基于XML的网关报文生成解析模板引擎算法

C-3 创建:汤新奇, 最后修改: 汤新奇 今天 23:10

日录 tangxingi

1、网关报文分析

2、基于XML的词法设计

3、基于XML的语法设计

序列化语法描述 反序列化语法描述

4、TTM框架 +angXinqi

1、网关报文分析

ngxinqi 5172776 tangxinqi 45172776 -3 C-3

tangxinqi 45172776 C-3

tangxinqi 45172776 tangxinqi 45172776 C-3

tangxino 4517277 C-3

tangxingi 45172776

tangxindi 45172776

目前网关使用的报文形式具有多样化的特性,主流包括XML、JSON,KEY-VALUE等。如何通过一套模板引擎来自动化的生成或者解析目标报文,接下来将详细阐述。以XML报文(实际可混合使用)为例,

ngXII	-1	tangx	tangx		tangx		tangx		tangx	
51727	70人代	45172776	45172776		45172776		45172776		45172776	XML
3-3	1	<pre><?xml version="1.0" encoding="utf-8"?></pre>	C-3		C-3		C-3		C-3	
	2	<cms></cms>								
	3	<eb>xinql</eb>	d l	tangxing!		tangxing		tangxing		tangxing
	4	45 *Pub> 451727		A5172776		45172776		45172776		A5172776
	5	<pre>C-3 <transcode1>SZFH_AXZF1</transcode1></pre>	21>	C-3		C-3		C-3		C-3
	6	<pre><transcode>SZFH_AXZF</transcode></pre>								
-vir	ng i 7		zvinai		zvinai		-vinqi		zvinai	
ing ^ .	768	<in> tang^-</in>	tang 172776		tang^		tang 7.		tang x	
3-3	9	<trxcode>10004</trxcode>	G-3		G-3		C-3		G-3	
	10		G 3		0.3		G G		6. 3	
	11					-1		-1		
	12	<(cms>xing) tangxin	16	tangxing tangx		tangxing1		tangxing1		tangxing tangxing
	分析报	文的生成步骤如下:		G-3		G-3		G-3		C-3

• 步骤一:生成TransCode节点: <TransCode1>SZFH_AXZF1</TransCode1>和<TransCode>SZFH_AXZF</TransCode>

步骤二:生成pub节点: <pub><TransCode>SZFH_AXZF</TransCode></pub>

- 步骤三: 生成Trxcode节点: <Trxcode>10004</Trxcode>
- 步骤四: 牛成in节点: <in> <Trxcode>10004</Trxcode></in>
- 步骤五:生成eb节点: <eb><pub><TransCode>SZFH_AXZF</TransCode> </pub><in> <Trxcode>10004</Trxcode></in></eb>
- 步骤六: 生成CMS节点: <CMS><eb><pub><TransCode>SZFH_AXZF</TransCode></pub><in><Trxcode>10004</Trxcode></in></eb></CMS>

angxing1 5172776 tangxino 4517277

通过分析可以得出,整体报文生成需要 步骤六->步骤五->步骤四->步骤三->步骤二->步骤 (其中->表示依赖关系), 当然步骤N不限于节点生成, 还可以包含加密、签名验签等。 tangxind tangxin 如果将上面步骤进行语义转化为如下图: 生成 CMS子 节点 tangxinqi 45172776 生成pub 生成in节 节点 点 生成 生成 TransCode 1 **TransCode** Trxcode 节点 节点 节点 报文生成- 语法树

问题点提出:如何定义一套语言及语法规则, tangxinqi tangxingi

2、基于XML的词法设计

- 序列化关键字: template-request、template-text、template-map、template-json、template-xml、template-object、template-array、foreach、key
- 反序列化关键字: template-response、template-text、template-xml、template-object、template-array、value、result-set、result-key

3、基于XML的语法设计

• 序列化语法描述

关键词	关键字指令ngi	关键字可接受指令	属性属性	含义	属性值	tangxinqi	
template-request	序列化指令	template-text		45172176 G-3	45172 <u>7</u> 76 C-3	45172776 G-3	
		template-map					
tangxing! A5172776	t	angxind (template-json	tangxing1 45172776	tang× 4517	ing1 2776	tangxing1 A5172776	

2020/7/11					基于XML的网关报文生成解析模板引擎算法			
tangxir 451727	iai 76	tangxingi 45172776	template-xml template-object		451/21	-	tangxinqi 45172776	
C-3		C-3	template-array		C-3		C-3	
	template_text, x i ng l 45172776	生成文本指令: tangxin 451727	文本、\${param}	id tangxi 45172	指令返回值ID, 可通过\$ID获取元素gx inq i	需全局唯一 tangx inq l 45172776		tangxingi 45172776
	C-3	指令处理文本指令返回文本	template-text	t-if		param != null		C-3
noxir	nai	ringi	template-json		- noxingi	xinai	ingxingi	
tangxir 451727 C-3	76	tang A 5 172776 45172776 C-3	template-xml72776	ignore	忽略指令结果值传递给父指令,其他指令可通过ID获取	false, true	tangxing1 45172776 C-3	
	tangxinqi 45172776	tangxin	template-array foreach	type-handler	指令返回值后置处理器 176 +angXinqi 176	handler的name,需全局唯一 tangX inq		tangxingi 45172776
	template-map	生成map指令 C-3	key	id C-3	指令返回值ID, 可通过\$ID获取元素3	需全局唯一 C−3		G-3
tangxir	nai	• 指令处理key-value对象	tangxind	t-if	判断语句,支持 == 、!= 、>= 、<= 、> 、< 、&& 、 、	param != null	. anaxinai	
451727 C-3	76	• 指令返回Map	45172776 6-3	ignore	忽略指令结果值传递给父指令,其他指令可通过ID获取	false, true	tangxingi 45172776 C-3	
0	template-json	生成json指令	key	id	指令返回值ID, 可通过\$ID获取元素	需全局唯一		
	tangxingi 45172776	• 指令处理key-value对象 Xin	d1 16	t-if tangxi	判断语句,支持 ==、!=、>=、<=、>、<、&&、 、	param != nultangxing1		tangxingi 45172776
	C-3	• 指令返回json字符串		ignore G-3	忽略指令结果值传递给父指令,其他指令可通过ID获取	false, true C-3		C-3
tangxir	nai	tangxingi	tangxing	type-handler	指令返回值后置处理器	handler的name,需全局唯一	tangxingi	
451727 C-3	template-xml	生成xml指令	文本、\${param}	id	指令返回值ID, 可通过\$ID获取元素	需全局唯一	tangxing1 45172776 C-3	
0		• 指令处理key-value对象	key	t-if	判断语句,支持 ==、!=、>=、<=、>、<、&&、 、	param != null		
	tangxing1 45172776	• 指令返回xml字符串 tangxin 451727	foreach	ignoretangxi	忽略指令结果值传递给父指令,其他指令可通过ID获取	false, true tangxing1		tangxingi 45172776
	C-3	C-3		type-handler	指令返回值后置处理器 0-3	handler的name,需全局唯一		C-3
tangxir	template-object	生成bean指令	key tangxind	id	指令返回值ID, 可通过\$ID获取元素	需全局唯一	tangxingi	
tang x 451727	70	• 指令处理key-value对象	45172776 C-3	t-if	判断语句,支持 ==、!=、>=、<=、>、<、&&、 、4517	param != null	tangxing1 45172776 C-3	
		• 指令返回bean对象		type	对象类型	类全路径名		
	tangxingi 45172776	tangxin 451727	16	ignoretangxi 45172	忽略指令结果值传递给父指令,其他指令可通过ID获取	false, true tangxingi 45172776		tangxingi 45172776
	C-3	C-3		C-3	C-3	C-3		C-3

指令返回值后置处理器

数组字符串的分隔符

数组字符串的开头

数组字符串的结尾

判断语句,支持 ==、!=、>=、<=、>、<、&&、||、

key的具体值

type-handler

separator

open

close

t-if

name tang)

template-object

template-array

文本、\${param}

template-text

template-map

template-ison

template-xml

template-object

template-array

key

foreach

tangxindi

tangxindi

key

key-value指令

• 构造key-value对象 0-3

handler的name, 需全局唯一

param != null angx inq i

45172776

字符串

字符串

字符串

字符串

tangx

tangxingi 45172776 C-3

key tangxinqi 45172776 C-3 tangxinal 45172776 C-3 tangxinai

• 反序列化语法描述

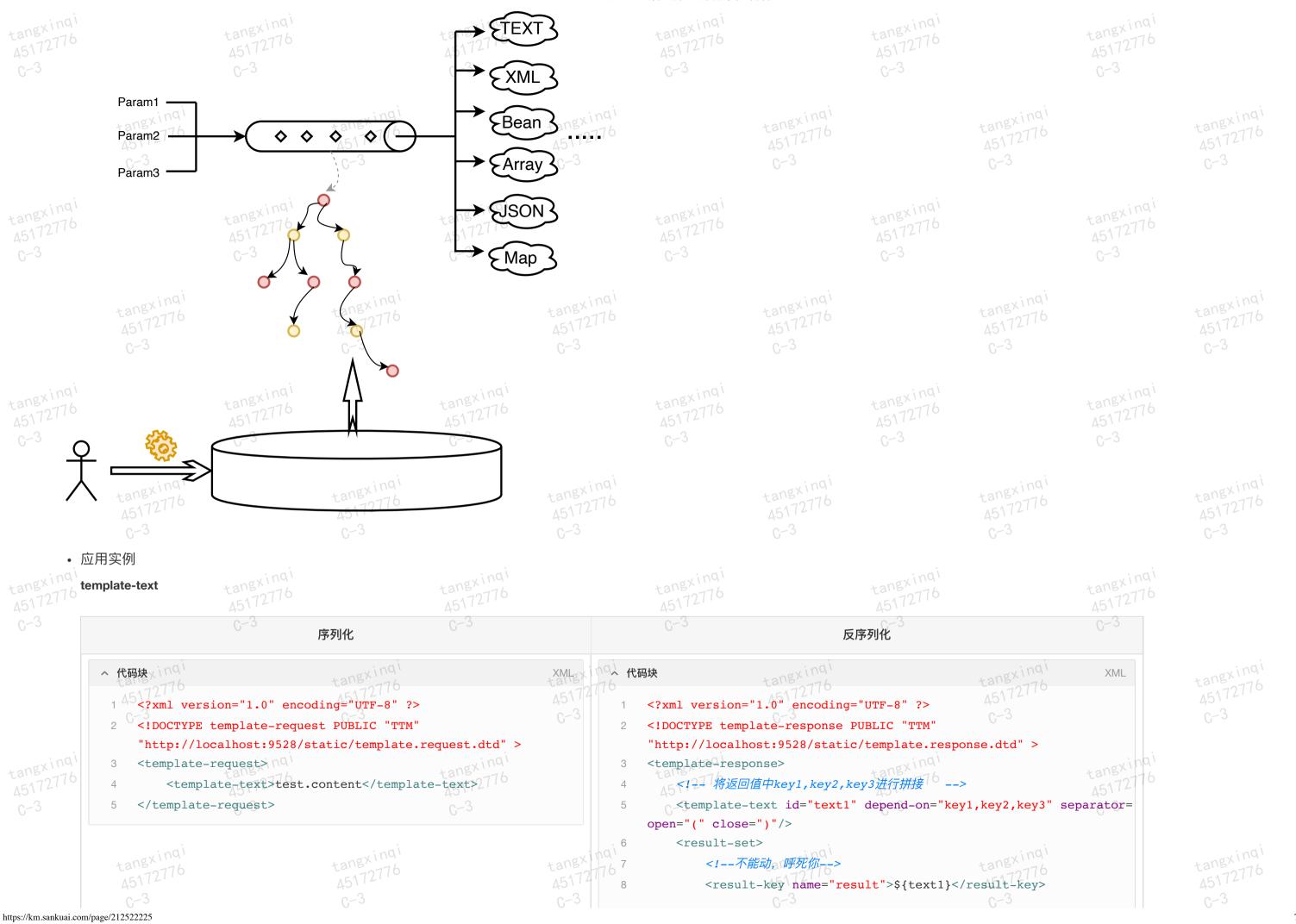
	关键词 tangxingi	关键字指令	tangxingi	关键字可接受指令	属性	i属性含义 tangxingi	属性值 tangxingi		tangxingi
	template-response	反序列化指令	45172776 C-3	template-text	- 4517277 C-3	+ tarish - 45172776 C-3	45172776 - G-3		tang A 45172776 C-3
				template-xml					
5Xir 727	76	tangxingi 45172776		template-object		tangxinqi 45172776 tangx 4517	(ina) 2776	tangxingi 45172776	
3		C-3		template-array		C-3		C-3	
	tangxingi		tangxingi	result-set	tangxino	tangxingi	tangxingi		tangxing
	template-text	解析文本指令	45172776 C-3	template-text	id 45\7277	指令返回值ID, 可通过\$ID获取元素	需全局唯一 45172176		tangxingi 45172776 C-3
		• 指令受理字符串		template-xml	t-if	判断语句,支持 ==、!=、>=、<=、>、<、&&、 、	param != null		
5Xir 727	ai 76	• 指令返回字符串 45/72776		tangxingi 45172776	depend-on	指令入参,可接受多个,","分隔 45\\\	需全局唯一	tangxingi 45172776	
3		C-3		C-3		open+\$key+separator+\$key+close拼接		C-3	
	agxingi		tangxingi	template-array	type-handler	指令入参前置处理器	handler的name,需全局唯一		axinai
	tangxingi 45172776 C-3		45112		separator 277	数组字符串的分隔符 45172776	字符 tangx ind 1 字符 45172776		tangxingi 45172776 C-3
	6-3		C-3	template-object	open	数组字符串的开头	字符串		G-3
5Xir 727	iai 76	tangxingi		tangxinqi 45172776	close	数组字符串的结尾 tangX	字符串	tangxinqi 45172776	
3	template-xml	解析xml指令		value G-3	id	指令返回值ID, 可通过\$ID获取元素	需全局唯一	G-3	
	, angxingi	指令受理xml字符指令返回Docume			t-if	判断语句,支持 ==、!=、>=、<=、>、<、&&、 、	nexingi		Longxingi
	tangxingi 45172776	• 指令返回Docume	45112		depend-on	tangx inq	节点ID或者参数key		tangxingi 45172776
	C-3		C-3		type-handler	指令入参前置处理器	handler的name,需全局唯一		C-3
5Xir	template-object	解析Bean指令		value angxingi 45172776	id	指令返回值ID,可通过\$ID获取元素	需全局唯一	tangxingi 45172776	
3		• 指令接受json字符	F串或者Map	C-3	t-if	判断语句,支持 ==、!=、>=、<=、>、<、&&、 、3	param != null	C-3	
	, angxingi	• 指令返回对象	Langxingi		depend-on	指令入参,不支持多个	节点ID或者参数key		kongxingi
	tangxingi 45172776 C-3		tangxing1 45172776 C-3		type-handler	指令入参前置处理器 45172776	handler的name,需全局唯一		tangxingi 45172776 C-3

020/7/11				麦	基于XML的网关报文生成解析模板引擎算法			
tangxi	nai	Langxingi	naxinai	type	对象类型,默认为HashMap	类全路径名	ingxinqi	
45114	template-array	解析Array指令	value45172776	id	指令返回值ID, 可通过\$ID获取元素	7 需全局唯一	tangxing1 45172776	
C-3		• 指令接受数组或者数组字符串	C-3	t-if	判断语句,支持 ==、!=、>=、<=、>、<、&&、 、	param != null	C-3	
	tangxingi 45172776	• 指令返回数组 tangXingi 45172776		depend-on	指令入参,不支持多个 tangx inq i	节点ID或者参数key,		tangxingi 45172776
	C-3	C-3		type-handler	指令入参前置处理器	handler的name,需全局唯一		C-3
tangxi	nai	tangxinqi	tangxinqi	separator	数组字符串的分隔符	字符	tangxingi	
451727 C-3	76	tangxing1 45172776 C-3	tang 7 45172776 C-3	open	数组字符串的开头 451	字符串	tangxing1 45172776 - C-3	
G 3			U U	close	数组字符串的结尾	字符串		
	value tangxingi 45172776	取值指令 tangxingi 45172776	空	id tangxino	指令返回值ID, 可通过\$ID获取元素	需全局唯一 _{tangxing} i		tangxinqi 45172776
	C-3	C-3		path C-3	取值路径 0-3	path1.path2.path3		G-3
tangxii 451727 C-3	result-set	返回数据指令	result-key tangxingi 45172776 C-3	type	对象类型,默认为HashMap +ang ^X +ang 45172776 451		tangxinqi 45172776 C-3	
	result-key	返回数据映射指令	文本、\${param}	name	对应的result-set的属性名称			
	tangxing1 45172776	• 将数据放入目标对象中的属性中		tangxind 4517277	tangxing1 45172776	tangxingi 45172776		tangxingi 45172776
	C-3	C-3	-	C-3	C-3	C-3		C-3

tangXi4\ TTM框架

• 简介: **TTM**(template to message) 框架是基于上述设计的xml模板语言,通过XML模板配置,翻译成JAVA可执行的目标代码,通过目标程序自动化生成或者解析目标数据,目标数据涵盖: TEXT、XML、JSON、Bean、Array、Map等类型。

		L Boart, Array	いぬりの人主。								
	架原理: tangxingi 45172776 C-3		tangxindi 45172776 C-3		tangxingi 45172776 C-3		tangxindi 45172776 C-3		tangxinqi 45172776 C-3		tangxino 4517277 C-3
tangxingi 45172776 C-3		tangxinqi 45172776 C-3		tangxingi 45172776 C-3		tangxingi 45172776 C-3		tangxingi 45172776 C-3		tangxingi 45172776 C-3	
	tangxinqi 45172776		tangxinqi 45172776		tangxinqi 45172776		tangxinqi 45172776		tangxinqi 45172776		tangxino 4517277



```
<?xml version="1.0" encoding="UTF-8" ?>
                                                                              <?xml version="1.0" encoding="UTF-8" ?>
     <!DOCTYPE template-request PUBLIC "TTM"</pre>
                                                                              <!DOCTYPE template-response PUBLIC "TTM'
     "http://localhost:9528/static/template.request.dtd" >
                                                                              "http://localhost:9528/static/template.response.dtd" >
                                                                              <template-response>
     <template-request>
         <template-xml>
                                                                                  <template-xml>
 5
6 tangxing | <![CDATA[<?xml version="1+00" encoding="UTF-8" ?>]]>
                                                                                      <value id="test1" path="xml.test1.test1 test1"/>
   45172776 <key name="xml">
                                                                                      <value id="test2" path="xml.test2"/>
                                                                                      <value id="test3" path="xml.test3"/>
                 <key name="test1">/
                     <key name="test1_test1">content1</key>
                                                                                  </template-xml>
                     <key name="test1_test2"/>
                                                                                  <result-set>
                 </key>
                                                                                      <result-key name="test1">${test1}</result-key>
                 <key name="test2">content2</key>
                                                                                      <result-key name="test2">${test2}</result-key>
                 <key name="test3">content3</key>
                                                                                      <result-key name="test3">${test3}</result-key>
 13
                 <key name="test4">content4</key>
                                                                         13
                                                                                  </result-set>
tangxingi </key>
                                                                              </template-response>
       727</template-xml>
 16 </template-request>
```

template-array

```
45172776
                                                                             4517271
                                               4517211
                                                                                                                                         451721
                               序列化
                                                                                                        反序列化
へ 代码块
                                                                XML
                                                                        へ 代码块
                                                                                                                                            XML
    <?xml version="1.0" encoding="UTF-8" ?>
                                                                             <?xml version="1.0" encoding="UTF-8" ?>
    <!DOCTYPE template-request PUBLIC "TTM"</pre>
                                                                             <!DOCTYPE template-response PUBLIC "TTM"</pre>
     "http://localhost:9528/static/template.request.dtd" >
                                                                              "http://localhost:9528/static/template.response.dtd" >
     <template-request>
                                                                             <template-response>
                                                                             template-array separator="\| open="(" close=")">
         <template-array separator="|" open="(" close=")">
             <template-text>${param1}</template-text>
                                                                                     <value id="value1" path="$[0]"/>
             <template-text>${param2}</template-text>
                                                                                     <value id="value2" path="$[1]"/>
             <template-text>${param3}</template-text>
                                                                                     <value id="value3" path="$[2]"/>
                                                             tangxindi
            <template-text>${param4}</template-text>
                                                                                     <value id="value4" path="$[3]"/>
        |</template-array>
                                                                                 </template-array>
10 </template-request>
                                                                                 <result-set>
```



 $\frac{C-3}{C-3}$

```
| PRONC | PRO
```

• 框架特性 (VS"友商")

1. 去中心化

TTM提供统一模板源(TemplateSource)接口,开发者可在不同场景下实现不同的子类,从而在不同的场景下加载不同的模板。TTM默认提供了ClassPathResourceTemplateSource子类实现,可将本地resources文件下作

2. 灵活性强

TTM提供标准指令切入点type-handler,开发者可通过实现TypeHandler接口,完成特定指令序列操作(例如先加密后Base64、先Base64后加密)

3. 三重防错机制 (100)

第一重机制:DTD文件-TTM提供标准的XML模板DTD校验文件,可以很好的控制模板格式,同时可以提示开发者具体指令详情,从而能够快速准确的配置模板

第二重机制:jar包版本控制-TTM通过严格的RELEASE版本控制,各个服务之间可以控制pom版本依赖来规避修改带来的问题,同时独立的TypeHandler子类实现,满足设计模式的"开闭原则"和"里氏替换原则"

第三重机制:由于"去中心化"特性,QA可通过引入jar包方式进行简单的二次开发,从而可以依次回归序列化和反序列化的各种case,保证上线前的序列化和反序列化的准确性

45172774. 具有可视化

基于XML的模板有大量成熟编辑器编辑(例如idea),提升了开发者的方便性和灵活性;另外统一格式且一体化的模板文件,可以很直观的看出目标结果的具体结构,这有利于问题排查和后期维护。于此同时,不需要投入 大量人力开发前端工具,可节省大量开发和维护人力。

• TTM接入-SpringBoot方式

https://km.sankuai.com/page/212522225

```
基于XML的网关报文生成解析模板引擎算法
tangxingi 1、pom依赖引入
                                                                                     tangxingi
                                                                                                                                             tangxingi
                                                        tangxingi
                                                                                                                 tangxingi
                            tangxingi
            へ 代码块
                                                                                                                                                               XML
                <dependency>
                    <groupId>com.txq.pay</groupId>
               tangx <artifactId>message-boot-starter</artifactId>
             4 /5/7 version>1.0.0-SNAPSHOT
             5 </dependency>
tangxingi 2、yml配置
                            tangxingi
                                                        tangxinqi
                                                                                                                                             tangxingi
                                                                                    tangxingi
           へ 代码块
                                                                                                                                                               XML
             1 ttm:
                  # 开启ttm
              enable: true
               # 初始化加载模板列表
             5 templates: anxin.send,anxin.send.response
          3、java引用
                                                                                                                                             tangxingi
                                                                                                                 tangxingi
           へ 代码块
                                                                                                                                                               Java
                @RunWith(SpringRunner.class)
                @SpringBootTest
                public class MessageTest {
                private MessageSerializer serializer;
                    @Autowired
                    public void setSerializer(MessageSerializer serializer) {
                        this.serializer = serializer;
            11
               A5/7 public void testMessageSerialize() {
                        ChannelContext context = new ChannelContext();
            15
                        context.setGwOutNo("1234567890");
                        context.setRecAccountNo("222222222222");
                        context.setRecAccountName("QDB");
            17
                        context.setRecBankNo("102100099996");
            18
                        context.setAccountNo("987654321");
            19
            20
                        context.setAmount(12345L);
            21 tangxing
                        context.setComment("TEST_COMMENT");
                        context.setTradeDate(new Date());
                        Map<String, String> result = serializer.serialize("anxin.send", context);
```

11/12

```
log.info("result is {}", result);
                              tangx
            25
                     }
            26
            27
                     @Test
                     public void testMessageDeserialize() {
            28
               String value = "reqData=PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0idXRmLTgiPz4KCjxDTVM+CiAgPGViPgogICAg\n" +
                                 "PHB1Yj4KICAqICAqPFRyYW5zQ29kZT5YWFq8L1RyYW5zQ29kZT4KICAqICAqPENJUz5YWFq8L0NJ\n" +
                                 "Uz4KICAqICAqPEJhbmtDb2RlPlhYWDwvQmFua0NvZGU+CiAqICAqIDxJRD5YWFq8L01EPqoqICAq\n" +
            31
            32
                                 "ICA8VHJhbkRhdGU+WFhYPC9UcmFuRGF0ZT4KICAgICAgPFRvYW5UaW11P1hYWDwvVHJhblRpbWU+\n" +
                                 "CiAgICAgIDxmU2Vxbm8+WFhYPC9mU2Vxbm8+CiAgICAgIDxSZXRDb2RlPjAwMDA8L1JldENvZGU+\n" +
            33
                              "CiAqICAqIDxSZXRNc2c+vbvS17PJuaY8L1JldE1zZz4KICAqIDwvcHViPqoqICAqPG91dD4KICAq\n" +
            34
                               "ICAgPENvbW1zZXE+WFhYPC9Db21tc2VxPgogICAgICA8VHJhbnNlcT4xMjM0NTY3ODkwOTg3NjU0\n" +
            35
                                 "MzI8L1RyYW5zZXE+CiAqICAqIDxNc2dzdGF1cy8+CiAqICA8L291dD4KICA8L2ViPqo8L0NNUz4=";
            36
                         Map<String, String> result = serializer.deserialize("anxin.send.response", value);
            37
                                                                                                        tangxinqi
                                                                          tangxinqi
            38 tangxing
                         log.info("response is {}", result);
                         value = "errorCode=FT#0999999";
            40
                         result = serializer.deserialize("anxin.send.response", value);
            41
                                                           tangxingi
                         log.info("response is {}", result);
             43
                     }
             44
                }
                                                                           2 仅供内部使用,未经授权,切勿外传
https://km.sankuai.com/page/212522225
```