

丰桥API-SDK(JAVA)使用说明

SDK工具

SDK中提供了一系列有效的工具类,为客户端系统对接丰桥平台提供了有效的通道。

比如,对于下订单操作,在SDK中提供了客户端调用的工具类,客户端系统通过这个工具类可以便捷地将下订单等接口请求发送到顺丰业务系统中,与对应的下订单等业务接口进行对接,从而实现业务功能。下面将举例做详细介绍。

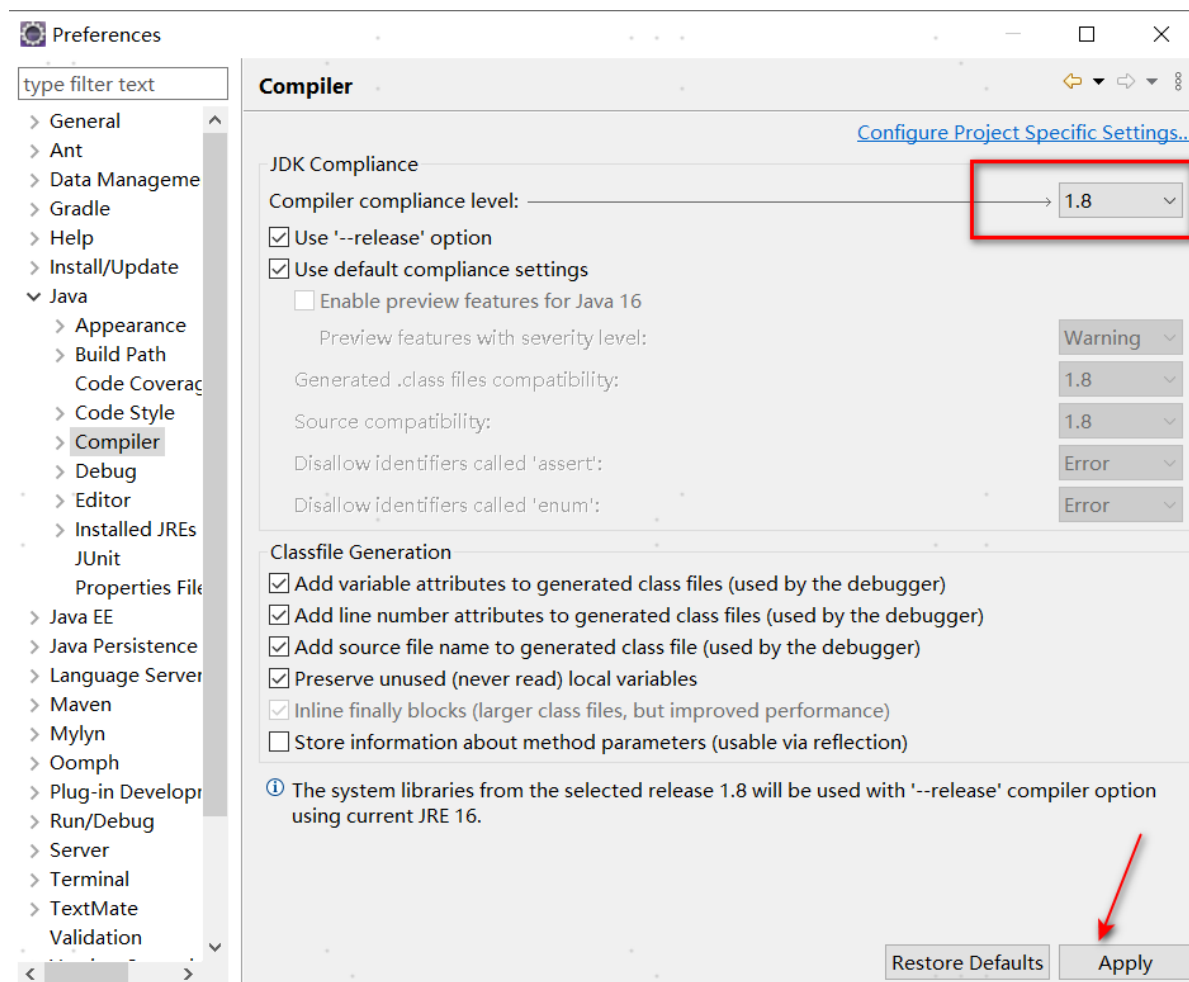
一、SDK下载介绍

- 下载说明

SDK(开发工具包)的调用是需要对用户的身份做合法性校验的,即clientCode及checkword,需要登录到丰桥平台,申请相应的丰桥服务,开发者申请控制台-业务对接-开发者对接,第三方软件对接申请控制台-业务对接-ERP用户对接。

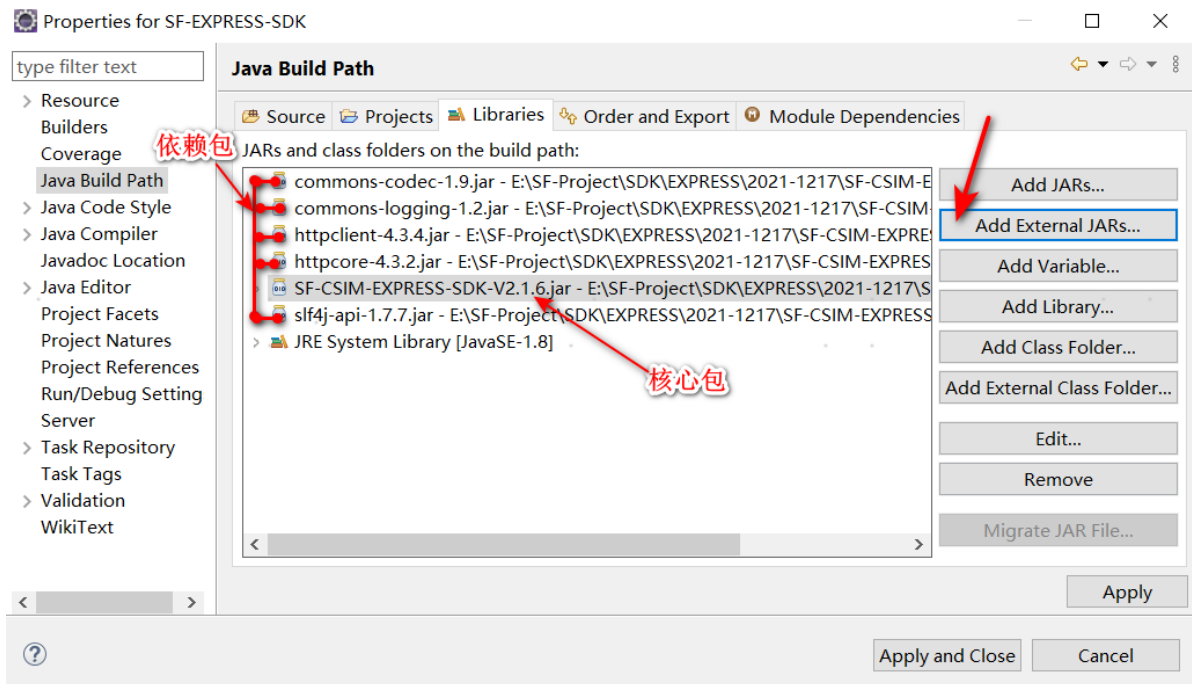
二、SDK使用说明(以java为例)

- 在“应用详情”中获取顾客编码 clientCode 以及校验码 checkword。
当在客户在丰桥平台,开发者申请开发者自助接入服务,第三方软件对接申请对接服务之后,并且配置所需要对接的接口信息,系统会将沙盒测试的clientCode和checkWord发送到您的邮箱。
- 准备开发环境1) 安装JDK (1.6及以上版本)
- 安装Eclipse(如:Eclipse Eclipse IDE for JEE Version: 2021-09 (4.21.0))
- 检查Eclipse的编译级别,请选择1.8
- 安装 java-sdk (以SF-CSIM-EXPRESS-SDK-V2.1.6.jar为例)



- 准备编码、创建java工程

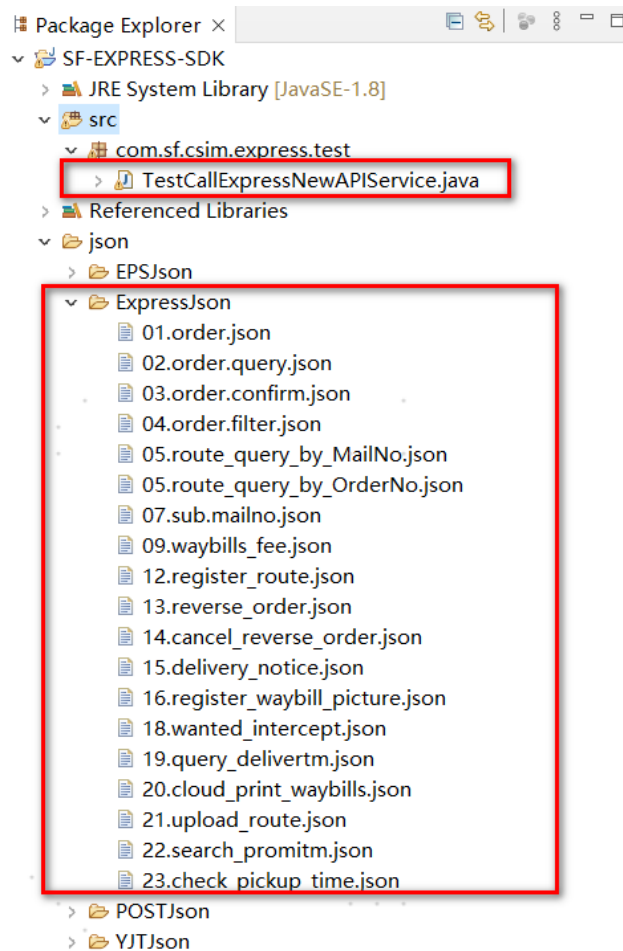
创建一个新的java工程,添加对SF-CSIM-EXPRESS-SDK-V2.1.6.jar的引用,并且添加jar运行所需的依赖包(在zip包内lib目录下所有的包),如下:



- 准备开发(下面以下单操作为例做详细介绍)

CallExpressServiceTools 是使用sdk-java的入口,下载的zip包内 json 文件夹对应的是丰桥服务接口对应的报文,如下订单使用的报文是 \json\EPSJson\EPS.1.COM_RECE_EPS_ORDER.json,请求对应的接口使用对应的请求报文(可以将请求报文的文件放置在java工程的目录内,方便测试调用。

下载的zip包java-demo目录还有个java版的测试类(TestCallExpressNewAPIService.java),可以在java工程中新建个测试目录如com.sf.test,完整的测试接口调用环境如下截图 (SDK包内json目录下所有文件可以直接导入到项目根目录下) :



丰桥快递API工具类 `CallExpressServiceTools`:

- 调用方法: `packageMsgData(String msgData,String timeStamp,String checkword)`;
- 请求参数: `String msgData, String timeStamp, String clientCode, String checkword`;

参数	描述
CALL_URL_BOX	客户端系统调用丰桥平台时对应接口的沙箱环境接口服务地址
CALL_URL_PROD	客户端系统调用丰桥平台时对应接口的生产环境接口服务地址
params	集合Map<String, String>, 包含 partnerID,requestID,serviceCode,timestamp,msgData,msgDigest 详见
msgData	接口的请求参数,报文格式为json
msgDigest	数字签名
clientCode	顾客编码,该字段通过在丰桥平台获取
checkWord	校验码,该字段通过在丰桥平台获取

- 返回参数: `String result`
调用丰桥服务对应接口的响应对象
- 核心对象: `CallExpressServiceTools` `toos=CallExpressServiceTools.getInstance()`;
使用该对象完成请求由客户系统端到丰桥接口服务系统的发送,该对象提供POST方法。具体使用方式也很简单,比如,当使用POST方法完成下单操作时,可以如下使用:

```

IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_CREATE_ORDER; //速运类--下订单接口
CallExpressServiceTools tools=CallExpressServiceTools.getInstance();
String msgData =tools.packageMsgData(standardService);
String msgDigest =tools.getMsgDigest(msgData,timeStamp,CHECK_WORD))
String result = HttpClientUtil.post(CALL_URL_BOX, params);

```

其中,CALL_URL_BOX 即为请求丰桥接口服务地址（分沙箱环境和生产环境）,msgData即为请求的业务参数。

使用小结总结:使用Tools工具类完成一项从客户端系统到丰桥平台的业务操作只需简单的三步。

第一步:使用正确的请求参数;

第二步:创建发送请求的核心对象tools封装请求参数和生成数字签名;

第三步:调用核心对象的对应方法完成请求的发送。

- 简单的sdk-java调用接口的代码如下:

```

//沙箱环境的地址
private static final String CALL_URL_BOX = "https://sfapi-sbox.sf-
express.com/std/service";
IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_CREATE_ORDER; //速运类--下订单接口

String timeStamp = String.valueOf(System.currentTimeMillis());
String msgData =tools.packageMsgData(standardService);

        params.put("partnerID", CLIENT_CODE); //对应顾客编码clientCode
        params.put("requestID", UUID.randomUUID().toString().replace("-", ""));
        params.put("serviceCode",standardService.getCode());// 服务代码
        params.put("timestamp", timeStamp);
        params.put("msgData", msgData); //实际请求业务报文
        params.put("msgDigest",
tools.getMsgDigest(msgData,timeStamp,CHECK_WORD));
String result = HttpClientUtil.post(CALL_URL_BOX, params);

```

- 完整的客户端调用sdk-java下单代码如下:

```

package com.sf.csim.express.test;

import java.io.UnsupportedEncodingException;
import java.util.HashMap;
import java.util.Map;
import java.util.UUID;
import com.sf.csim.express.service.CallExpressServiceTools;
import com.sf.csim.express.service.HttpClientUtil;
import com.sf.csim.express.service.IServiceCodeStandard;
import com.sf.csim.express.service.code.ExpressServiceCodeEnum;

public class TestCallExpressNewAPIService {

private static final String CLIENT_CODE = ""; //此处替换为您在丰桥平台获取的顾客编码
private static final String CHECK_WORD = ""; //此处替换为您在丰桥平台获取的校验码

```

```

//沙箱环境的地址
private static final String CALL_URL_BOX = "https://sfapi-sbox.sf-
express.com/std/service";
//生产环境的地址
private static final String CALL_URL_PROD = "https://sfapi.sf-
express.com/std/service";

    public static void main(String[] args) throws UnsupportedEncodingException {
        /**ExpressServiceCodeEnum    对应速运类-快递APIs
        POSTServiceCodeEnum          对应速运类-驿站APIs
        YJTServiceCodeEnum           对应解决方案-医寄通APIs
        EPSServiceCodeEnum           对应解决方案-快递管家APIs
        详情见code目录下枚举类，客户可自行修改引用的该类
        **/
        IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_CREATE_ORDER; //下订单
        // IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_SEARCH_ORDER_RESP; //查订单
        // IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_UPDATE_ORDER; //订单取消
        // IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_FILTER_ORDER_BSP; //订单筛选
        // IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_SEARCH_ROUTES; //查路由
        // IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_GET_SUB_MAILNO; //子单号
        // IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_QUERY_SFWAYBILL; //查运费
        // IServiceCodeStandard standardService =
ExpressServiceCodeEnum.EXP_RECE_REGISTER_ROUTE; //注册路由
        // IServiceCodeStandard standardService =

        CallExpressServiceTools tools=CallExpressServiceTools.getInstance();

        // set common header
        Map<String, String> params = new HashMap<String, String>();

        String timeStamp = String.valueOf(System.currentTimeMillis());
        String msgData =tools.packageMsgData(standardService);

        params.put("partnerID", CLIENT_CODE); // 顾客编码
        params.put("requestID", UUID.randomUUID().toString().replace("-", ""));
        params.put("serviceCode", standardService.getCode()); // 接口服务码
        params.put("timestamp", timeStamp);
        params.put("msgData", msgData);
        params.put("msgDigest",
tools.getMsgDigest(msgData, timeStamp, CHECK_WORD));

        String result = HttpClientUtil.post(CALL_URL_BOX, params);
        System.out.println("===调用地址 ==="+CALL_URL_BOX);
        System.out.println("===顾客编码 ==="+CLIENT_CODE);
        System.out.println("===返回结果: " +result);

    }

```

```
}
```

注意:

- SDK包code目录下提供速运类API及解决方案API的接口枚举类源码，客户可以自行维护到本系统的项目中修改服务代码和调用的接口报文的映射关系，PrePathEnum可以修改API请求的默认路径
- 下单时订单号(orderid)不能重复,否则会由于“重复下单”导致下单失败