

视频直播

# 云直播服务端 SDK 参考

文档版本 01  
发布日期 2026-03-09



版权所有 © 华为云计算技术有限公司 2026。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

---

## 目录

---

1 SDK 概述.....	1
2 Java SDK 使用指导.....	5
3 Python SDK 使用指导.....	12
4 Go SDK 使用指导.....	17
5 PHP SDK 使用指导.....	22
6 修订记录.....	27

# 1 SDK 概述

华为云提供了直播服务端SDK，您可以直接集成服务端SDK来调用直播的相关API，从而实现对直播服务的快速操作。目前视频直播提供了Java、Python、Go和PHP四种语言SDK供开发者使用，如表1-1所示。

## 说明

SDK已知安全漏洞及修复版本、规避方案，请单击表1-1中的SDK对应语言的Github地址进行查看。

如果无法正常打开GitHub仓库，请检查您所使用的网络是否可以正常访问公网。由于GitHub的服务器部署在国外，国内用户访问时容易遇到无法打开的情况，请切换网络后再尝试打开。

表 1-1 服务端 SDK

语言	Github地址	参考文档
JAVA	<a href="#">huaweicloud-sdk-java-v3</a>	<a href="#">Java SDK使用指导</a>
Python	<a href="#">huaweicloud-sdk-python-v3</a>	<a href="#">Python SDK使用指导</a>
Go	<a href="#">huaweicloud-sdk-go-v3</a>	<a href="#">Go SDK使用指导</a>
PHP	<a href="#">huaweicloud-sdk-php-v3</a>	<a href="#">PHP SDK使用指导</a>

支持的直播API如表1-2所示，后续SDK将会持续升级，支持所有直播API。

表 1-2 SDK 与 API 对应关系

Java SDK	Python SDK	Go SDK	API参考
createDomain	create_domain	CreateDomain	<a href="#">创建直播域名</a>
deleteDomain	delete_domain	DeleteDomain	<a href="#">删除直播域名</a>

Java SDK	Python SDK	Go SDK	API参考
updateDomain	update_domain	UpdateDomain	<a href="#">修改直播域名</a>
showDomain	show_domain	ShowDomain	<a href="#">查询直播域名</a>
createDomainMapping	create_domain_mapping	CreateDomainMapping	<a href="#">域名映射</a>
deleteDomainMapping	delete_domain_mapping	DeleteDomainMapping	<a href="#">删除直播域名映射关系</a>
createTranscodingsTemplate	create_transcodings_template	CreateTranscodingsTemplate	<a href="#">创建直播转码模板</a>
updateTranscodingsTemplate	update_transcodings_template	UpdateTranscodingsTemplate	<a href="#">修改直播转码模板</a>
deleteTranscodingsTemplate	delete_transcodings_template	DeleteTranscodingsTemplate	<a href="#">删除直播转码模板</a>
showTranscodingsTemplate	show_transcodings_template	ShowTranscodingsTemplate	<a href="#">查询直播转码模板</a>
createStreamForbidden	create_stream_forbidden	CreateStreamForbidden	<a href="#">禁止直播推流</a>
updateStreamForbidden	update_stream_forbidden	UpdateStreamForbidden	<a href="#">修改禁推属性</a>
deleteStreamForbidden	delete_stream_forbidden	DeleteStreamForbidden	<a href="#">禁推恢复</a>
listStreamForbidden	list_stream_forbidden	ListStreamForbidden	<a href="#">查询禁止直播推流列表</a>
showBandwidth	show_bandwidth	ShowBandwidth	<a href="#">查询直播加速的带宽数据</a>
listLiveSampleLogs	list_live_sample_logs	ListLiveSampleLogs	<a href="#">获取直播播放日志</a>
createRecordRule	create_record_rule	CreateRecordRule	<a href="#">创建录制规则</a>
listRecordRules	list_record_rules	ListRecordRules	<a href="#">查询录制规则列表</a>
updateRecordRule	update_record_rule	UpdateRecordRule	<a href="#">修改录制规则</a>
deleteRecordRule	delete_record_rule	DeleteRecordRule	<a href="#">删除录制规则</a>

Java SDK	Python SDK	Go SDK	API参考
showRecordRule	show_record_rule	ShowRecordRule	<a href="#">查询录制规则配置</a>
runRecord	run_record	RunRecord	<a href="#">提交录制控制命令</a>
createRecordCallbackConfig	create_record_callback_config	CreateRecordCallbackConfig	<a href="#">创建录制回调配置</a>
listRecordCallbackConfigs	list_record_callback_configs	ListRecordCallbackConfigs	<a href="#">查询录制回调配置列表</a>
updateRecordCallbackConfig	update_record_callback_config	UpdateRecordCallbackConfig	<a href="#">修改录制回调配置</a>
showRecordCallbackConfig	show_record_callback_config	ShowRecordCallbackConfig	<a href="#">查询录制回调配置</a>
deleteRecordCallbackConfig	delete_record_callback_config	DeleteRecordCallbackConfig	<a href="#">删除录制回调配置</a>
showTraffic	show_traffic	ShowTraffic	<a href="#">查询直播加速的流量数据</a>
showOnlineUsers	show_online_users	ShowOnlineUsers	<a href="#">查询直播播放在线人数</a>
listBandwidthDetail	list_bandwidth_detail	ListBandwidthDetail	<a href="#">查询播放带宽趋势接口</a>
listDomainTrafficDetail	list_domain_traffic_detail	ListDomainTrafficDetail	<a href="#">查询播放流量趋势接口</a>
listDomainBandwidthPeak	list_domain_bandwidth_peak	ListDomainBandwidthPeak	<a href="#">查询播放带宽峰值接口</a>
listDomainTrafficSummary	list_domain_traffic_summary	ListDomainTrafficSummary	<a href="#">查询播放流量汇总接口</a>
listUsersOfStream	list_users_of_stream	ListUsersOfStream	<a href="#">查询观众趋势接口</a>
listQueryHttpCode	list_query_http_code	ListQueryHttpCode	<a href="#">查询直播拉流HTTP状态码接口</a>
listTranscodeData	list_transcode_data	ListTranscodeData	<a href="#">查询转码用量接口</a>
listRecordData	list_record_data	ListRecordData	<a href="#">查询录制用量接口</a>
listSnapshotData	list_snapshot_data	ListSnapshotData	<a href="#">查询截图用量接口</a>

Java SDK	Python SDK	Go SDK	API参考
showUpBandwidth	show_up_bandwidth	ShowUpBandwidth	<a href="#">查询上行带宽数据接口</a>
showStreamCount	show_stream_count	ShowStreamCount	<a href="#">查询域名维度推流数据接口</a>
listHistoryStreams	list_history_streams	ListHistoryStreams	<a href="#">查询历史推流列表接口</a>
showStreamPortrait	show_stream_portrait	ShowStreamPortrait	<a href="#">查询播放画像信息接口</a>
listSingleStreamFramerate	list_single_stream_framerate	ListSingleStreamFramerate	<a href="#">查询推流帧率数据接口</a>
listSingleStreamBitrate	list_single_stream_bitrate	ListSingleStreamBitrate	<a href="#">查询推流码率数据接口</a>

# 2 Java SDK 使用指导

本章节介绍了Java SDK的使用说明，您可以参考本章节进行快速集成开发。

## 开发前准备

- 已[注册](#)华为账号并开通华为云，完成[实名认证](#)。
- 具备已备案的域名用于直播推流和播放，并在视频直播控制台[添加推流和播放域名](#)，且已完成[域名关联](#)。
- 已具备开发环境，支持Java JDK 1.8及以上版本。
- 已获取华为云账号对应的Access Key（AK）和Secret Access Key（SK）。请在华为云控制台“我的凭证 > 访问密钥”页面上创建和查看您的AK/SK。具体请参见[访问密钥](#)。
- 已获取直播服务对应区域的项目ID，请在华为云控制台“我的凭证 > API凭证”页面上查看项目ID。具体请参见[API凭证](#)。

## 注意事项

当用户自己引入的三方库版本和Live业务依赖的三方库版本发生冲突时，例如Jackson，okhttp3版本冲突等。可以引入如下bundle包(3.0.40-rc版本后)，重定向SDK依赖的第三方库版本，解决版本冲突问题。详情请访问[SDK开发中心](#)，查看视频直播Java SDK的“整个SDK Bundle包”内容。

注意：bundle包已经包含了core包和云服务集合包，不需要再单独引入core包和服务包。否则根据maven依赖的解析顺序，可能会导致bundle包不生效。

```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-bundle</artifactId>
  <version>${version}</version>
</dependency>
```

## 安装 SDK

您可以通过Maven方式获取和安装SDK，首先需要在您的操作系统中[下载并安装Maven](#)，安装完成后您只需要在Java项目的pom.xml文件中加入相应的依赖项即可。

使用服务端SDK前，您需要安装“huaweicloud-sdk-core”和“huaweicloud-sdk-live”，具体的SDK版本号请参见[SDK开发中心](#)。

**注意**

以下代码示例中的**version**值，请根据实际的SDK版本号进行替换。

```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-core</artifactId>
  <version>3.1.11</version>
</dependency>
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-live</artifactId>
  <version>3.1.11</version>
</dependency>
```

## 开始使用

### 步骤1 导入依赖模块。

```
// 用户身份认证
import com.huaweicloud.sdk.core.auth.BasicCredentials;
// 请求异常类
import com.huaweicloud.sdk.core.exception.ClientRequestException;
import com.huaweicloud.sdk.core.exception.ServerResponseException;
// Http配置
import com.huaweicloud.sdk.core.http.HttpConfig;
// 导入live的客户端
import com.huaweicloud.sdk.live.v1.LiveClient;
// 导入待请求接口的request和response类
import com.huaweicloud.sdk.live.v1.model.ShowTranscodingsTemplateRequest;
import com.huaweicloud.sdk.live.v1.model.ShowTranscodingsTemplateResponse;
// 日志打印
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

### 步骤2 配置客户端属性。

#### 1. 默认配置。

```
// 使用默认配置
HttpConfig config = HttpConfig.getDefaultHttpConfig();
```

#### 2. （可选操作）配置代理。

```
// （可选）使用代理服务器
// 代理服务器的password直接写入代码，会有很大安全风险。建议密文形式存储在配置文件或者环境变量中，待使用时再解密，以确保安全。
// 代理配置设置前，请先在本地环境中设置环境变量PROXY_PASSWORD
config.withProxyHost("http://proxy.huaweicloud.com")
    .withProxyPort(8080)
    .withProxyUsername("test")
    .withProxyPassword(System.getenv("PROXY_PASSWORD"));
```

#### 3. （可选操作）配置连接。

```
// （可选）配置连接超时
config.withTimeout(3);
```

#### 4. （可选操作）配置SSL。

```
// （可选）配置跳过服务端证书校验
config.withIgnoreSSLVerification(true);
```

### 步骤3 初始化认证信息。

支持以下两种认证方式，您可以根据实际情况进行选择。

- 使用永久AK/SK

需要先获取永久AK和SK，以及projectId，您可以参考[开发前准备](#)获取。

```
BasicCredentials credentials = new BasicCredentials()
    .withAk(ak)
```

```
.withSk(sk)
.withProjectId(projectId)
```

- 使用临时AK/SK

首先需要获取临时AK、SK和SecurityToken，您可以[通过token获取](#)或者[通过委托授权获取](#)。

```
BasicCredentials credentials = new BasicCredentials()
    .withAk(ak)
    .withSk(sk)
    .withSecurityToken(securityToken)
    .withProjectId(projectId)
```

相关参数说明如下所示：

- **ak**：华为云账号Access Key，建议以密文形式存储在配置文件或者环境变量中，待使用时再解密，以确保安全。
- **sk**：华为云账号Secret Access Key，建议以密文形式存储在配置文件或者环境变量中，待使用时再解密，以确保安全。
- **projectId**：云服务所在区域的项目ID，根据您需要操作的项目所属区域选择对应的项目ID。
- **securityToken**：采用临时AK/SK认证场景下的安全票据。

#### 步骤4 初始化客户端。

```
//初始化直播服务的客户端
LiveClient liveClient = LiveClient.newBuilder()
    .withHttpConfig(config)
    .withCredential(credentials)
    .withRegion(region)
    .build();
```

**region**：直播服务应用区域和各服务的终端节点，具体请参见[地区和终端节点](#)。

#### 步骤5 发送请求并查看响应。

```
// 初始化请求，以调用查询转码模板接口为例
ShowTranscodingsTemplateResponse showTranscodingsTemplateResponse =
liveClient.showTranscodingsTemplate(
    new ShowTranscodingsTemplateRequest().withDomain(domain) // domain 为需要查询的域名
);
logger.info(showTranscodingsTemplateResponse.toString());
```

#### 步骤6 异常处理。

表 2-1 异常处理

一级分类	一级分类说明	二级分类	二级分类说明
ConnectionException	连接类异常	HostUnreachableException	网络不可达、被拒绝
		SslHandShakeException	SSL认证异常
RequestTimeoutException	响应超时异常	CallTimeoutException	单次请求，服务器处理超时未返回
		RetryOutageException	在重试策略消耗完成以后，仍无有效的响应

一级分类	一级分类说明	二级分类	二级分类说明
ServiceResponseException	服务器响应异常	ServerResponseException	服务端内部错误, Http响应码: [500,]
		ClientRequestException	请求参数不合法, Http响应码: [400, 500)

```
// 异常处理
try {
    ShowTranscodingsTemplateResponse showTranscodingsTemplateResponse =
liveClient.showTranscodingsTemplate(
    new ShowTranscodingsTemplateRequest().withDomain("play.example.huaweicloud.com")
);
} catch (ServiceResponseException e) {
    logger.error("HttpStatusCode: " + e.getHttpStatusCode());
    logger.error("RequestId: " + e.getRequestId());
    logger.error("ErrorCode: " + e.getErrorCode());
    logger.error("ErrorMsg: " + e.getErrorMsg());
}
```

### 步骤7 异步客户端使用。

```
// 初始化异步客户端
LiveAPIAsyncClient liveAsyncClient = LiveAPIAsyncClient.newBuilder()
    .withHttpConfig(config)
    .withCredential(credentials)
    .withRegion(region)
    .build();

// 发送异步请求
CompletableFuture<ShowTranscodingsTemplateResponse> future =
LiveAPIAsyncClient.showTranscodingsTemplate(
    new ShowTranscodingsTemplateRequest().withDomain("play.example.huaweicloud.com")
);

// 获取异步请求结果
showTranscodingsTemplateResponse response = future.get();
logger.info(response.toString());
```

### 步骤8 访问日志。

SDK在运行的时候采用了slf4j进行日志打印, 如果在运行代码实例时, 未配置日志实现库, 会有提示如下:

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

您可以根据目标项目实际情况引入对应的日志实现, 请在对应的工程项目的pom.xml文件中引入日志实现的依赖, 如下所示:

- slf4j

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-simple</artifactId>
  <version>1.7.21</version>
</dependency>
logback

<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
```

```
<version>1.2.3</version>
</dependency>
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-core</artifactId>
  <version>1.2.3</version>
</dependency>
```

- log4j  
<dependency>  
 <groupId>log4j</groupId>  
 <artifactId>log4j</artifactId>  
 <version>1.2.17</version>  
</dependency>

SDK默认会打印访问日志，每次请求都会有一条记录，日志名称为“HuaweiCloud-SDK-Access”，日志格式如下所示：

```
"{httpMethod} {uri}" {statusCode} {responseContentLength} {requestId}
```

其中“requestId”是华为云APIG返回的请求ID，可以用于问题跟踪。

可以根据项目情况在对应的日志配置文件中对访问日志进行屏蔽，或者单独打印到独立文件中。例如在logback中关闭访问日志：

```
<logger name="HuaweiCloud-SDK-Access" level="OFF"> </logger>
```

### 步骤9 原始HTTP侦听器。

在某些场景下可能对业务发出的HTTP请求进行Debug，需要看到原始的HTTP请求和返回信息，SDK提供侦听器功能获取原始的和加密的HTTP请求和返回信息。

#### 注意

原始信息打印仅在debug阶段使用，请不要在生产系统中将原始的Http头和Body信息打印到日志中，这些信息并未加密且其中包含敏感数据；当Body体为二进制内容，即Content-Type标识为二进制时，body为“\*\*\*\*”，详细内容不输出。

```
HttpConfig config = new HttpConfig().addHttpListener(HttpListener.forRequestListener(requestListener ->
// 注册侦听器后打印Http Request原始信息，请勿在生产系统中使用
logger.debug("REQUEST: {} {} {} {}",
  requestListener.httpMethod(),
  requestListener.uri(),
  requestListener.headers().entrySet().stream().flatMap(entry ->
    entry.getValue().stream().map(value -> entry.getKey() + " : " + value))
    .collect(Collectors.joining(";")),
  requestListener.body().orElse(""));
.addHttpListener(HttpListener.forResponseListener(responseListener ->
// 注册侦听器后打印Http Request原始信息，请勿在生产系统中使用
logger.debug("RESPONSE: {} {} {} {} {}",
  responseListener.httpMethod(),
  responseListener.uri(),
  responseListener.statusCode(),
  responseListener.headers().entrySet().stream().flatMap(entry ->
    entry.getValue().stream().map(value -> entry.getKey() + " : " + value))
    .collect(Collectors.joining(";")),
  responseListener.body().orElse(""));

LiveClient liveClient = LiveClient.newBuilder()
  .withHttpConfig(config)
  .withCredential(auth)
  .withRegion(region)
  .build();
```

---结束

## 代码示例

调用前请根据实际情况替换如下变量：{your endpoint string} 和 {your project id}。

认证用的AK、SK直接写入代码，会有很大安全风险，建议密文形式存放在配置文件或者环境变量中，待使用时再解密，以确保安全。

本示例以AK、SK保存在环境变量中为例。运行本示例前，请先在本地环境中设置环境变量HUAWEICLOUD\_SDK\_AK和HUAWEICLOUD\_SDK\_SK。

```
package com.huaweicloud.sdk.test;

// 用户身份认证
import com.huaweicloud.sdk.core.auth.BasicCredentials;
// 请求异常类
import com.huaweicloud.sdk.core.exception.ClientRequestException;
import com.huaweicloud.sdk.core.exception.ServerResponseException;
// HTTP配置
import com.huaweicloud.sdk.core.http.HttpConfig;
// 导入直播服务的客户端
import com.huaweicloud.sdk.live.v1.LiveClient;
// 导入待请求接口的request和response类
import com.huaweicloud.sdk.live.v1.model.ShowTranscodingsTemplateRequest;
import com.huaweicloud.sdk.live.v1.model.ShowTranscodingsTemplateResponse;
// 日志打印
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Application {
    private static final Logger logger = LoggerFactory.getLogger(Application.class);

    public static void showTranscodingsTemplate(LiveClient client) {
        try {
            // 初始化请求，以调用查询转码模板接口为例
            ShowTranscodingsTemplateResponse showTranscodingsTemplateResponse =
client.showTranscodingsTemplate(
                new ShowTranscodingsTemplateRequest().withDomain("play.example.huaweicloud.com")
            );
            // 输出json格式的字符串响应
            logger.info(showTranscodingsTemplateResponse.toString());
        } catch (ClientRequestException e) {
            logger.error("HttpStatusCode: " + e.getHttpStatusCode());
            logger.error("RequestId: " + e.getRequestId());
            logger.error("ErrorCode: " + e.getErrorCode());
            logger.error("ErrorMsg: " + e.getErrorMsg());
        }
    }

    public static void main(String[] args) {
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");
        String endpoint = "{your endpoint string}";
        String projectId = "{your project id}";

        // 配置客户端属性
        HttpConfig config = HttpConfig.getDefaultHttpConfig();
        config.withIgnoreSSLVerification(true);

        // 创建认证
        BasicCredentials auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk)
            .withProjectId(projectId);

        // 创建liveClient实例并初始化
        LiveClient liveClient = LiveClient.newBuilder()
            .withHttpConfig(config)
            .withCredential(auth)
            .withRegion(region)
```

```
        .build();  
        showTranscodingsTemplate(liveClient);  
    }  
}
```

# 3 Python SDK 使用指导

本章节介绍了Python SDK的使用说明，您可以参考本章节进行快速集成开发。

## 开发前准备

- 已[注册](#)华为账号并开通华为云，完成[实名认证](#)。
- 具备已备案的域名用于直播推流和播放，并在视频直播控制台[添加推流和播放域名](#)，且已完成[域名关联](#)。
- 已具备开发环境，支持Python 3及以上版本。
- 已获取华为云账号对应的Access Key（AK）和Secret Access Key（SK）。请在华为云控制台“我的凭证 > 访问密钥”页面上创建和查看您的AK/SK。具体请参见[访问密钥](#)。
- 已获取直播服务对应区域的项目ID，请在华为云控制台“我的凭证 > API凭证”页面上查看项目ID。具体请参见[API凭证](#)。

## 安装 SDK

视频直播服务端SDK支持python3及以上版本。执行“python --version”检查当前python的版本信息。

使用服务端SDK前，您需要安装“huaweicloudsdkcore”和“huaweicloudsdklive”，具体的SDK版本号请参见[SDK开发中心](#)。

- 使用pip安装  
执行如下命令安装华为云Python SDK核心库以及相关服务库：

```
# 安装核心库
pip install huaweicloudsdkcore
# 安装Live服务库
pip install huaweicloudsdklive
```

- 使用源码安装  
执行如下命令安装华为云Python SDK核心库以及相关服务库：

```
# 安装核心库
cd huaweicloudsdkcore-${version}
python setup.py install

# 安装Live服务库
cd huaweicloudsdklive-${version}
python setup.py install
```

## 开始使用

### 步骤1 导入依赖模块。

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials, GlobalCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcore.http.http_config import HttpConfig
# 导入指定Live的库
from huaweicloudsdklive.v1 import *
```

### 步骤2 配置客户端属性。

#### 1. 默认配置。

```
# 使用默认配置
config = HttpConfig.get_default_config()
```

#### 2. （可选操作）配置代理。

```
# 使用代理服务器（可选）
# 代理服务器的password直接写入代码，会有很大安全风险。建议密文形式存储在配置文件或者环境变量中，待使用时再解密，以确保安全。
# 代理配置设置前，请先在本地环境中设置环境变量PROXY_PASSWORD
config.proxy_protocol = 'http'
config.proxy_host = 'proxy.huaweicloud.com'
config.proxy_port = 80
config.proxy_user = 'username'
config.proxy_password = os.environ['PROXY_PASSWORD']
```

#### 3. （可选操作）配置连接。

```
# （可选）配置连接超时，支持统一指定超时时长timeout=timeout，或分别指定超时时长
timeout=(connect timeout, read timeout)
config.timeout = 3
```

#### 4. （可选操作）配置SSL。

```
# （可选）配置跳过服务端证书校验
config.ignore_ssl_verification = True
# 配置服务器端CA证书，用于SDK验证服务端证书合法性
config.ssl_ca_cert = ssl_ca_cert
```

### 步骤3 初始化认证信息。

支持两种方式认证，您可以根据实际情况进行选择。

- 使用永久AK/SK

首先需要获取永久AK和SK，以及projectId，您可以参考[开发前准备](#)获取。

```
credentials = BasicCredentials(ak, sk, project_id)
```

- 使用临时AK/SK

首先需要获取临时AK、SK和SecurityToken，您可以[通过token获取](#)或者[通过委托授权获取](#)。

```
credentials = BasicCredentials(ak, sk, project_id).with_security_token(security_token)
```

相关参数说明如下所示：

- **ak**：华为云账号Access Key，建议以密文形式存储在配置文件或者环境变量中，待使用时再解密，以确保安全。
- **sk**：华为云账号Secret Access Key，建议以密文形式存储在配置文件或者环境变量中，待使用时再解密，以确保安全。
- **project\_id**：云服务所在项目ID，根据你想操作的项目所属区域选择对应的项目ID。
- **security\_token**：采用临时AK/SK认证场景下的安全票据。

### 步骤4 初始化客户端。

```
# 初始化直播服务的客户端
client = LiveAPIClient.new_builder(LiveAPIClient) \
```

```
.with_http_config(config) \
.with_credentials(credentials) \
.with_endpoint(endpoint) \
.with_file_log(path="test.log", log_level=logging.INFO) \ # 日志打印至文件
.with_stream_log(log_level=logging.INFO) \ # 日志打印至控制台
.build()
```

- **endpoint**: 直播服务应用区域和各服务的终端节点，具体请参见[地区和终端节点](#)。
- **with\_file\_log**支持如下配置：
  - path: 日志文件路径。
  - log\_level: 日志级别，默认INFO。
  - max\_bytes: 单个日志文件大小，默认为10485760 bytes。
  - backup\_count: 日志文件个数，默认为5个。
- **with\_stream\_log**支持如下配置：
  - stream: 流对象，默认sys.stdout。
  - log\_level: 日志级别，默认INFO。

打开日志开关后，每次请求将打印访问日志，格式如下：

```
'%(asctime)s %(thread)d %(name)s %(filename)s %(lineno)d %(levelname)s %(message)s'
```

**步骤5** 发送请求并查看响应。

```
// 初始化请求，以调用查询转码模板接口为例
request = huaweicloudsdklive.ShowTranscodingsTemplateRequest("play.example.huaweicloud.com")
response = client.show_transcodings_template(request)
print(response)
```

**步骤6** 异常处理。

表 3-1 异常处理

一级分类	一级分类说明	二级分类	二级分类说明
ConnectionException	连接类异常	HostUnreachableException	网络不可达、被拒绝
		SslHandShakeException	SSL认证异常
RequestTimeoutException	响应超时异常	CallTimeoutException	单次请求，服务器处理超时未返回
		RetryOutageException	在重试策略消耗完成以后，仍无有效的响应
ServiceResponseException	服务器响应异常	ServerResponseException	服务端内部错误，Http响应码：[500,]
		ClientRequestException	请求参数不合法，Http响应码：[400, 500)

```
# 异常处理
try:
    request = huaweicloudsdklive.ShowTranscodingsTemplateRequest("play.example.huaweicloud.com")
```

```
response = client.show_transcodings_template(request)
except exception.ServiceResponseException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

### 步骤7 异步客户端使用。

```
# 初始化异步客户端，以初始化为例
live_client = LiveAsyncClient.new_builder(LiveAsyncClient) \
    .with_http_config(config) \
    .with_credentials(credentials) \
    .with_endpoint(endpoint) \
    .build()

# 发送异步请求
request = huaweicloudsdklive.ShowTranscodingsTemplateRequest("play.example.huaweicloud.com")
response = live_client.show_transcodings_template_async(request)

# 获取异步请求结果
print(response.result())
```

### 步骤8 原始Http侦听器。

在某些场景下可能对业务发出的Http请求进行Debug，需要看到原始的Http请求和返回信息，SDK提供侦听器功能获取原始的和加密的Http请求和返回信息。

#### 注意

原始信息打印仅在debug阶段使用，请不要在生产系统中将原始的Http头和Body信息打印到日志，这些信息并未加密且其中包含敏感数据；当Body体为二进制内容，即Content-Type标识为二进制时body为"\*\*\*"，详细内容不输出。

```
def response_handler(**kwargs):
    logger = kwargs.get("logger")
    response = kwargs.get("response")
    request = response.request

    base = "> Request %s %s HTTP/1.1" % (request.method, request.path_url) + "\n"
    if len(request.headers) != 0:
        base = base + "> Headers:" + "\n"
        for each in request.headers:
            base = base + " %s : %s" % (each, request.headers[each]) + "\n"
        base = base + "> Body: %s" % request.body + "\n\n"

    base = base + "< Response HTTP/1.1 %s " % response.status_code + "\n"
    if len(response.headers) != 0:
        base = base + "< Headers:" + "\n"
        for each in response.headers:
            base = base + " %s : %s" % (each, response.headers[each],) + "\n"
        base = base + "< Body: %s" % response.content
    logger.debug(base)

client = LiveAPIClient.new_builder(LiveAPIClient)\
    .with_http_config(config) \
    .with_credentials(credentials) \
    .with_endpoint(endpoint) \
    .with_file_log(path="test.log", log_level=logging.INFO) \
    .with_stream_log(log_level=logging.INFO) \
    .with_http_handler(HttpHandler().add_response_handler(response_handler)) \
```

HttpHandler支持“add\_request\_handler”和“add\_response\_handler”方法。

----结束

## 代码示例

调用前请根据实际情况替换如下变量：{your endpoint string} 以及 {your project id}。

认证用的AK、SK直接写入代码，会有很大安全风险，建议密文形式存放在配置文件或者环境变量中，待使用时再解密，以确保安全。

本示例以AK、SK保存在环境变量中为例。运行本示例前，请先在本地环境中设置环境变量HUAWEICLOUD\_SDK\_AK和HUAWEICLOUD\_SDK\_SK。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcore.http.http_config import HttpConfig
from huaweicloudsdklive.v1 import *

def show_transcodings_template(client):
    try:
        request = huaweicloudsdklive.ShowTranscodingsTemplateRequest("play.example.huaweicloud.com")
        response = client.show_transcodings_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

if __name__ == "__main__":
    ak = os.environ["HUAWEICLOUD_SDK_AK"]
    sk = os.environ["HUAWEICLOUD_SDK_SK"]
    endpoint = "{your endpoint}"
    project_id = "{your project id}"

    config = HttpConfig.get_default_config()
    config.ignore_ssl_verification = True
    credentials = BasicCredentials(ak, sk, project_id)

    live_client = LiveAPIClient.new_builder(LiveAPIClient) \
        .with_http_config(config) \
        .with_credentials(credentials) \
        .with_endpoint(endpoint) \
        .build()

    show_transcodings_template(live_client)
```

# 4 Go SDK 使用指导

本章节介绍了Go SDK的使用说明，您可以参考本章节进行快速集成开发。

## 开发前准备

- 已[注册](#)华为账号并开通华为云，完成[实名认证](#)。
- 具备已备案的域名用于直播推流和播放，并在视频直播控制台[添加推流和播放域名](#)，且已完成[域名关联](#)。
- 已具备开发环境，支持go 1.14及以上版本。
- 已获取华为云账号对应的Access Key（AK）和Secret Access Key（SK）。请在华为云控制台“我的凭证 > 访问密钥”页面上创建和查看您的AK/SK。具体请参见[访问密钥](#)。
- 已获取直播服务对应区域的项目ID，请在华为云控制台“我的凭证 > API凭证”页面上查看项目ID。具体请参见[API凭证](#)。

## 安装 SDK

视频直播服务端Go SDK支持go 1.14及以上版本。执行go version检查当前Go的版本信息。

使用go get安装华为云Go SDK，执行如下命令安装华为云Go SDK库以及相关依赖库，具体的SDK版本号请参见[SDK开发中心](#)。

```
# 安装华为云Go库
go get github.com/huaweicloud/huaweicloud-sdk-go-v3
# 安装依赖
go get github.com/json-iterator/go
```

## 开始使用

### 步骤1 导入依赖模块。

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/config"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/http/handler"
    live "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/live/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/live/v1/model"
    "net/http"
    "os"
)
```

## 步骤2 配置客户端属性。

### 1. 默认配置

```
# Use default configuration
httpConfig := config.DefaultHttpConfig()
```

### 2. (可选操作) 配置代理。

```
// 根据需要配置网络代理
// 代理服务器的password直接写入代码, 会有很大安全风险。建议密文形式存储在配置文件或者环境变量中, 待使用时再解密, 以确保安全。
// 代理配置设置前, 请先在本地环境中设置环境变量PROXY_PASSWORD
httpConfig.WithProxy(config.NewProxy().
    WithSchema("http").
    WithHost("proxy.huaweicloud.com").
    WithPort(80).
    WithUsername("testuser").
    WithPassword(os.Getenv("PROXY_PASSWORD")))
```

### 3. (可选操作) 配置连接。

```
httpConfig.WithTimeout(30);
```

### 4. (可选操作) 配置SSL。

```
// 根据需要配置是否跳过SSL证书校验
httpConfig.WithIgnoreSSLVerification(true);
```

## 步骤3 初始化认证信息。

支持两种方式认证, 您可以根据实际情况进行选择。

- 使用永久AK/SK

首先需要获取永久AK和SK, 以及projectId, 您可以参考[开发前准备](#)获取。

```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()
```

- 使用临时AK/SK

首先需要获取临时AK、SK和SecurityToken, 您可以[通过token获取](#)或者[通过委托授权获取](#)。

```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    WithSecurityToken(securityToken).
    Build()
```

相关参数说明如下所示:

- **ak**: 华为云账号Access Key, 建议以密文形式存储在配置文件或者环境变量中, 待使用时再解密, 以确保安全。
- **sk**: 华为云账号Secret Access Key, 建议以密文形式存储在配置文件或者环境变量中, 待使用时再解密, 以确保安全。
- **projectId**: 云服务所在项目ID, 根据您需要操作的项目所属区域选择对应的项目ID。
- **securityToken**: 采用临时AK/SK认证场景下的安全票据。

## 步骤4 初始化客户端。

```
# 初始化直播服务的客户端
client := live.NewLiveClient(
    live.LiveClientBuilder().
    WithEndpoints(endpoints).
    WithCredential(auth).
```

```
WithHttpConfig(config.DefaultHttpConfig()).  
Build()
```

**endpoint**: 直播服务应用区域和各服务的终端节点, 具体请参见[地区和终端节点](#)。

### 步骤5 发送请求并查看响应。

```
// 初始化请求, 以调用接口查询直播转码模板为例  
request := &model.ShowTranscodingsTemplateRequest{  
    Domain: "play.example.huaweicloud.com",  
}  
response, err := client.ShowTranscodingsTemplate(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}
```

### 步骤6 异常处理。

表 4-1 异常处理

一级分类	一级分类说明
ServiceResponseError	service response error
url.Error	connect endpoint error

```
# 异常处理  
response, err := client.ShowTranscodingsTemplate(request)  
if err == nil {  
    fmt.Println(response)  
} else {  
    fmt.Println(err)  
}
```

### 步骤7 原始Http侦听器。

在某些场景下可能对业务发出的Http请求进行Debug, 需要看到原始的Http请求和返回信息, SDK提供侦听器功能获取原始的和加密的Http请求和返回信息。

#### 注意

原始信息打印仅在debug阶段使用, 请不要在生产系统中将原始的Http头和Body信息打印到日志, 这些信息并未加密且其中包含敏感数据; 当Body体为二进制内容, 即Content-Type标识为二进制时body为"\*\*\*", 详细内容不输出。

认证用的ak和sk直接写到代码中有很大的安全风险, 建议在配置文件或者环境变量中密文存放, 使用时解密, 确保安全。

本示例以ak和sk保存在环境变量中为例, 运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD\_SDK\_AK和HUAWEICLOUD\_SDK\_SK。

```
func RequestHandler(request http.Request) {  
    fmt.Println(request)  
}  
  
func ResponseHandler(response http.Response) {  
    fmt.Println(response)  
}
```

```
client := live.NewLiveAPIClient(  
    live.LiveAPIClientBuilder().  
        WithEndpoints([]string{"{your endpoint}"}).  
        WithCredential(  
            basic.NewCredentialsBuilder().  
                WithAk(os.Getenv("HUAWEICLOUD_SDK_AK")).  
                WithSk(os.Getenv("HUAWEICLOUD_SDK_SK")).  
                WithProjectId("{your project id}").  
                Build()).  
        WithHttpConfig(config.DefaultHttpConfig()).  
        WithIgnoreSSLVerification(true).  
        WithHttpHandler(httphandler.  
            NewHttpHandler().  
                AddRequestHandler(RequestHandler).  
                AddResponseHandler(ResponseHandler))).  
    Build())
```

----结束

## 代码示例

调用前请根据实际情况替换如下变量：{your endpoint} 以及 {your project id}。

认证用的AK、SK直接写入代码，会有很大安全风险，建议密文形式存放在配置文件或者环境变量中，待使用时再解密，以确保安全。

本示例以AK、SK保存在环境变量中为例。运行本示例前，请先在本地环境中设置环境变量HUAWEICLOUD\_SDK\_AK和HUAWEICLOUD\_SDK\_SK。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/config"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/httphandler"  
    live "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/live/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/live/v1/model"  
    "net/http"  
    "os"  
)  
  
func RequestHandler(request http.Request) {  
    fmt.Println(request)  
}  
  
func ResponseHandler(response http.Response) {  
    fmt.Println(response)  
}  
  
func main() {  
    client := live.NewLiveAPIClient(  
        live.LiveAPIClientBuilder().  
            WithEndpoints([]string{"{your endpoint}"}).  
            WithCredential(  
                basic.NewCredentialsBuilder().  
                    WithAk(os.Getenv("HUAWEICLOUD_SDK_AK")).  
                    WithSk(os.Getenv("HUAWEICLOUD_SDK_SK")).  
                    WithProjectId("{your project id}").  
                    Build()).  
            WithHttpConfig(config.DefaultHttpConfig()).  
            WithIgnoreSSLVerification(true).  
            WithHttpHandler(httphandler.  
                NewHttpHandler().  
                    AddRequestHandler(RequestHandler).  
                    AddResponseHandler(ResponseHandler))).  
    Build())
```

```
request := &model.ShowTranscodingsTemplateRequest{
    Domain: "play.example.huaweicloud.com",
}
response, err := client.ShowTranscodingsTemplate(request)
if err == nil {
    fmt.Println("%+v\n",response)
} else {
    fmt.Println(err)
}
}
```

# 5 PHP SDK 使用指导

本章节介绍了PHP SDK的使用说明，您可以参考本章节进行快速集成开发。

## 开发前准备

- 已[注册](#)华为账号并开通华为云，完成[实名认证](#)。
- 具备已备案的域名用于直播推流和播放，并在视频直播控制台[添加推流和播放域名](#)，且已完成[域名关联](#)。
- 已具备开发环境，支持 PHP 5.6及以上版本。
- 已获取华为云账号对应的Access Key ( AK )和Secret Access Key ( SK )。请在华为云控制台“我的凭证 > 访问密钥”页面上创建和查看您的AK/SK。具体请参见[访问密钥](#)。
- 已获取直播服务对应区域的项目ID，请在华为云控制台“我的凭证 > API凭证”页面上查看项目ID。具体请参见[API凭证](#)。

## 安装 SDK

视频直播服务端SDK支持PHP 5.6及以上版本。执行“php --version”，检查当前PHP的版本信息。

推荐使用Composer安装SDK。

Composer是PHP的依赖管理工具，允许您在项目中声明依赖关系并安装这些依赖。

```
# 安装 Composer
curl -sS https://getcomposer.org/installer | php
# 安装 PHP SDK
composer require huaweicloud/huaweicloud-sdk-php
```

安装完毕后，您需要引入Composer的自动加载文件。

```
require 'path/to/vendor/autoload.php';
```

## 开始使用

### 步骤1 导入依赖模块。

```
namespace HuaweiCloud\SDK\Live\V1\Model;
require_once "vendor/autoload.php";
use HuaweiCloud\SDK\Core\Auth\BasicCredentials;
use HuaweiCloud\SDK\Core\Http\HttpConfig;
use HuaweiCloud\SDK\Core\Exceptions\ConnectionException;
use HuaweiCloud\SDK\Core\Exceptions\RequestTimeoutException;
use HuaweiCloud\SDK\Core\Exceptions\ServiceResponseException;
```

```
// 导入指定Live的库  
use HuaweiCloud\SDK\Live\V1\LiveClient;
```

## 步骤2 配置客户端属性。

### 1. 默认配置。

```
// 使用默认配置  
$config = HttpConfig::getDefaultConfig();
```

### 2. (可选操作) 配置代理。

```
// 使用代理服务器 (可选)  
// 代理服务器的password直接写入代码, 会有很大安全风险。建议密文形式存储在配置文件或者环境变量  
// 中, 待使用时再解密, 以确保安全。  
// 代理配置设置前, 请先在本地环境中设置环境变量PROXY_PASSWORD  
$config->setProxyProtocol('http');  
$config->setProxyHost('proxy.huawei.com');  
$config->setProxyPort(8080);  
$config->setProxyUser('username');  
$config->setProxyPassword(getenv('PROXY_PASSWORD'));
```

### 3. (可选操作) 配置连接。

```
// 配置连接超时 (可选), 支持统一指定超时时长timeout=timeout, 或分别指定超时时长  
// timeout=(connect timeout, read timeout)  
$config->setConnectionTimeout(3);
```

### 4. (可选操作) 配置SSL。

```
# (可选) 配置跳过服务端证书校验  
$config->setIgnoreSslVerification(true);  
# 配置服务器端CA证书, 用于SDK验证服务端证书合法性  
$config->setCertFile("{yourCertFile}");
```

## 步骤3 初始化认证信息。

支持两种方式认证, 您可以根据实际情况进行选择。

- 使用永久AK/SK

首先需要获取永久AK和SK, 以及projectId, 您可以参考[开发前准备](#)获取。

```
$basicCredentials = new BasicCredentials($ak,$sk,$projectId);
```

- 使用临时AK/SK

首先需要获得临时AK、SK和SecurityToken, 您可以[通过token获取](#)或者[通过委托授权获取](#)。

```
$basicCredentials = BasicCredentials(ak, sk, projectId).withSecurityToken(securityToken);
```

相关参数说明如下所示:

- **ak**: 华为云账号Access Key, 建议以密文形式存储在配置文件或者环境变量中, 待使用时再解密, 以确保安全。
- **sk**: 华为云账号Secret Access Key, 建议以密文形式存储在配置文件或者环境变量中, 待使用时再解密, 以确保安全。
- **project\_id**: 云服务所在项目ID, 根据您需要操作的项目所属区域选择对应的项目ID。
- **securitytoken**: 采用临时AK/SK认证场景下的安全票据。

## 步骤4 初始化客户端。

```
// 初始化直播服务的客户端  
$client = LiveClient::newBuilder(new LiveClient)  
->withHttpConfig($config)  
->withEndpoint($endpoint)  
->withCredentials($credentials)  
->build();  
$request = new CreateDomainRequest();
```

**endpoint**: 直播服务应用区域和各服务的终端节点, 具体请参见[地区和终端节点](#)。

**步骤5** 发送请求并查看响应。

```
// 初始化请求，以调用查询转码模板接口为例
$request = new ShowTranscodingsTemplateRequest("play.example.huaweicloud.com");
$response = $client->ShowTranscodingsTemplate($request);
echo $response;
```

**步骤6** 异常处理。

**表 5-1** 异常处理

一级分类	一级分类说明	二级分类	二级分类说明
ConnectionException	连接类异常	HostUnreachableException	网络不可达、被拒绝
		SslHandShakeException	SSL认证异常
RequestTimeoutException	响应超时异常	CallTimeoutException	单次请求，服务器处理超时未返回
		RetryOutageException	在重试策略消耗完成以后，仍无有效的响应
ServiceResponseException	服务器响应异常	ServerResponseException	服务端内部错误，Http响应码：[500,]
		ClientRequestException	请求参数不合法，Http响应码：[400, 500)

```
// 异常处理
try {
    $response = $client->ShowTranscodingsTemplate($request);
} catch (ConnectionException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (RequestTimeoutException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (ServiceResponseException $e) {
    echo "\n";
    echo $e->getHttpStatusCode(). "\n";
    echo $e->getErrorCode() . "\n";
    echo $e->getErrorMsg() . "\n";
}
```

**步骤7** 原始Http侦听器。

在某些场景下可能对业务发出的Http请求进行Debug，需要看到原始的Http请求和返回信息，SDK提供侦听器功能来获取原始的和加密的Http请求和返回信息。

**注意**

原始信息打印仅在debug阶段使用，请不要在生产系统中将原始的Http头和Body信息打印到日志，这些信息并未加密且其中包含敏感数据；当Body体为二进制内容，即Content-Type标识为二进制时body为"\*\*\*\*"，详细内容不输出。

```

$requestHandler = function ($argsMap) {
    if (isset($argsMap['request'])) {
        $sdkRequest = $argsMap['request'];
        $requestHeaders = $sdkRequest->headerParams;
        $requestBase = "> Request " . $sdkRequest->method . ' ' .
            $sdkRequest->url . "\n";
        if (count($requestHeaders) > 0) {
            $requestBase = $requestBase . '> Headers: ' . "\n";
            foreach ($requestHeaders as $key => $value) {
                $requestBase = $requestBase . ' ' . $key . ': ' .
                    $value . "\n";
            }
            $requestBase = $requestBase . '> Body: ' .
                $sdkRequest->body . "\n\n";
        }
        if (isset($argsMap['logger'])) {
            $logger = $argsMap['logger'];
            $logger->addDebug($requestBase);
        }
    }
};

$responseHandler = function ($argsMap) {
    if (isset($argsMap['response'])) {
        $response = $argsMap['response'];
        $responseBase = "> Response HTTP/1.1 " .
            $response->getStatusCode() . "\n";
        $responseHeaders = $response->getHeaders();
        if (count($responseHeaders) > 0) {
            $responseBase = $responseBase . '> Headers: ' . "\n";
            foreach ($responseHeaders as $key => $value) {
                $valueToString = "";
                if (is_array($value)) {
                    $valueToString = ".join($value);
                }
                $responseBase = $responseBase . ' ' . $key . ': ' .
                    $valueToString . "\n";
            }
            $responseBody = $response->getBody();
            $responseBase = $responseBase . '> Body: ' . (string)
                $responseBody . "\n\n";
        }
        if (isset($argsMap['logger'])) {
            $logger = $argsMap['logger'];
            $logger->addDebug($responseBase);
        }
    }
};

$httpHandler = new HttpHandler();
$httpHandler->addRequestHandlers($requestHandler);
$httpHandler->addResponseHandlers($responseHandler);

$iamClient = LiveClient::newBuilder()
    ->withHttpConfig($config)
    ->withEndpoint($endpoint)
    ->withCredentials(null)
    ->withStreamLogger($stream = 'php://stdout', $logLevel = Logger::INFO) // 日志打印至控制台
    ->withFileLogger($logPath='./test_log.txt', $logLevel = Logger::INFO) // 日志打印至文件
    ->withHttpHandler($httpHandler)
    ->build();

```

----结束

## 代码示例

调用前请根据实际情况替换如下变量：{your endpoint string} 以及 {your project id}。

认证用的AK、SK直接写入代码，会有很大安全风险，建议密文形式存放在配置文件或者环境变量中，待使用时再解密，以确保安全。

本示例以AK、SK保存在环境变量中为例。运行本示例前，请先在本地环境中设置环境变量HUAWEICLOUD\_SDK\_AK和HUAWEICLOUD\_SDK\_SK。

```
<?php
namespace HuaweiCloud\SDK\Live\V1\Model;
require_once "vendor/autoload.php";
use HuaweiCloud\SDK\Core\Auth\BasicCredentials;
use HuaweiCloud\SDK\Core\Http\HttpConfig;
use HuaweiCloud\SDK\Core\Exceptions\ConnectionException;
use HuaweiCloud\SDK\Core\Exceptions\RequestTimeoutException;
use HuaweiCloud\SDK\Core\Exceptions\ServiceResponseException;
use HuaweiCloud\SDK\Live\V1\LiveClient;

$ak = getenv('HUAWEICLOUD_SDK_AK');
$sk = getenv('HUAWEICLOUD_SDK_SK');
$endpoint = "https://live.cn-north-4.myhuaweicloud.com";
$projectId = "";
$credentials = new BasicCredentials($ak,$sk,$projectId);
$config = HttpConfig::getDefaultConfig();
$config->setIgnoreSslVerification(true);

$client = LiveClient::newBuilder(new LiveClient)
->withHttpConfig($config)
->withEndpoint($endpoint)
->withCredentials($credentials)
->build();
$request = new ShowTranscodingsTemplateRequest();

try {
    $response = $client->ShowTranscodingsTemplate($request);
} catch (ConnectionException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (RequestTimeoutException $e) {
    $msg = $e->getMessage();
    echo "\n". $msg ."\n";
} catch (ServiceResponseException $e) {
    echo "\n";
    echo $e->getHttpStatusCode(). "\n";
    echo $e->getErrorCode() . "\n";
    echo $e->getErrorMsg() . "\n";
}
echo "\n";
echo $response;
```

# 6 修订记录

发布日期	修订记录
2021-07-30	第二次正式发布。 本次更新说明如下： <ul style="list-style-type: none"><li>• 新增“PHP SDK使用指导”。</li></ul>
2020-10-30	第一次正式发布。