订单其他约束备注
s3FileId	是	String	文件上传返回的s3FileId
toothDesign	是	Array[ToothDesignModel]	单品牙位详情信息数

			组
--	--	--	---

3. 牙位设计信息 ToothDesignModel 可从 7.9 接口获取设计类型

参数	是否必填	参数类型	说明
code	是	Integer	3 级设计类型编码 7.9 接口 level 3 designCode 例: 21101
enName	是	String	3 级设计类型英文名称 例: Full Crown
zhName	是	String	3 级设计类型中文名称 例: 全冠
pidCode	是	Integer	2 级设计类型编码 7.9 接口 level 2 designCode 例: 21100
pidEnName	否	String	2 级设计类型英文名称 例: Anatomy
pidZhName	否	String	2 级设计类型中文名称 例: 解剖冠
rootCode	否	Integer	1 级设计类型编码 7.9 接口 level 1

			designCode 例: 21000
tooth	是	Array	设计类型对应牙位数字数组 牙位索引系统:FDI 符号 例: [21,31]

• 响应字段

参数	类型	说明
data	Boolean	布尔值表达调用是否成功

• 请求示例如下

```
POST /design-basic-service/openApi/v1/addOrderByGeneral HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{
  "designSoftware": 1001,
  "designVersion": 100101,
  "orders": [
    {
      "s3FileId": "03244ef643654d8f9a085f7c420d8b84",
      "fileName": "model_B1_P MMA_LOST_MTN_DELK_8,9_20230818.zip",
      "orderNo": "10076557200401",
      "remark": "",
      "toothDesign": [
        {
          "tooth": [18, 17],
          "code": 21401,
          "enName": "Sectioned Model",
          "pidCode": 21400,
          "pidEnName": "",
          "pidZhName": ""
        }
      ]
    }
  ]
}
```

<pre> "zhName": "分割代型牙模" }, { "sortTimeStamp": 1783581071549, "code": 21501, "enName": "Connector", "pidCode": 21500, "pidEnName": "", "pidZhName": "", "zhName": "连接体", "tooth": [27, 26] }, { "tooth": [27, 26], "code": 21204, "enName": "Anatomical Coping Pontic", "pidCode": 21200, "pidEnName": "", "pidZhName": "", "zhName": "解剖型内冠缺失位" }, { "code": 21101, "enName": "Full Crown", "pidCode": 21100, "pidEnName": "Anatomy", "pidZhName": "解剖冠", "tooth": [8, 9], "zhName": "全冠" }], "patient": "", "designCategoryCode": "21000" }] } { </pre>

```

{
  "code": 200,
  "data": true,
  "message": "成功",
  "detailMessage": null,
  "success": true,
  "error": false
}

```

8. STM 接口

8.1. 查询对应 STM 工具的方案列表

- 功能：根据授权的账户所在组织，获取组织下面对应 STM 工具所有开通云端订单任务账户的方案信息。
- 前置条件：用户需要有使用 STM 工具的权限。STM 工具需打开接收云端订单的配置项。正畸工具可在工具内部自行打开该配置，其他工具若无此配置项，需要联系 HeyGears 技术支持进行开启。
- 接口地址

请求地址	请求类型
/stm-service/openApi/v1/getPlanList	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在空白字符	Bearer w6iupfsv0xbb4ix8ibd9

- 请求参数

参数	是否必填	参数类型	说明
stmCode	是	Integer	STM 工具编码 9003: 固定 9009: 正畸 9011: 种植
orthoAppType	否	Integer	正畸应用类型: 1001 普通 正畸, 1002 正畸分步 (仅 正畸必填)

- 响应字段

参数	类型	说明
userCode	Integer	用户编码
userRealName	String	用户名称
userEmail	String	用户邮箱
plans	List<Plan>	方案列表
Plan		
planId	String	方案 Id
planName	String	方案名称
userName	String	用户名

- 请求示例如下

```
POST /stm-service/openApi/v1/getPlanList HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
```

```
{
  "stmCode": 9009,
  "orthoAppType": 1001
}

{
  "code": 200,
  "data": [
    {
      "userCode": 10001,
      "userName": "zhangsan",
      "userRealName": "张三",
      "userEmail": "zhangsan@heygears.com",
      "plans": [
        {
          "planId": "1011006_ortho_1763016504085",
          "planName": "正畸方案 1"
        },
        {
          "planId": "1011080_ortho_1758024896040",
          "planName": "正畸方案 2"
        }
      ]
    }
  ],
  "message": "成功",
  "detailMessage": null,
  "success": true,
  "error": false
}
```

8.2. 提交 STM 工具的云端订单

- 功能：可以选择对应账户推送云端订单，只要账户在 STM 工具登录后就会自动获取订单信息和执行。

- 接口地址

请求地址	请求类型
/stm-service/openApi/v1/createOrder	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
orderName	是	String	订单名称 (stmCode 为正畸 9009 时, 如果订单中的模型是上下颌, 需要传输上颌文件名+英文分号 ";" +下颌文件名)
stmCode	是	Integer	STM 工具编码 9003: 固定 9009: 正畸 9011: 种植
userCode	是	Integer	接收订单的用户编码

planId	是	String	方案 ID (stmCode 为正畸 9009 时, 请注意方案的“保留咬合关系”的开关, 需要与 orderName 参数传参字段保持一致。打开时, orderName 需要传上下颌; 关闭时, 传单颌。)
designApp	否	Integer	设计软件(固定和种植需要传入该参数) 1001: 3Shape 1002: ExoCAD
tagContent	是	String	打标内容
filePath	是	String	订单路径(目录/打包文件) 目录: 传入单个订单目录的绝对路径 打包文件: 先把单个待设计模型或者订单目录打包为 zip 文件, 再传入 zip 文件可访问的全路径
fileProtocol	否	Integer	文件传输协议(该参数可以为空则默认为 1001) 1001: File(共享文件夹/挂载的目录) 1002: HTTP/HTTPS 1003: SFTP (SSH File

			Transfer Protocol) 1004: S3 1005: FTPS (FTP Secure) ----- 暂时先支持 File
--	--	--	--

• 响应字段

参数	类型	说明
orderId	String	云端订单 Id

• 请求示例如下

POST /stm-service/openApi/v1/createOrder HTTP/1.1 Authorization: Bearer {access_token} Content-Type: application/json <pre>{ "orderId": "某某某的口扫", "stmCode": 9009, "userCode": 1001, "planId": "1011006_ortho_1763016504085", "designApp": 1001, "tagContent": "牙模 001", "filePath": "\\192.168.0.100\\data\\share\\某某某 001 订单", "fileProtocol": 1001 }</pre>	<pre>{ "code": 200, "data": { "orderId": "019EF291-DDB9-728B-83B3-A905908B392A" }, "message": "成功", "detailMessage": null, "success": true, "error": false }</pre>
--	--

}

8.3. 查询 STM 工具云端订单

- 功能：可以查询 7 天内的云端订单信息，各查询条件之间为 AND 关系；无需过滤的条件请留空。
- 接口地址

请求地址	请求类型
/stm-service/openApi/v1/getOrderList	POST

- 请求头

参数	内容	说明	示例
Authorization	Bearer {AccessToken}	该字段的值为通过文档 2.2 中授权方式获取到的 AccessToken。需注意 Bearer 与 AccessToken 之间存在一个空白字符	Bearer w6iupfsv0xbb4ix8ibdq

- 请求参数

参数	是否必填	参数类型	说明
stmCode	是	Integer	STM 工具编码 9003: 固定 9009: 正畸 9011: 种植

createDate	否	String	创建日期，格式： yyyy-MM-dd
userCode	否	Integer	用户编码
userEmail	否	String	用户邮箱
orderId	否	String	订单 Id
orderName	否	String	订单名称
status	否	Integer	订单状态 1001: 待处理 1002: 运算中 1003: 待确认 1004: 已确认 1005: 运算失败 1006: 设计异常
pageNum	否	Integer	页码，默认为 1，举例： 3
pageSize	否	Integer	每页记录数，默认为 10

- 响应字段

参数	类型	说明
userCode	Integer	用户编码
userEmail	String	用户邮箱
orders	List<Order>	订单列表

Order		
orderId	String	订单 Id
orderName	String	订单名称
createTime	String	创建时间
updateTime	String	最近一次更新时间
status	Integer	订单状态 1001: 待处理 1002: 运算中 1003: 待确认 1004: 已确认 1005: 运算失败 1006: 设计异常
remark	String	订单备注信息
userName	String	用户名
userRealName	String	用户名称

- 请求示例如下

```

POST /stm-service/openApi/v1/getOrderList HTTP/1.1
Authorization: Bearer {access_token}
Content-Type: application/json
{
  "stmCode": 9009,
  "createDate": "2026-06-20",
  "userCode": 1001,
  "userEmail": "zhangsan@heygears.com",
  "orderId": "019EF291-DDB9-728B-83B3-A905908B392A",
  "orderName": "某某某的口扫",

```

```
{
  "status": 1001,
  "pageNum": 1,
  "pageSize": 10
}

{
  "code": 200,
  "data": [
    {
      "userCode": 10001,
      "userName": "zhangsan",
      "userRealName": "张三",
      "userEmail": "zhangsan@heygears.com",
      "orders": [
        {
          "orderId": "019EF291-DDB9-728B-83B3-A905908B392A",
          "orderName": "某某某的口扫",
          "createTime": "2026-06-20 17:12:01",
          "updateTime": "2026-06-20 17:12:30",
          "status": 1003,
          "remark": ""
        },
        {
          "orderId": "019EF291-DDB9-728B-83B3-A905908B392A",
          "orderName": "某某某的口扫",
          "createTime": "2026-06-20 17:13:10",
          "updateTime": "2026-06-20 17:13:12",
          "status": 1001,
          "remark": "订单读取失败"
        }
      ]
    }
  ],
  "message": "成功",
  "detailMessage": null,
  "success": true,
  "error": false
}
```

8.4. STM 工具云端订单变更通知(消息订阅使用)

- 功能：当云端订单变更为“待确认”、“已确认”、“运算失败”会推送消息到关联队列，可以通过消息订阅或者定时轮询获取。
- 队列名 stm.cloud.order.<stmCode>（例如 stm.cloud.order.9009）：每个工具是单独的队列，如果有多个消费端要注意避免重复消费。
- 消息体字段

参数	类型	说明
eventId	String	事件唯一标识 (UUID)
messageVersion	String	消息版本号，固定 v1.0
eventTime	Long	时间戳
userCode	Integer	用户编码
userRealName	String	用户真名
userEmail	String	用户邮箱
orderId	String	订单 Id
orderName	String	订单名称
createTime	String	订 单 创 建 时 间 (yyyy-MM-dd HH:mm:ss, 东八区)
updateTime	String	订 单 最 近 更 新 时 间

		(yyyy-MM-dd HH:mm:ss, 东八区)
status	Integer	订单状态 1001: 待处理 1002: 运算中 1003: 待确认 1004: 已确认 1005: 运算失败 1006: 设计异常
remark	String	订单备注信息
stmCode	Integer	STM 工具编码

- 消息体示例如下

```
{
  "eventId": "6f2f8e8b-2f8c-4d44-9a6a-6d9d1c1b8a11",
  "messageVersion": "v1.0",
  "eventTime": 1782977781246,
  "stmCode": 9009,
  "userCode": 10001,
  "userRealName": "张三",
  "userEmail": "zhangsan@heygears.com",
  "orderId": "019EF291-DDB9-728B-83B3-A905908B392A",
  "orderName": "某某某的口扫",
  "createTime": "2026-06-20 17:12:01",
  "updateTime": "2026-06-20 17:12:30",
  "status": 1003,
  "remark": ""
}
```

