**中科软险累计风险保额系统**

**接口文档说明书**

**Sinosoft**

**中科软科技股份有限公司**

**2016年9月**



# 概述

## 主要涉税业务项目：

# 接口协议

接口是以REST协议对外提供服务，通过统一的URL如：http://localhost:8080/tax-vat/router的URL就可以访问服务获得响应，接入系统可以自由地随时随地访问服务。

## 请求模型

请求模型由系统参数及业务参数组成：

### 请求报文

Json：

reauestMessage={

"riskCode":"2014",

"name":"张三",

"sex":"性别",

"birthDay":"生日",

"idType":"证件类型",

"idNo":"证件号码",

"amnt":"保额",

"prem":"保费"

}

Xml：

待补充 已补充

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<reauestMessage>

<riskCode>2014</riskCode>

<name>张三</name>

<sex>性别</sex>

<birthDay>生日</birthDay>

<idType>证件类型</idType>

<idNo>证件号码</idNo>

<amnt>保额</amnt>

<prem>保费</prem>

</reauestMessage>

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| riskCode | 是 | 险种代码 |
| name | 是 | 姓名 |
| sex | 是 | 性别 |
| birthDay | 是 | 出生日期：2012-01-01 |
| idType | 是 | 证件类型 |
| idNo | 是 | 证件号码 |
| amnt | 是 | 保额 |
| prem | 是 | 保费 |

### 系统参数

系统参数是由平台统一定义的一组参数，每个服务都拥有这些参数，用以传送框架级的参数信息。平台共有7个系统参数，如下表：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 是否必须 | 参数说明 |
| appKey | 是 | 应用键对应用一个密钥seeret。要调用服务，必须通过申请获取appKey/seeret后，才能进行应用的开发。 |
| sessionId | 否 | 会话ID,一般是一个36位的UUID,登录服务后获取 |
| method | 是 | 服务方法名，如: 价税分离 |
| v | 是 | 服务方法的版本号，如 1.0、 2.0 等。 一个具体的服务方法上 method+v 两者唯一确定。因  此服务平台必须保证所有服务的 method+v 的唯一性。 |
| messageFormat | 否 | 通信报文格式，可选值为 xml 和 json，默认为 xml。 |
| locale | 否 | 本地化类型，默认为 zh\_CN。 |
| sign | 否 | 签名串，请求参数的签名，服务平台通过它验证请求数据的合法性。 |

### 业务级参数

业务级参数，顾名思义是由业务逻辑需要自行定义的，每个服务 API 都可以定义若干

个自己的业务级参数。后续的章节将会根据不同的接口描述它自己的业务级参数。

## 响应模型

请参照

### 分体式响应报文

服务开放平台对服务的处理结果统一由响应报文告之调用终端， 不管服务处理正确与否， 都必须对应一个响应报文。如果响应报文既要包含正确的业务响应报文，又要包括错误的响应报文， 将增加服务响应报文格式的复杂度。平台已经建立了一个可以描述所有类型错误的错误模型，因此错误响应报文的格式是统一的。业务服务的结果响应报文是业务相关的，也就是说，每个业务服务方法的正确业务响应报文是不一样的。

基于以上的分析， 平台将正确业务响应报文和错误处理响应报文独立开来，如果服务正确执行，返回正确的业务响应报文，否则返回错误响应报文。

### 成功报文模型

Json：

{"requestTime":1461061925740,"responseTime":1461061926348, "riskCode":"2014","name":"张三","sex":"性别","birthDay":"生日","idType":"证件类型","idNo":"证件号码","resultDetailed":[{"type":"寿险","totalAmnt ":"累计保额"," totalprem ":"累计保费"},{"type":"意外险","totalAmnt ":"累计保额"," totalprem ":"累计保费"},{"type":"重疾险","totalAmnt ":"累计保额"," totalprem ":"累计保费"},{"type":"防癌险","totalAmnt ":"累计保额"," totalprem ":"累计保费"},]}

Xml：

待补充 已补充

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<success>

<requestTime>1461061925740</requestTime>

<responseTime>1461061926348</responseTime>

<ventureInfo>

<riskCode>2014</riskCode>

<name>张三</name>

<sex>性别</sex>

<birthDay>生日</birthDay>

<idType>证件类型</idType>

<idNo>证件号码</idNo>

<resultDetailed>

<type>寿险</type>

<totalAmnt>累计保额</totalAmnt>

<totalprem >累计保费</totalprem >

</resultDetailed>

<resultDetailed>

<type>意外险</type>

<totalAmnt>累计保额</totalAmnt>

<totalprem >累计保费</totalprem >

</resultDetailed>

<resultDetailed>

<type>重疾险</type>

<totalAmnt>累计保额</totalAmnt>

<totalprem >累计保费</totalprem >

</resultDetailed>

<resultDetailed>

<type>防癌险</type>

<totalAmnt>累计保额</totalAmnt>

<totalprem >累计保费</totalprem >

</resultDetailed>

<ventureInfo>

</success>

### 错误报文模型

当服务发生错误时， Rop 将返回统一格式的错误报文，它由一个主错误和若干个子错

误组成。下面即是一个完整的错误报文：

Json：

{"errorToken":"@@$-ERROR\_TOKEN$-@@","code":"33","message":"非法的参数","solution":"请查看根据服务接口对参数格式的要求","subErrors":[{"code":" isv.invalid-paramete:salary ","message":"参数salary无效，格式不对、非法值、越界等"}]}

Xml：

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<error code="33">

<message>非法的参数</message>

<solution>请查看根据服务接口对参数格式的要求</solution>

<subErrors>

<subError code="isv.invalid-paramete:salary">

<message>参数salary无效，格式不对、非法值、越界等</message>

</subError>

<subError code="isv.invalid-paramete:userName">

<message>参数userName无效，格式不对、非法值、越界等</message>

</subError>

</subErrors>

</error>

<error>代表主错误，它拥有一个对应编码，此外还包括错误消息及解决方法。 subErrors

元素中包含多个子错误，详细说明是哪些参数违反数据了校验规则， 子错误的编码附带了

违反校验规则的参数名，对定位错误非常有帮助。

系统错误编码如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 错误编码 | 错误说明 | 错误编码 | 错误说明 | 错误编码 | 错误说明 |
| 1 | 服务不可用 | 20 | 缺少 sessionId参数 | 29 | 非法的版本参数 |
| 2 | 开发者权限不足 | 21 | 无效的 sessionId参数 | 30 | 不支持的版本号 |
| 3 | 用户权限不足 | 22 | 缺少 appKey 参数 | 31 | 无效报文格式类型 |
| 4 | 图片上传失败 | 23 | 无效的 appKey 参数 | 32 | 缺少必选参数 |
| 5 | HTTP 方法被禁止 | 24 | 缺少签名参数 | 33 | 非法的参数 |
| 6 | 编码错误 | 25 | 无效签名 | 34 | 用户调用服务的次数超限 |
| 7 | 请求被禁止 | 26 | 缺少方法名参数 | 35 | 会话调用服务的次数超限 |
| 8 | 服务已经作废 | 27 | 不存在的方法名 | 36 | 应用调用服务的次数超限 |
| 9 | 业务逻辑出错 | 28 | 缺少版本参数 | 37 | 应用调用服务的频率超限 |

### 报文输出格式

默认情况下， 平台的响应报文是以 XML 的格式输出的。除 XML 之外， 平台还支持JSON 格式的响应报文。您只要将 messageFormat的系统级参数设置为 json，就会返回 JSON 格式的响应报文了。

值得注意的是，当前 平台不支持混合使用 XML 和 JSON 的报文格式，也即请求和响应的数据格式要么采用 XML，要么采用 JSON。如果希望采用 XML 格式的请求参数值，那么响应的报文也必须是 XML。

# 接口说明

**接口功能说明**

为业务系统提供计划进项费用的增值税价税计算及分离，并保存计算结果数据到中间表。

**接口调用请求说明**

http请求方式: POST

url: [http://10.132.2.139:8080/venture/](http://10.132.2.139:8080/venture/router)*[router](http://10.132.2.139:8080/venture/router)*

**url是指本接口调用的路径，本文提供的是开发测试环境，以后不同环境部署后会提供相关环境真正的url地址，建议接入系统能够以设计成可配置的，以便进行不同环境（开发测试环境、SIT环境、UAT环境、生产环境…）的使用**

**POST数据说明**

POST数据示例如下：

**系统参数据：**

|  |  |  |
| --- | --- | --- |
| 参数名称 | 是否必须 | 参数说明 |
| appKey | 是 |  |
| method | 是 | 服务方法名，取值：venture.totalprem |
| v | 是 | 服务方法的版本号，取值：1.0 |
| messageFormat | 否 | 通信报文格式，可选值为 xml 和 json，默认为 xml。 |
| locale | 否 | 本地化类型，默认为 zh\_CN。 |

**业务参数：**

"resultDetailed":[{"type":"寿险","totalAmnt ":"累计保额"," totalprem ":"累计保费"},{"type":"意外险","totalAmnt ":"累计保额"," totalprem ":"累计保费"},{"type":"重疾险","totalAmnt ":"累计保额"," totalprem ":"累计保费"},{"type":"防癌险","totalAmnt ":"累计保额"," totalprem ":"累计保费"},]

参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必须** | **说明** |
| type | 是 | 风险类别：A01-寿险、A02-意外险、A03-重疾险、A04-防癌险 |
| totalAmnt | 是 | 累计保额 |
| totalprem | 是 | 累计保费 |
|  | 是 |  |

# 接口调用JAVA例程说明

|  |
| --- |
| **public** **class** ClientDemo{  //服务接口的URL,建议可以配置  **private** **static** **final** String *TARGET\_URL*="http://localhost:8080/venture/router";  **private** **static** **final** String *CHARSET\_NAME* = "utf-8";//通讯的编码格式为utf-8字符集    /\*\*  \* 主调方法  \*/  **public** **static** **void** main(String[] args) {  //for(int i =0 ; i<10;i++){  **try** {  URL targetUrl = **new** URL(*TARGET\_URL*);  HttpURLConnection httpConnection = (HttpURLConnection) targetUrl.openConnection();  httpConnection.setDoOutput(**true**);  httpConnection.setRequestMethod("POST");//以post提交  //httpConnection.setRequestProperty("Content-Type", "text/html");//设置提交内容格式  httpConnection.setRequestProperty("Accept-Charset", *CHARSET\_NAME*);//设置提交的编码格式  httpConnection.setRequestProperty("contentType", *CHARSET\_NAME*);//设置提交的编码格式  httpConnection.setConnectTimeout(3000);//设置联接主机超时时间（单位：毫秒）  httpConnection.setReadTimeout(3000);//设置从主机读取数据超时（单位：毫秒）  httpConnection.setDoOutput(**true**);  Map<String,String> form = **new** HashMap<String,String>();  //组装请求报文  //组装系统级参数据，没顺序要求，先设置哪个都可以  form.put("appKey", "30");//应用键  form.put("method", "venture.totalprem");//服务方法名  form.put("v", "1.0");//服务版本号  form.put("messageFormat", "json");//响应报文格式  form.put("locale", "zh\_CN");//响应报文语言  //组装业务参数据  form.put("separationMain", *getSeparationMain*());  String params = *formatForm*(form);//格式化请求报文    //发送报文  OutputStream outputStream = httpConnection.getOutputStream();  outputStream.write(params.getBytes());  outputStream.flush();  //如果请求不成成功，返回200码，比果网络不通等情况  **if** (httpConnection.getResponseCode() != 200) {  **throw** **new** RuntimeException("Failed : HTTP error code : "  + httpConnection.getResponseCode());  }  //请求成功，成功或失败的情况  String responseStr = **new** String(*readInputStream*(httpConnection.getInputStream()), *CHARSET\_NAME*);//取出响应报文  System.*out*.println(responseStr);  httpConnection.disconnect();//关闭连接  /\*\*\*\*\*\*\*\*以下是如果服务失败时的报格式\*\*\*\*\*\*\*\*  {"errorToken":"@@$-ERROR\_TOKEN$-@@","code":"1","message":"服务不可用","solution":"服务目前无法使用，请保存好错误信息并和服务平台管理员联系","subErrors":[{"code":"isp.tax-searation-service-timeout","message":"调用tax.searation服务超时，该服务的超时限制为10秒，请和服务平台提供商联系。"}]}  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  } **catch** (MalformedURLException e) {  //请各系统自行处理导常，例程只是打印在控制台，请不要参照。  e.printStackTrace();    } **catch** (IOException e) {  //请各系统自行处理导常，例程只是打印在控制台，请不要参照。  e.printStackTrace();    }  //}  }    /\*\*  \* 格式化表单数据  \* **@param** form  \* **@return**  \* **@throws** UnsupportedEncodingException  \*/  **public** **static** String formatForm(Map<String,String> form) **throws** UnsupportedEncodingException{  StringBuilder builder = **new** StringBuilder();  Iterator<Entry<String, String>> it = form.entrySet().iterator();  String key;  String value;  **while**(it.hasNext()){  Map.Entry<String, String> entry = (Map.Entry<String, String>)it.next();  key=entry.getKey();  value=entry.getValue();  builder.append(URLEncoder.*encode*(key, *CHARSET\_NAME*));  **if**(value != **null**){  builder.append('=');  builder.append(URLEncoder.*encode*(value, *CHARSET\_NAME*));  }  **if**(it.hasNext()){  builder.append('&');  }  }  **return** builder.toString();  }    /\*\*  \* 生成请求费用主信息JSON字符串  \*/  **public** **static** String getSeparationMain(){  StringBuffer str = **new** StringBuffer();  str.append("{\"riskCode \":\"2014\",\n");//险种代码  str.append("\"name \":\"张三\",\n");//费用类型  str.append("\"sex \":\"0\",\n");//业务主键  str.append("\"birthDay \":\"2016-01-01\",\n");//生日  str.append("\"idType \":\"01\",\n");//证件类型  str.append("\"idNo \":\"12345678\",\n");//证件号码  str.append("\"amnt \":\"10000\",\n");//保额  str.append("\"prem \":\"20000\",\n");//保费  **return** str.toString();  }    /\*\*  \* 从输入流中读取数据  \* **@param** inStream  \* **@return**  \* **@throws** Exception  \*/  **public** **static** **byte**[] readInputStream(InputStream inStream) **throws** IOException{  ByteArrayOutputStream outStream = **new** ByteArrayOutputStream();  **byte**[] buffer = **new** **byte**[1024];  **int** len = 0;  **while**( (len = inStream.read(buffer)) !=-1 ){  outStream.write(buffer, 0, len);  }  **byte**[] data = outStream.toByteArray();//网页的二进制数据  outStream.close();  inStream.close();  **return** data;  }  } |
|  |