


# 直播间数据开放能力-接入文档V2

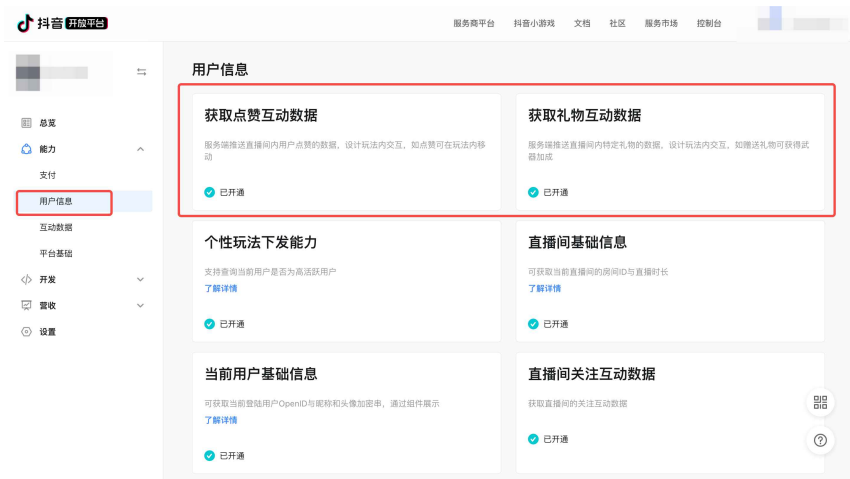
接入过程中遇到问题，请优先通过FAQ自查解决


## 前置工作

1. 前往[抖音开放平台](#) 申请创建互动玩法
2. 在平台上完成小玩法的资质认证，然后在“功能管理”栏完成对应直播间数据能力申请(若看不到，则找  申请)
  - a. 获取直播间评论数据能力



- b. 获取直播间礼物和点赞数据能力



3. 联系运营  ，审批能力申请
4. 申请通过后，开发者可在开放平台-控制台-直播小玩法-开发-开发配置中，为自己的弹幕玩法进行参数配置
  - a. 开放平台链接：<https://developer.open-douyin.com/console>



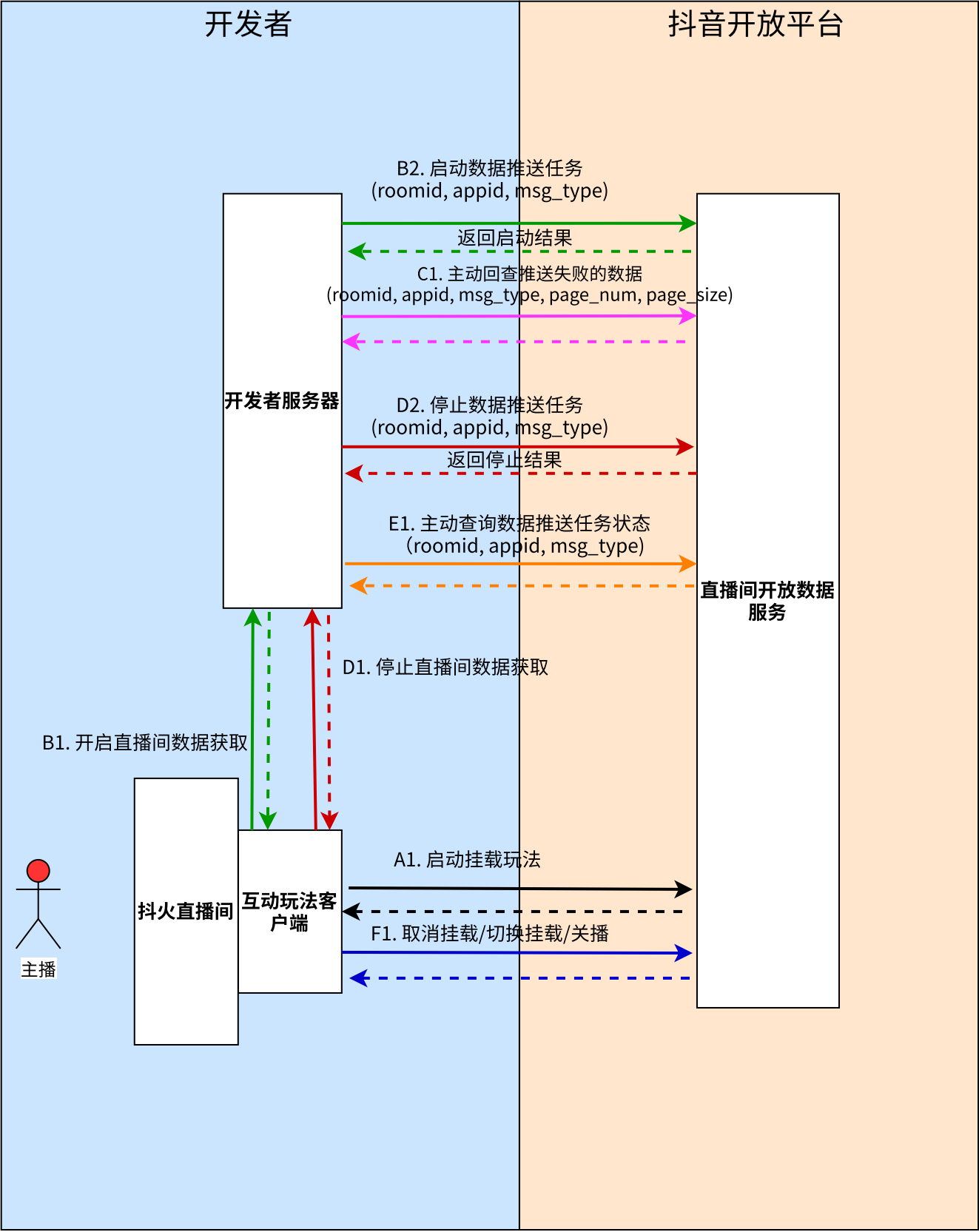
# 可靠性协议

消息类型	时效性保证	时序性保证	重复性保证	可靠性保证	备注
评论消息	秒级别	不保证绝对时序，但是时间段内相对时序基本是有保证的	可能会重复，需要开发者根据消息id做重复判断	不保证可靠性，会有数据丢失的情况	
礼物消息	秒级别	不保证绝对时序，但是时间段内相对时序基本是有保证的	可能会重复，需要开发者根据消息id做重复判断	尽力而为地保证可靠性，提供有限的丢失消息回查接口供开发者回查	
点赞消息	秒级别	不保证绝对时序，但是时间段内相对时序基本是有保证的	可能会重复，需要开发者根据消息id做重复判断	不保证可靠性，会有数据丢失的情况	

# 接入流程

- 抖音平台将「一定周期内直播间数据推送给开发服务器的行为抽象为“数据推送任务”」，数据推送任务由（直播间id, 小玩法id, 直播间消息类型）三者唯一确定，在小玩法被挂载的过程中，允许开发者自主多次启动和停止推送任务
- 抖音平台提供启动任务/停止任务/查询任务状态/回查丢失数据的接口给开发者接入，接口支持幂等调用，开发者只需要关注以下几条逻辑链路：
  - B: 启动直播间数据推送任务：注意，只有直播间中正在被挂载的小玩法才允许开发者启动数据推送任务。主播取消挂载小玩法或者关播后，抖音平台会自动停止数据推送
  - C: 主动回查推送失败的数据：注意，当前只有礼物数据支持回查，而且有缓存数量和时间限制。一般平台支持缓存前10w条数据，缓存一天，支持一天内过来分页查询

- D: 停止直播间数据推送任务
- E: 查询直播间数据推送任务状态



→  
A: 主播挂载玩法

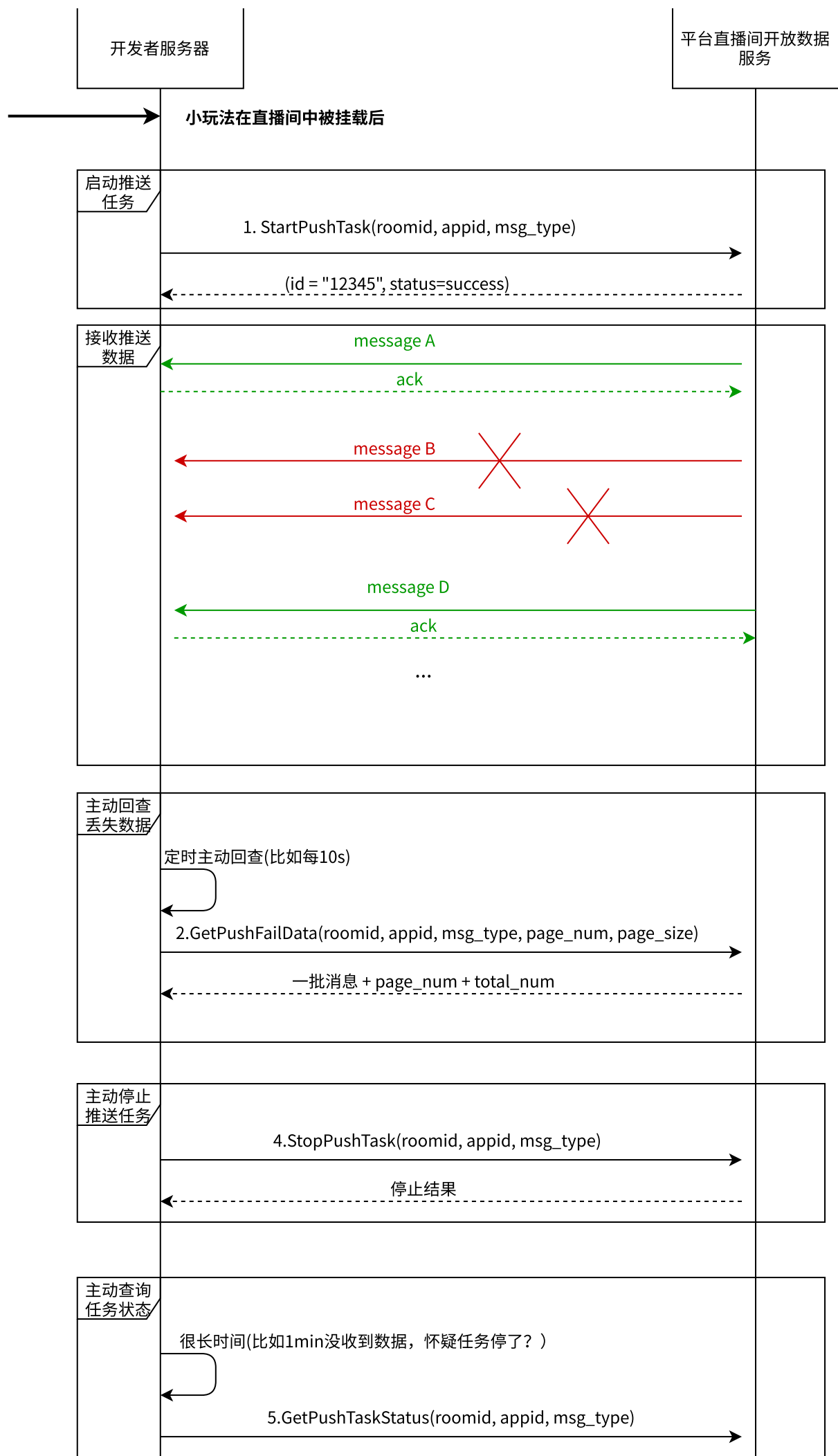
→  
B: 玩法启动数据推送任务

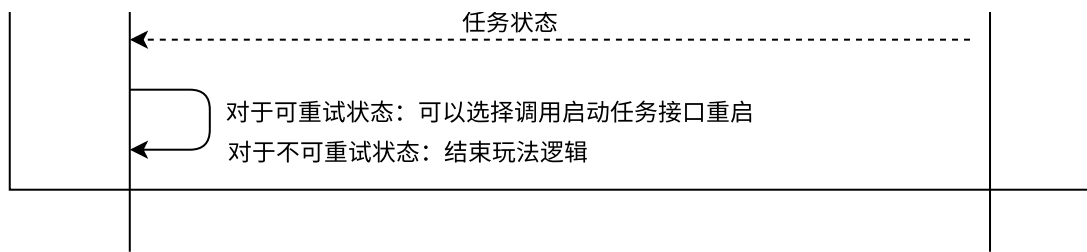
→  
C: 玩法回查缺失的序列号的数据

→  
D: 玩法停止数据推送任务

→  
E: 玩法查询数据推送任务状态

→  
F: 主播取消挂载玩法





## API接口

### 获取access\_token

该接口用于获取开放能力接口的调用凭证access\_token。

使用详情可参考: <https://microapp.bytedance.com/docs/zh-CN/mini-app/develop/server/interface-request-credential/get-access-token>

! access\_token有效期为2小时，需要定期刷新。

### 启动任务

💡 注意：不同类型的数据需要启动不同的任务单独监听，比如礼物数据单独启动一个，评论数据单独启动一个

启动直播间数据推送，启动成功后，直播间数据才会同步推送给开发者服务器

频率限制：单个 app\_id 调用上限为 10 次/秒。

#### 请求地址

1 POST [https://webcast.bytedance.com/api/live\\_data/task/start](https://webcast.bytedance.com/api/live_data/task/start)

#### 请求参数

- 请求Headers

	A	B	C	D
1	字段	数据类型	必填	说明
2	access-token	string	是	通过接口获取的access_token
3	content-type	string	是	必须包含 application/json

- 请求Body

	A	B	C	D
1	字段	数据类型	必填	说明
2	roomid	string	是	直播间id
3	appid	string	是	小玩法id
4	msg_type	string	是	直播间消息类型，需要前置申请开通了对应类型直播间数据能力才可以调用。 1. 评论：live_comment 2. 礼物：live_gift 3. 点赞：live_like

返回值

	A	B	C
1	字段	数据类型	说明
2	err_no	int	请求错误码，0表示成功，非0表示失败
3	err_msg	string	非0错误码时，携带额外的错误提示信息
4	logid	string	请求链路id, 用于出问题时提供给开平具体定位
5	data	struct	请求成功时的返回数据 { "task_id": "123456" // 任务id, string, 每次启动任务都会返回一个任务id }

返回数据示例

- 请求

```
1 curl -X POST "https://webcast.bytedance.com/api/live_data/task/start" -d
  '{"roomid":"123456","appid":"tt1234567cac","msg_type":"live_comment"}' -H
  "access-token:your_token" -H "content-type:application/json"
```

- 返回

```
1 {
2   "err_no": 0,
3   "err_msg": "ok",
4   "logid": "20220927122238291",
5   "data": {
6     "task_id": "763535353"
7   }
8 }
```

• 错误码说明

	A	B	C
1	错误名称	错误码	错误原因
2	Required Parameters Are Absent	40023	缺少必要的请求参数
3	Invalid Access Token	40022	无效的AccessToken
4	Service Unavailable	10001	服务内部错误，调用下游服务失败
5	PushTaskCanNotStart	5003019	推送任务不符合开启的限制条件

停止任务

停止直播间数据推送成功后，抖音平台会等停止前产生的存量数据推送完成，然后停止推送新数据  
频率限制：单个 app\_id 调用上限为 10 次/秒。

请求地址

```
1 POST https://webcast.bytedance.com/api/live_data/task/stop
```

请求参数

• 请求Headers

	A	B	C	D
1	字段	数据类型	必填	说明
2	access-token	string	是	通过接口获取的access_token
3	content-type	string	是	必须包含 application/json

• 请求Body

	A	B	C	D
1	字段	数据类型	必填	说明
2	roomid	string	是	直播间id
3	appid	string	是	小玩法id
4	msg_type	string	是	直播间消息类型，需要前置申请开通了对应类型直播间数据能力才可以调用。 1. 评论：live_comment 2. 礼物：live_gift 3. 点赞：live_like

返回值



	A	B	C
1	字段	数据类型	说明
2	err_no	int	请求错误码, 0表示成功, 非0表示失败
3	err_msg	string	非0错误码时, 携带额外的错误提示信息
4	logid	string	请求链路id, 用于出问题时提供给开平具体定位
5	data	struct	请求成功时的返回数据 {}

返回数据示例

- 请求

```
1 curl -X POST "https://webcast.bytedance.com/api/live_data/task/stop" -d
  '{"roomid":"123456","appid":"tt1234567cac","msg_type":"live_comment"}' -H
  "access-token:your_token" -H "content-type:application/json"
```

- 返回

```
1 {
2   "err_no": 0,
3   "err_msg": "ok",
4   "logid": "20220927122238291",
5   "data": {
6   }
7 }
```

- 错误码说明

	A	B	C
1	错误名称	错误码	错误原因
2	Required Parameters Are Absent	40023	缺少必要的请求参数
3	Invalid Access Token	40022	无效的AccessToken
4	Service Unavailable	10001	服务内部逻辑错误

查询任务状态

抖开提供接口给开发者查询推送任务状态，只有响应“运行中”状态任务才是正常运行，响应：

- 不存在：任务已经被删除，可能是因为主播取消挂载/已关播

- 未启动：任务还未被启动，需要开发者调用启动任务接口

频率限制：单个 app\_id 调用上限为 10 次/秒。

请求地址

```
1 GET https://webcast.bytedance.com/api/live_data/task/get
```

请求参数

- 请求Headers

	A	B	C	D
1	字段	数据类型	必填	说明
2	access-token	string	是	通过接口获取的access_token
3	content-type	string	是	必须包含 application/json

- 请求Query

	A	B	C	D
1	字段	数据类型	必填	说明
2	roomid	string	是	直播间id
3	appid	string	是	小玩法id
4	msg_type	string	是	直播间消息类型，需要前置申请开通了对应类型直播间数据能力才可以调用。 1. 评论：live_comment 2. 礼物：live_gift 3. 点赞：live_like

返回值

	A	B	C
1	字段	数据类型	说明
2	err_no	int	请求错误码，0表示成功，非0表示失败
3	err_msg	string	非0错误码时，携带额外的错误提示信息
4	logid	string	请求链路id, 用于出问题时提供给开平具体定位
5	data	struct	请求成功时的返回数据 { "status": 1, // int, 取值：1 任务不存在 2任务未启动 3任务运行中 }

返回数据示例

- 请求

```
1 curl -X GET "https://webcast.bytedance.com/api/live_data/task/get?
  roomid="123456"&appid=tt1234567cac&msg_type=live_comment" -H "access-
  token:your_token" -H "content-type:application/json"
```


- 返回

```
1 {
2   "err_no": 0,
3   "err_msg": "ok",
4   "logid": "20220927122238291",
5   "data": {
6     "status": 1
7   }
8 }
```

- 错误码说明

	A	B	C
1	错误名称	错误码	错误原因
2	Required Parameters Are Absent	40023	缺少必要的请求参数
3	Invalid Access Token	40022	无效的AccessToken
4	Service Unavailable	10001	服务内部逻辑错误

## 分页查询推送失败数据(仅支持礼物数据)

 注意：补偿数据的查询没有ack机制，需要开发者自行维护补偿状态，比如当前补偿到第几页了，下次该读取第几页

开发者可以定时请求api分页查询推送失败的数据，如果返回空则代表全部推送成功

频率限制：单个 app\_id 调用上限为 10 次/秒。

### 请求地址

```
1 GET https://webcast.bytedance.com/api/live_data/task/fail_data/get
```

### 请求参数

• 请求Headers

	A	B	C	D
1	字段	数据类型	必填	说明
2	access-token	string	是	通过接口获取的access_token
3	content-type	string	是	必须包含 application/json

• 请求Query

	A	B	C	D
1	字段	数据类型	必填	说明
2	roomid	string	是	直播间id
3	appid	string	是	小玩法id
4	msg_type	string	是	直播间消息类型，需要前置申请开通了对应类型直播间数据能力才可以调用。 1. 评论：live_comment 2. 礼物：live_gift 3. 点赞：live_like
5	page_num	int	是	页码：注意，需要从1开始
6	page_size	int	是	每页数据条数, 最大不超过100

返回值

	A	B	C
1	字段	数据类型	说明
2	err_no	int	请求错误码，0表示成功，非0表示失败
3	err_msg	string	非0错误码时，携带额外的错误提示信息
4	logid	string	请求链路id, 用于出问题时提供给开平具体定位
5	data	struct	请求成功时的返回数据 { "page_num": 1, // 数据对应的页码 "total_count": 10, // 总数据条数，注意不是页数，也不是当前页的数据条数 "data_list": [ // 当页的数据列表 { "roomid": "12345", // string类型，消息的房间id "msg_type": "live_gift", // string类型，表示消息类型 "payload": "[...]" // string类型，对应推送协议中的payload字符串，需 } ] }

返回数据示例

请求

```
1 curl -X GET "https://webcast.bytedance.com/api/live_data/task/fail_data/get?
   roomid=123456&appid=tt1234567cac&msg_type=live_comment&page_num=1&page_size=10"
   -H "access-token:your_token" -H "content-type:application/json"
```

返回

```
1 {
2     "err_no": 0,
3     "err_msg": "ok",
4     "logid": "20220927122238291",
5     "data": {
6         "page_num": 1,
7         "total_count": 100,
8         "data_list": [ // 当页的数据列表
9             {
10                 "roomid": "12345", // string类型, 消息的房间id
11                 "msg_type": "live_gift", // string类型, 表示消息类型
12                 "payload": "[...]" // string类型, 对应推送协议中的payload字符串, 需要unmarshal
13             }
14         ]
15     }
16 }
```

错误码说明

	A	B	C
1	错误名称	错误码	错误原因
2	Required Parameters Are Absent	40023	缺少必要的请求参数
3	Invalid Access Token	40022	无效的AccessToken
4	Request params error	10011	请求的分页查询参数不合法

礼物置顶



直播挂载小玩法后，开发者可以根据玩法场景自主控制当前场景需要置顶的礼物，前提是置顶的礼物在玩法维度礼物配置中。调用置顶礼物接口的条件：

- 已开通“获取礼物互动数据能力”，详情见前置工作
- 置顶的礼物数量不能超过6个，且必须在前置工作中配置的监听的礼物列表中

频率控制：单个app\_id调用上限为100次/秒。

注意

1. 当前调用礼物置顶延时不稳定，最长会有10秒左右的延迟，后续平台会优化这个延时
2. 需要重新打开礼物面板才能刷新置顶

请求地址

```
1 POST https://webcast.bytedance.com/api/gift/top_gift
```

请求参数

- 请求Headers

	A	B	C	D
1	字段	数据类型	必填	说明
2	x-token	string	是	通过接口获取的access_token
3	content-type	string	是	必须包含 application/json

请求Body

	A	B	C	D
1	字段	数据类型	必填	说明
2	room_id	string	是	直播间id
3	app_id	string	是	小玩法id
4	sec_gift_id_list	[]string	是	置顶礼物的加密礼物id，长度不得小于1且不得超过6，会过不在玩法维度礼物配置中的id。

返回值

	A	B	C
1	字段	数据类型	说明
2	errcode	int	请求错误码，0表示成功，非0表示失败
3	errmsg	string	非0错误码时，携带额外的错误提示信息
4	data	struct	请求成功时的返回数据 { "success_top_gift_id_list": [ "aa" ] // 成功置顶的礼物id }

返回数据示例

请求

```
1 curl -X POST 'https://webcast.bytedance.com/api/gift/top_gift' -H 'x-token:
  your_token' -H 'content-Type: application/json' -d '{
2     "room_id" : "123456",
3     "app_id": "your_app_id",
4     "sec_gift_id_list" : ["aa","bb"]
5 }'
```

返回

```
1 {
2     "err_no": 0,
3     "err_msg": "ok",
4     "logid": "20220927122238291",
5     "data": {
6         "success_top_gift_id_list": [
7             "aa", "bb"
8         ]
9     }
10 }
```


错误码说明

	A	B	C
1	错误名称	错误码	错误原因
2	the call condition is not met	50030	不符合调用条件
3	access token is expired	40004	无效的AccessToken
4	request params are invalid	40001	请求参数错误
5	Service Unavailable	10001	服务内部逻辑错误
6	system error	-1	服务系统内部错误

数据推送


推送协议

当前属于快速验证收益阶段，暂不支持长连接接入。后续平台会看业务发展情况，推动技术升级



**在高热直播间场景下，为了抖音平台数据推送服务的稳定运行，平台做如下数据推送限制：**

- 默认数据推送QPS限制为100，直播间数据量越大，数据丢失可能性越大
- 数据推送给开发者服务端，如果推送请求响应超过 2秒 (礼物数据是3s) 或者返回【非 2XX】状态码，平台认为推送失败，连续10次推送失败会触发推送的熔断，平台会直接丢弃数据若干秒再尝试推送



**!!! 开发者请务必校验数据签名，验证数据来源的合法性，否则存在被伪造数据攻击的危险，需自行担责**

- 推送方式：http回调
- 推送参数：

Method	POST		
参数形式	参数名	类型	说明
Header	x-nonce-str	string	签名用的随机字符串
	x-timestamp	int64	发送消息的毫秒级时间戳
	x-signature	string	



			请求签名，业务方接收后需要计算和校验签名，防伪造和篡改
	x-roomid	string	房间id
	x-msg-type	string	消息类型 live_comment: 直播间评论 live_gift: 直播间送礼 live_like: 直播间点赞
	content-type	string	固定值：application/json
body	~	string	具体类型消息payload对象的json序列化字符串

- **签名方式：**开发者请务必校验数据签名，验证数据来源的合法性，否则存在被伪造数据攻击的危险，需自行承担
  - 对表格中header参数(x-signature, content-type除外)，按key字典序从小到大排序, 排序后，将key-value按顺序连接起来。如：key1=value1&key2=value2。
  - 再直接拼接(无需连接符)上body字符串和secret（推送配置中的字段), 也就是前置工作中需要开发者提供给开平的签名密钥。**注意，字符串需要使用utf-8编码**
  - 把拼接好的字符串进行md5计算(16bytes)，并对md5计算结果进行base64编码，编码结果便是signature

go示例：

```

1  /**
2  比如:
3      header := map[string]string{
4          "x-nonce-str": "123456",
5          "x-timestamp": "456789",
6          "x-roomid":    "268",
7          "x-msg-type":  "live_gift",
8      }
9      bodyStr := "abc123你好"
10     secret := "123abc"
11
12     rawData为: x-msg-type=live_gift&x-nonce-str=123456&x-roomid=268&x-timestamp=4567,
13     signature为: PDcKhdlSrKEJif6uMKD2dw==
14  */
15 func signature(header map[string]string, bodyStr, secret string) string {
16     keyList := make([]string, 0, 4)
17     for key, _ := range header {
18         keyList = append(keyList, key)

```

```

19     }
20     sort.Slice(keyList, func(i, j int) bool {
21         return keyList[i] < keyList[j]
22     })
23     kvList := make([]string, 0, 4)
24     for _, key := range keyList {
25         kvList = append(kvList, key+"="+header[key])
26     }
27     urlParams := strings.Join(kvList, "&")
28     rawData := urlParams + bodyStr + secret
29     md5Result := md5.Sum([]byte(rawData))
30     return base64.StdEncoding.EncodeToString(md5Result[:])
31 }

```

java示例:

```

1  import java.security.MessageDigest;
2  import java.util.*;
3
4  /**
5   * @jdk >= 1.8
6   * @param header = {
7   *               "x-nonce-str": "123456",
8   *               "x-timestamp": "456789",
9   *               "x-roomid":    "268",
10 *               "x-msg-type":  "live_gift",
11 *           }
12 * @param bodyStr = "abc123你好"
13 * @param secret = "123abc"
14 * @return PDcKhdlSrKEJif6uMKD2dw==
15 */
16 public static String signature(Map<String, String> header, String bodyStr, String secret) {
17     List<String> keyList = new ArrayList<>(4);
18     header.forEach((key, val) -> keyList.add(key));
19     Collections.sort(keyList, String::compareTo);
20
21     List<String> kvList = new ArrayList<>(4);
22     for (String key : keyList) {
23         kvList.add(key + "=" + header.get(key));
24     }
25     String urlParams = String.join("&", kvList);
26     String rawData = urlParams + bodyStr + secret;
27     MessageDigest md = MessageDigest.getInstance("MD5");
28     md.update(rawData.getBytes());
29     return Base64.getEncoder().encodeToString(md.digest());

```

js示例: [抖音小玩法校验数据签名-js](#)

- 开发者对回调的响应: 响应2XX就平台会认为是成功的ack

如果开放平台推送请求收到的应答不符合规范或超时未收到, 开放平台会认为推送失败且不会继续推送。http应答码为2XX才会当作正常响应, 推送成功率会影响后续推送的频率。

## 评论数据-payload

```
1 [{
2   "msg_id": "123456781", // string类型id
3   "sec_openid": "xxxx", // 评论用户的加密openid, 当前其实没有加密
4   "content": "xxxx", // 评论内容
5   "avatar_url": "xxx", // 评论用户头像
6   "nickname": "xxxx", // 评论用户昵称(不加密)
7   "timestamp": 1649068964, // 评论毫秒级时间戳
8 }]
```

## 礼物数据-payload

```
1 [{
2   "msg_id": "123456782", // string类型id
3   "sec_openid": "xxxx", // 用户的加密openid, 当前其实没有加密
4   "sec_gift_id": "xxxx", // 加密的礼物id
5   "gift_num": 123, // 送出的礼物数量
6   "gift_value": 10000, // 礼物总价值, 单位分
7   "avatar_url": "xxx", // 用户头像
8   "nickname": "xxxx", // 用户昵称(不加密)
9   "timestamp": 1649068964, // 礼物毫秒级时间戳
10 }]
```

## 点赞数据-payload

```
1 [{
2   "msg_id": "123456783", // string类型id
3   "sec_openid": "xxxx", // 点赞用户的加密openid, 当前其实没有加密
4   "like_num": "xxxx", // 点赞数量, 上游2s合并一次数据
5   "avatar_url": "xxx", // 点赞用户头像
6   "nickname": "xxxx", // 点赞用户昵称(不加密)
```

```
7     "timestamp":1649068964, // 点赞毫秒级时间戳
8  }]
```

## 附录

### 礼物id映射表

[📖 互动玩法礼物池-for开发者](#)

## FAQ



### 自查手册

[📖 直播间接入数据开放能力-FAQ & 自查手册](#)

### 其他问题

[其他开发调试问题](#)

## 版本更新

- 12-09  郑庆鑫 应  李宁 同步的结论，删除openid的“跨局”和”加密“配置，默认跨局不加密，也就是说返回的sec\_openid就是用户在玩法中原始的openid
- 12-16 确定升级api, 将int64 id字段改为string类型，方便js开发者接入
- 12-20 完成api升级, 将int64 id字段改为string类型，方便js开发者接入