

快速接入

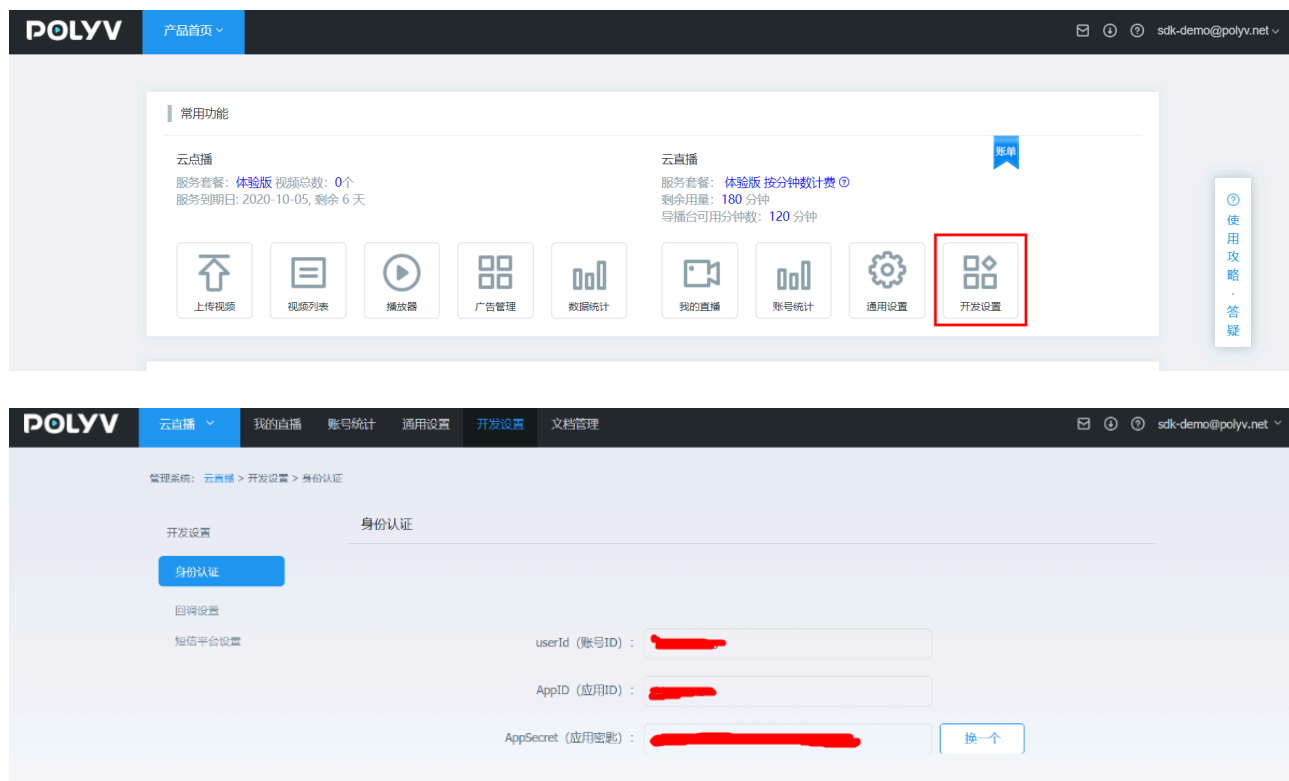
前提条件

在安装和使用保利威研讨会Java SDK功能前，确保B端已经具备如下环境：

- 安装Java环境。

保利威研讨会Java SDK支持1.8或更高版本。

- 注册保利威账号并获取访问密钥信息（UserId、AppId、AppSecret）。



1. 添加Maven依赖

将如下依赖加入到项目的POM文件中：

注意：受 Maven 中央仓库网络限制，阿里云云效 Maven 中央代理仓库可能会出现依赖查找不到的情况，建议使用中央仓库：<https://repo1.maven.org/maven2/>

```
<dependency>
  <groupId>net.polyv</groupId>
  <artifactId>polyv-java-seminar-sdk</artifactId>
  <version>2.2.8</version>
</dependency>
```

注意：为了消除Java的冗长代码，SDK使用了jdk1.8的最新特性lombok，请在IntelliJ或者Eclipse 中配置对lombok的支持

```
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.10</version>
</dependency>
```

2.初始化系统

在执行测试代码之前，需要B端先初始化系统配置，包括UserId \ AppId \ AppSecret，如没有以上信息，请参考本文[前提条件](#)部分获取，初始化示例代码如下，任选一种：

```
/**
 * 初始化配置,请配置自己的账号信息
 */
public static void initPolyvSeminar(){
    String userId = "xxx";
    String appId = "xxx";
    String appSecret = "xxx";
    SeminarGlobalConfig.init(appId, userId, appSecret);
    log.debug("--初始化完成--");
}
```

```
/**
 * 初始化配置并初始化 HTTP CLIENT 连接池超时时间和最大连接数配置,请配置自己的账号信息
 */
public static void initPolyvSeminar(){
    String userId = "xxx";
    String appId = "xxx";
    String appSecret = "xxx";
    Integer timeOut = 20000; //HTTP CLIENT 连接池超时时间
    Integer maxClientNum = 100; //HTTP CLIENT 最大连接数
    SeminarGlobalConfig.init( appId, userId, appSecret, timeOut , maxClientNum);
    log.debug("--初始化完成--");
}
```

以上代码一般配置于随系统启动执行一次的全局初始化中，如用spring框架，可以参考如下：

```
package net.polyv.live.config;

import org.springframework.beans.BeansException;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ApplicationContextAware;
import org.springframework.stereotype.Component;

import lombok.extern.slf4j.Slf4j;

/**
 * 默认启动配置类
 * @author: thomas
 */
@Slf4j
@Component
public class StartupListener implements ApplicationContextAware {

    @Override
    public void setApplicationContext(ApplicationConteXt applicationConteXt) throws BeansException {
        String userId = "xxx";
        String appId = "xxx";
        String appSecret = "xxx";
        SeminarGlobalConfig.init(appId,userId,appSecret);
        log.debug("--初始化完成--");
    }

}
```

3.执行测试代码

测试创建一个研讨会频道，单元测试代码如下：

```

import org.junit.Assert;

import com.alibaba.fastjson.JSON;

import lombok.extern.slf4j.Slf4j;
import net.polyv.common.v1.exception.PloyvSdkException;
import net.polyv.seminar.v1.config.SeminarGlobalConfig;
import net.polyv.seminar.v1.entity.channel.SeminarCreateChannelV2Request;
import net.polyv.seminar.v1.entity.channel.SeminarCreateChannelV2Response;
import net.polyv.seminar.v1.service.channel.impl.SeminarChannelServiceImpl;

@Slf4j
public class SeminarChannelDemo {

    /**
     * 调用demo，必须处理PloyvSdkException。
     * <p>
     * 参数合法性校验：SDK采用自定义验证框架对输入参数进行校验，如有参数不合格，将抛出PloyvSdkException异常，exception的
message
     * 包括具体校验不通过的字段信息，此异常是运行时异常，必须捕获处理相关业务逻辑；
     * <p>
     * 解析返回数据：解析返回数据，如SDK调用正常成功，将封装响应对象，正常返回，如服务器返回错误信息，SDK将抛出
PloyvSdkException异常，exception的message
     * 包括具体服务器执行错误信息，此异常是运行时异常，必须捕获处理相关业务逻辑；
     * @param args
     */
    public static void main(String[] args) {
        //全局初始化，此处应该全局执行一次
        String userId = "xxx";
        String appId = "xxx";
        String appSecret = "xxx";
        SeminarGlobalConfig.init(appId, userId, appSecret);
        log.debug("--初始化完成--");
        try {
            //封装API请求对象
            SeminarCreateChannelV2Request seminarCreateChannelV2Request = new SeminarCreateChannelV2Request();
            seminarCreateChannelV2Request.setName("测试研讨会创建") //设置频道主题信息
                .setSeminarHostPassword("666888") //研讨会主持人密码
                .setSeminarAttendeePassword("888888"); //研讨会参会人密码
            //调用SDK请求保利威服务器
            SeminarCreateChannelV2Response seminarCreateChannelV2Response =
                new SeminarChannelServiceImpl().createChannelV2(
                    seminarCreateChannelV2Request);
            Assert.assertNotNull(seminarCreateChannelV2Response);
            //正常返回做B端正常的业务逻辑
            if (seminarCreateChannelV2Response != null) {
                //to do something .....
                log.debug("研讨会频道创建成功 {}", JSON.toJSONString(seminarCreateChannelV2Response));
                log.debug("入会地址: https://meet.polyv.net/?channelId={}, 主持人登录密码: {}, 参会人登录密码: {}",
                    seminarCreateChannelV2Response.getChannelId(),
                    seminarCreateChannelV2Request.getSeminarHostPassword(),
                    seminarCreateChannelV2Request.getSeminarAttendeePassword());
                log.debug("观众观看地址: https://live.polyv.cn/watch/{}",
                    seminarCreateChannelV2Response.getChannelId());
            }
        }
    }
}

```

```

    } catch (PloyvSdkException e) {
        //参数校验不合格 或者 请求服务器端500错误, 错误信息见PloyvSdkException.getMessage()
        log.error(e.getMessage(), e);
        // 异常返回做B端异常的业务逻辑, 记录log 或者 上报到ETL 或者回滚事务
    } catch (Exception e) {
        log.error("SDK调用异常", e);
    }
}
}
}

```

执行代码后, 控制台应有如下关键输出, 表示整合完成:

```

[main] DEBUG net.polyv.common.v1.base.HttpUtil - http 请求 url: https://api.polyv.net/live/v4/channel/create?
requestId=2c272a5abc194d84bc0269bec4ea7395&appId=frr1r1zazn3&sign=823E7D930DC34877F557076A892C1C64&timestamp=1644
995639858 , 请求参数: {"appId":"frr1r1zazn3","name":"测试研讨会创
建","newScene":"seminar","requestId":"2c272a5abc194d84bc0269bec4ea7395","seminarAttendeePassword":"888888","semin
arHostPassword":"666888","sign":"823E7D930DC34877F557076A892C1C64","timestamp":"1644995639858"}
[main] DEBUG net.polyv.common.v1.base.HttpUtil - http 请求结果:
{"code":200,"status":"success","requestId":"bd4fbb48e79149e88f5225751a64ec7b.72.16449956414666319","data":
{"channelId":2820970,"userId":"1b448be323","scene":null,"channelPasswd":null,"seminarHostPassword":"666888","sem
inarAttendeePassword":"888888"},"success":true}
[main] DEBUG cn.timelost.spring.SeminarChannelDemo - 研讨会频道创建成功
{"channelId":"2820970","seminarAttendeePassword":"888888","seminarHostPassword":"666888","userId":"1b448be323"}

```

还可以官网登录验证创建是否成功:



至此, B端已经完成保利威研讨会Java SDK安装配置, 可以使用研讨会SDK进行其他功能开发和测试, 如B端接入过程有任何问题, 直接使用右下角在线客服找到售后技术支持提问, 请将问题的运行环境、操作步骤、错误反馈信息、联系方式同步反馈, 便于问题的快速定位和解决。