环境介绍

开发环境:

jdk8

zookeeper3.4.14

activemq 5.15.9

camel 3.0.0-M2

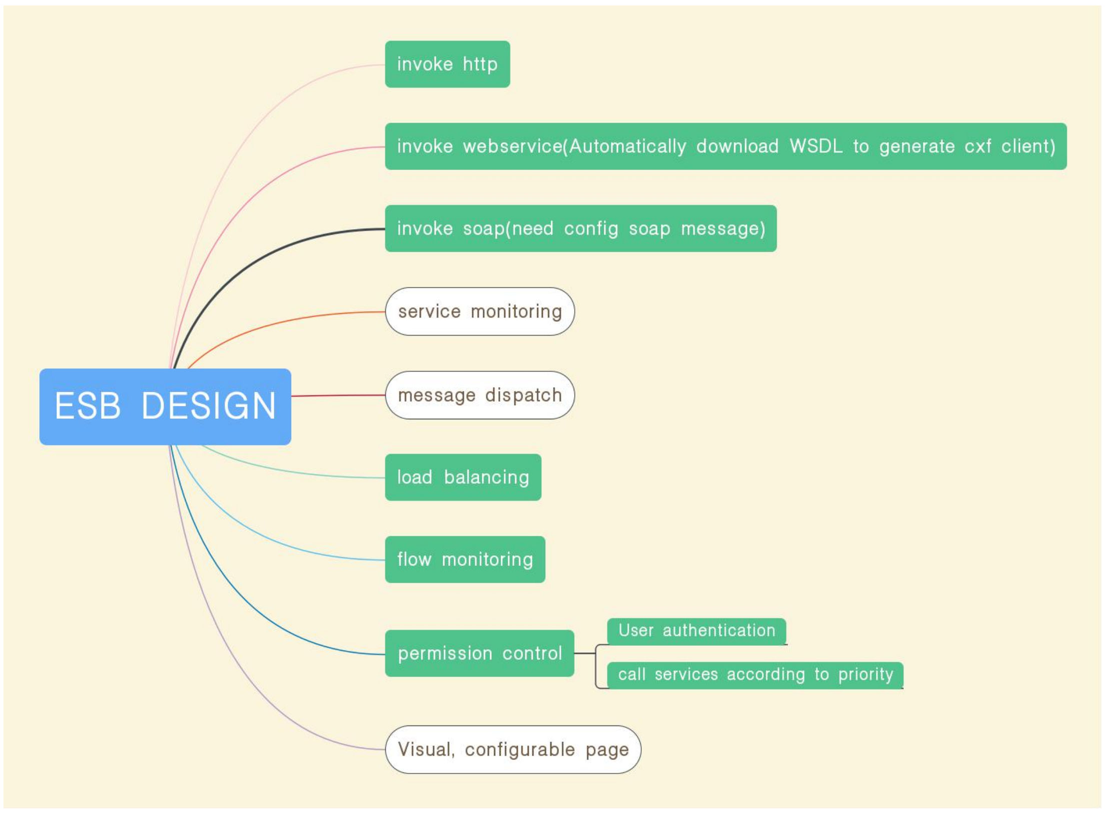
开发工具:

eclipse

框架:

基于 springcloud

功能脑图(绿色节点是已经完成的):


ESB DESIGN
invoke http
invoke webservice(Automatically download WSDL to generate cxf client)
invoke soap(need config soap message)
service monitoring
message dispatch
load balancing
flow monitoring
permission control
User authentication
call services according to priority
Visual, configurable page

功能简介：

服务总线，目前提供 http,webservice,soap 三种形式的调用;文件和图片要转为二进制再进行传输，目前服务总线采用两级的接口即 site/service,site code 是指站点，service 是指某个站点下面的具体服务器，显然 site/service 路径是唯一的。

注册服务和调用服务都支持 json 格式和 xml 格式的调用,以哪种格式调用就以哪种格式返回

删除服务仅支持 json 格式调用

用户注册时普通的 http 调用

调用手册

调用流程

第一步要先注册用户，根据 user register 接口传参数进行注册,注册完后会返回 token,这个 token 要在后续加在头参数中具体看调用文档。

第二部注册服务，把你目标的接口资源配进来，例如目标资源 hello:

```
@ResponseBody
@RequestMapping("/hello")
public String hello(String arg1, String arg2){
}
```

那么要把这个 http://localhost:8080/hello 这个地址(url 关键字)配进去,因为他有两个参数,且均是 string 类型，以 XML 为例

```
<param>
<key><![CDATA[arg0]]></key>
<type>1</type>
</param>
<param>
<key><![CDATA[arg1]]></key>
<type>1</type>
</param>
```

调用最好也要按顺序调用

如果是 webservice 则要配置 wsdl 地址

注册时要向头中添加 Authorization,其值为 token

第三步调用服务

调用服务主要配置 site 和 service 和参数,如果是无参数的资源,可以忽略 param 关键字,还是以第二部中的 hello 为例，且格式为 XML

```
<root>
<siteCode>site code</siteCode>
<serviceCode>service code</serviceCode>
<offline>离线</offline>
<params>
<value><![CDATA[value 1]]></value>
<key><![CDATA[arg0]]></key><!--这里一定时 arg0,需要跟资源参数名对上 -->
<params>
```

```
    <params>
    <value><![CDATA[value 2]]></value>
    <key><![CDATA[arg1]]></key><!--这里一定时 arg1,需要跟资源参数名对上 -->
    <params>
  </root>
```

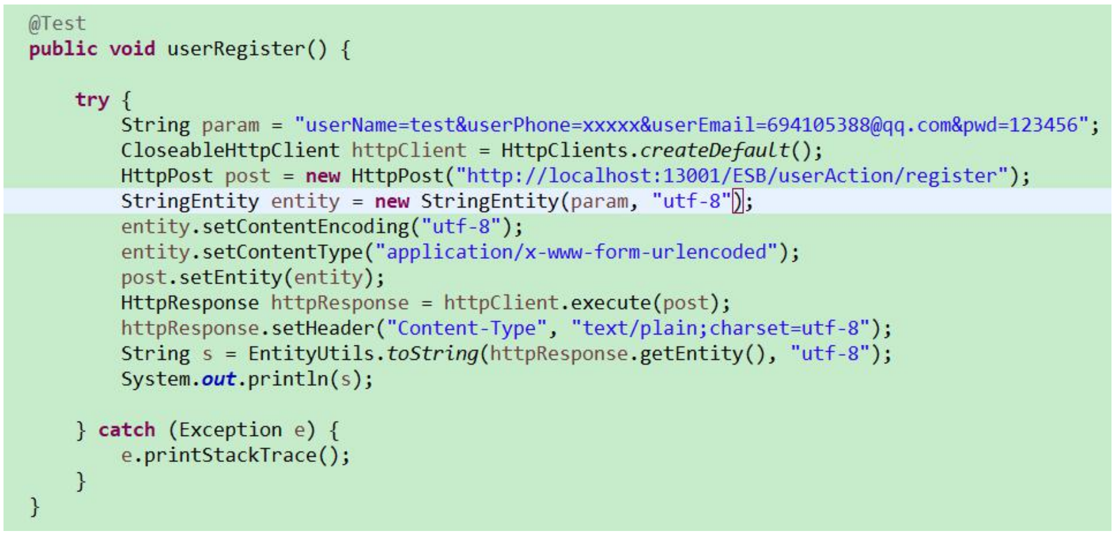
调用时要向头中添加 Authorization,其值为 token

以上说明的接口如下所示，每个接口下面都附有测试代码的路径

用户注册并获得 token

	user register			
Request	Direction	Terminal ----> WEB Server		
	transfer protocol	http/post,get		
	transmission format			
	interafce name	/userAction/register		
	param	description	necessary	param type
	userName	user name	True	string
	userPhone	user phone	True	string
	userEmail	email	True	string
	encrypt	No		
	test address			
remark				
Response	Direction	WEB Server ----> Terminal		
	transfer protocol	http /JSON		
	encrypt	No		
	return description	{“code”:100, “token”:”token”}		
	param	description	param type	
	code	100:success	string	
	token	token	string	
remark	①code : 100:success;101:false;			

code path: test.com.camel.RegisterJsonTest.java



```java
@Test
public void userRegister() {

    try {
        String param = "userName=test&userPhone=xxxxx&userEmail=694105388@qq.com&pwd=123456";
        CloseableHttpClient httpClient = HttpClients.createDefault();
        HttpPost post = new HttpPost("http://localhost:13001/ESB/userAction/register");
        StringEntity entity = new StringEntity(param, "utf-8");
        entity.setContentEncoding("utf-8");
        entity.setContentType("application/x-www-form-urlencoded");
        post.setEntity(entity);
        HttpResponse httpResponse = httpClient.execute(post);
        httpResponse.setHeader("Content-Type", "text/plain;charset=utf-8");
        String s = EntityUtils.toString(httpResponse.getEntity(), "utf-8");
        System.out.println(s);

    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

服务注册

json 格式注册

	register service with json			
Request	Direction	Terminal ----> WEB Server		
	transfer protocol	http/post		
	transmission format	Json		
	interface name	/registerAction/registerWithJson		
	param	description	necessary	param type
	siteCode	site code	Yes	string
	serviceCode	service code	Yes	string
	url	resource;if type=2 url=wsld	Yes	string

	type	1:http;
		2:webservice;
		3:soap
		4:upload file
			Yes	int

	params	param key	No	string
	soap	it is necessary when type=3 时参数用问号代替	No	string
	encrypt	No		

	test address		

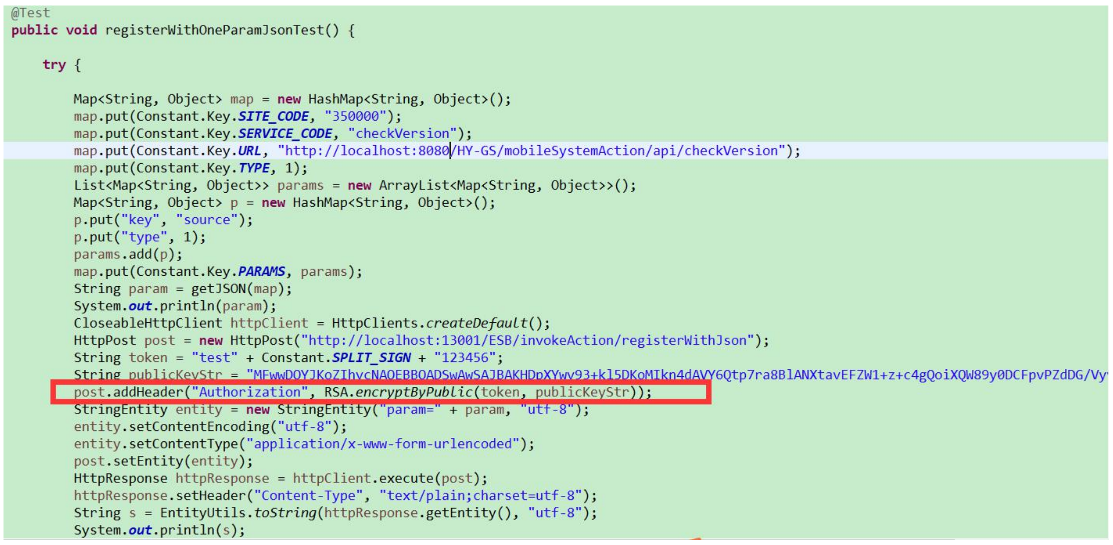
remark	param={
	"siteCode":"site code",
	"serviceCode":"service code",
	"url":resources",
	"type":1,//1:http;2:webservice;3:soap"
	params":[{"key":"","type":1},{}...{}]//1:text
	}
	put token:Authorization=token into head		

Response	Direction	WEB Server ----> Terminal	
	transfer protocol	http +JSON	
	encrypt	No	

	return description	{ “code” :100, “desc” :” success”]}
		{ “code” :101, “desc” :” failure” }]}	

	参数	说明	参数类型
	code		string
remark	①code : 100:success;101:failure;		

code path: test.com.camel.RegisterJsonTest.java



```java
@Test
public void registerWithOneParamJsonTest() {

    try {

        Map<String, Object> map = new HashMap<String, Object>();
        map.put(Constant.Key.SITE_CODE, "350000");
        map.put(Constant.Key.SERVICE_CODE, "checkVersion");
        map.put(Constant.Key.URL, "http://localhost:8080/HY-GS/mobileSystemAction/api/checkVersion");
        map.put(Constant.Key.TYPE, 1);
        List<Map<String, Object>> params = new ArrayList<Map<String, Object>>();
        Map<String, Object> p = new HashMap<String, Object>();
        p.put("key", "source");
        p.put("type", 1);
        params.add(p);
        map.put(Constant.Key.PARAMS, params);
        String param = getJSON(map);
        System.out.println(param);
        CloseableHttpClient httpClient = HttpClients.createDefault();
        HttpPost post = new HttpPost("http://localhost:13001/ESB/invokeAction/registerWithJson");
        String token = "test" + Constant.SPLIT_SIGN + "123456";
        String publicKeyStr = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAKHDpXYwv93+kl5DKoMIkn4dAVY6Qtp7ra8BlANXtavEFZW1+z+c4gQoiXQW89y0DCFpvPZdDG/Vy
        post.addHeader("Authorization", RSA.encryptByPublic(token, publicKeyStr));
        StringEntity entity = new StringEntity("param=" + param, "utf-8");
        entity.setContentEncoding("utf-8");
        entity.setContentType("application/x-www-form-urlencoded");
        post.setEntity(entity);
        HttpResponse httpResponse = httpClient.execute(post);
        httpResponse.setHeader("Content-Type", "text/plain;charset=utf-8");
        String s = EntityUtils.toString(httpResponse.getEntity(), "utf-8");
        System.out.println(s);
```

xml 格式调用

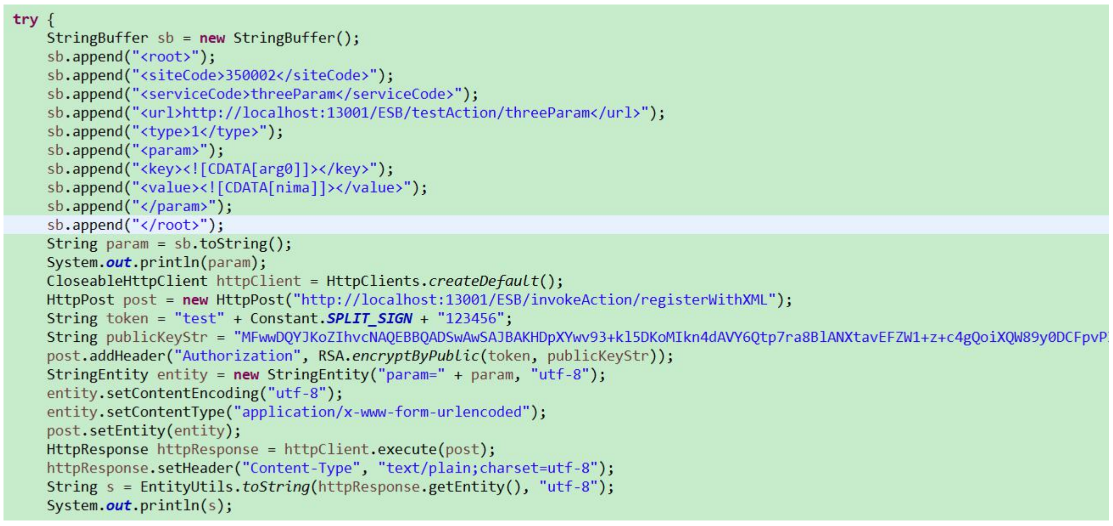
	register service with xml			
Request	Direction	Terminal ----> WEB Server		
	transfer protocol	http/post		
	transmission format	Json		
	interface name	/registerAction/registerWithXML		
	param	description	necessary	param type
	siteCode	site code	Yes	string
	serviceCode	service code	Yes	string
	url	resource;if type=2 url=wsld	Yes	string

	type	1:http;
		2:webservice;
		3:soap
			Yes	int

	param	param key	No	string
	soap	it is necessary when type=3 时参数用问号代替	No	string
	encrypt	No		
	test address			
remark	param=<root> <siteCode>site code</siteCode> <serviceCode>service code</serviceCode> <url>rescource</url> <type>1:http;2:webservice;3:soap</type> <param> <key><![CDATA[param 1]]></key> <type>1:text;2:file</type> </param> <param> <key><![CDATA[param 2]]></key>			

	<type>1:text;2:file</type> </param> </root> put token:Authorization=token into head		
Response	Direction	WEB Server ----> Terminal	
	transfer protocol	http +JSON	
	encrypt	No	
	return description	<root> <code>100</code> <desc>success</desc> </root> <root> <code>101</code> <desc>failure</desc> </root>	
	参数	说明	参数类型
	code		string
remark	①code : 100:success;101:failure;		

code path: test.com.camel.RegisterXMLTest.java



```
try {
    StringBuffer sb = new StringBuffer();
    sb.append("<root>");
    sb.append("<siteCode>350002</siteCode>");
    sb.append("<serviceCode>threeParam</serviceCode>");
    sb.append("<url>http://localhost:13001/ESB/testAction/threeParam</url>");
    sb.append("<type>1</type>");
    sb.append("<param>");
    sb.append("<key><![CDATA[arg0]]></key>");
    sb.append("<value><![CDATA[nima]]></value>");
    sb.append("</param>");
    sb.append("</root>");
    String param = sb.toString();
    System.out.println(param);
    CloseableHttpClient httpClient = HttpClients.createDefault();
    HttpPost post = new HttpPost("http://localhost:13001/ESB/invokeAction/registerWithXML");
    String token = "test" + Constant.SPLIT_SIGN + "123456";
    String publicKeyStr = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAKHDpXYwv93+kl5DKoMIkn4dAVY6Qtp7ra8BlANXtavEFZW1+z+c4gQoiXQW89y0DCFpvP
    post.addHeader("Authorization", RSA.encryptByPublic(token, publicKeyStr));
    StringEntity entity = new StringEntity("param=" + param, "utf-8");
    entity.setContentEncoding("utf-8");
    entity.setContentType("application/x-www-form-urlencoded");
    post.setEntity(entity);
    HttpResponse httpResponse = httpClient.execute(post);
    httpResponse.setHeader("Content-Type", "text/plain;charset=utf-8");
    String s = EntityUtils.toString(httpResponse.getEntity(), "utf-8");
    System.out.println(s);
```

调用服务

json 格式调用

	invoke service with json			
Request	Direction	Terminal ----> WEB Server		
	transfer protocol	http/post		
	transmission format	Json		
	interface name	/invokeAction/invokeWithJson		
	param	description	necessary	param type
	siteCode	site code	Yes	string
	serviceCode	service code	Yes	string
	offline	0(default);1	Yes	string
	params	param key	No	string
	encrypt	No		
	test address			

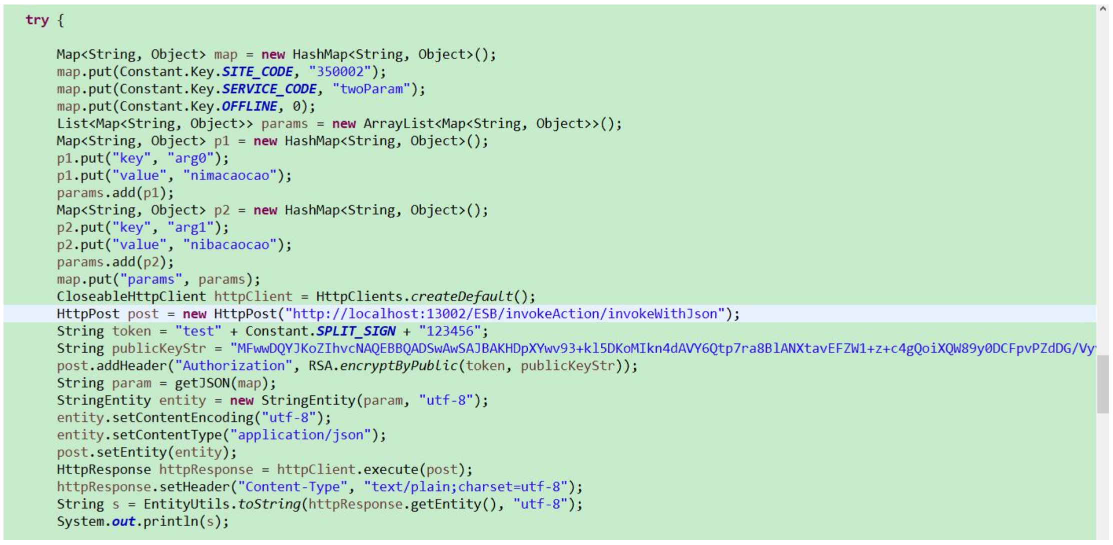
remark	param={
	"siteCode":"site code",
	"serviceCode":"service code",
	“offline”:1,
	"params":[{"value":"param 1",”key”:””},{"value":"param12",”key”:””}...{}]
	}
	put token:Authorization=token into head			

Response	Direction	WEB Server ----> Terminal		
	transfer protocol	http +JSON		
	encrypt	No		

	return description	{ “code” :100,routeId:””, “desc” :” success” ,data:[{“content of success”}]}
		{ “code” :101,routeId:”” “desc” :” 成功” ,data:[{“exception message”}]}		

	参数	说明		参数类型
	code			string
remark	①code : 100:success;101:failure;			

code path: test.com.camel.InvokeJsonTest.java



```
try {

    Map<String, Object> map = new HashMap<String, Object>();
    map.put(Constant.Key.SITE_CODE, "350002");
    map.put(Constant.Key.SERVICE_CODE, "twoParam");
    map.put(Constant.Key.OFFLINE, 0);
    List<Map<String, Object>> params = new ArrayList<Map<String, Object>>();
    Map<String, Object> p1 = new HashMap<String, Object>();
    p1.put("key", "arg0");
    p1.put("value", "nimacaocao");
    params.add(p1);
    Map<String, Object> p2 = new HashMap<String, Object>();
    p2.put("key", "arg1");
    p2.put("value", "nibacaocao");
    params.add(p2);
    map.put("params", params);
    CloseableHttpClient httpClient = HttpClients.createDefault();
    HttpPost post = new HttpPost("http://localhost:13002/ESB/invokeAction/invokeWithJson");
    String token = "test" + Constant.SPLIT_SIGN + "123456";
    String publicKeyStr = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAKHDpXYwv93+kl5DKoMIkn4dAVY6Qtp7ra8BlANXtavEFZW1+z+c4gQoiXQW89y0DCFpvPZdDG/Vy
    post.addHeader("Authorization", RSA.encryptByPublic(token, publicKeyStr));
    String param = getJSON(map);
    StringEntity entity = new StringEntity(param, "utf-8");
    entity.setContentEncoding("utf-8");
    entity.setContentType("application/json");
    post.setEntity(entity);
    HttpResponse httpResponse = httpClient.execute(post);
    httpResponse.setHeader("Content-Type", "text/plain;charset=utf-8");
    String s = EntityUtils.toString(httpResponse.getEntity(), "utf-8");
    System.out.println(s);
```

XML 格式调用

	invoke service with XML			
Request	Direction	Terminal ----> WEB Server		
	transfer protocol	http/post		
	transmission format	Json		
	interface name	/invokeAction/invokeWithXML		
	param	description	necessary	param type
	siteCode	site code	Yes	string
	serviceCode	service code	Yes	string
	offline	0(default);1	Yes	string
	params	param key	No	string
	encrypt	No		
	test address			
remark	param=<root> <siteCode>site code</siteCode> <serviceCode>service code</serviceCode> <offline></offline> <params>			

```
    <value><![CDATA[value 1]]></value>
    <key><![CDATA[key1]]></key>
    <params>
    <params>
    <value><![CDATA[value 2]]></value>
    <key><![CDATA[key 2]]></key>
    <params>
</root>
```

put token:Authorization=token into head

Response	Direction	WEB Server ----> Terminal	
	transfer protocol	http +JSON	
	encrypt	No	
	return description		

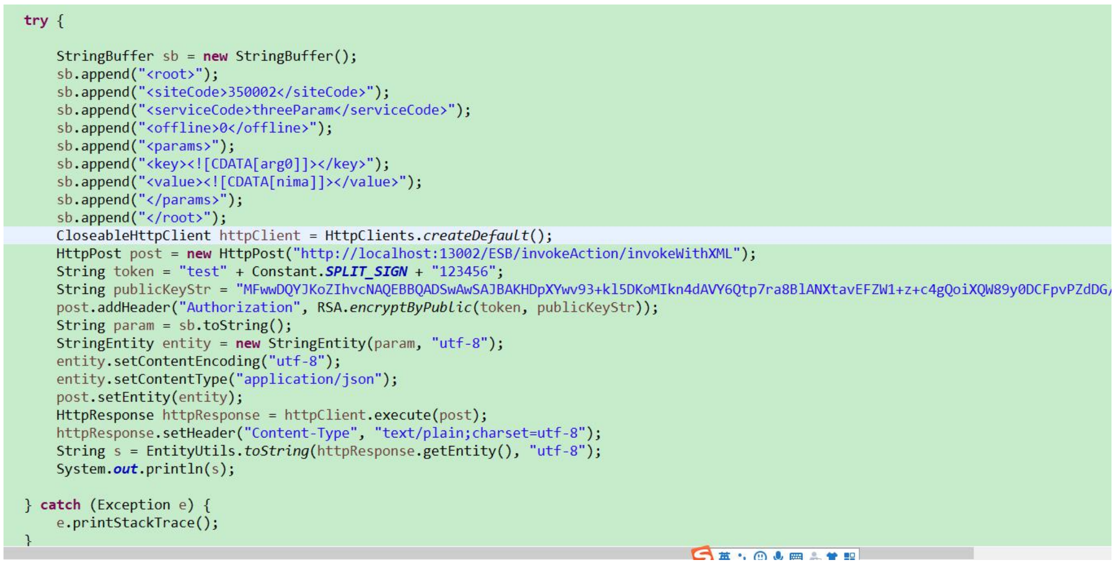
```
<root>
<code>100</code>
<desc>success</desc>
<routeId></routeId>
<data><![CDATA[content of success]]></data>
</root>

<root>
<data><![CDATA[exception message]]></data>
<routeId></routeId>
<code>101</code>
<desc>failure</desc>
</root>
```

参数	说明	参数类型
code		string

remark	①code : 100:success;101:failure;

code path: test.com.camel.InvokeJsonTest.java



```java
try {

    StringBuffer sb = new StringBuffer();
    sb.append("<root>");
    sb.append("<siteCode>350002</siteCode>");
    sb.append("<serviceCode>threeParam</serviceCode>");
    sb.append("<offline>0</offline>");
    sb.append("<params>");
    sb.append("<key><![CDATA[arg0]]></key>");
    sb.append("<value><![CDATA[nima]]></value>");
    sb.append("</params>");
    sb.append("</root>");
    CloseableHttpClient httpClient = HttpClients.createDefault();
    HttpPost post = new HttpPost("http://localhost:13002/ESB/invokeAction/invokeWithXML");
    String token = "test" + Constant.SPLIT_SIGN + "123456";
    String publicKeyStr = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAKHDpXYwv93+kl5DKoMIkn4dAVY6Qtp7ra8BlANXtavEFZW1+z+c4gQoiXQW89y0DCFpvPZdDG
    post.addHeader("Authorization", RSA.encryptByPublic(token, publicKeyStr));
    String param = sb.toString();
    StringEntity entity = new StringEntity(param, "utf-8");
    entity.setContentEncoding("utf-8");
    entity.setContentType("application/json");
    post.setEntity(entity);
    HttpResponse httpResponse = httpClient.execute(post);
    httpResponse.setHeader("Content-Type", "text/plain;charset=utf-8");
    String s = EntityUtils.toString(httpResponse.getEntity(), "utf-8");
    System.out.println(s);

} catch (Exception e) {
    e.printStackTrace();
}
```

json 格式文件上传(单个图片)

格式说明	调用			
Request	Direction	Terminal ----> WEB Server		
	传输协议	http/post		
	传输格式	Json		
	接口名	/invokeAction/invokeUploadWithJson		
	参数	说明	是否必须	参数类型
	siteCode	站点编码	必填	string
	serviceCode	服务号	必填	string
	offline	1 离线;0:即时(默认)	必填	int
	files	文件	必填	string
	params	参数 key	必填	string
	是否加密	否		
	测试地址			
备注	param={ "siteCode":"站点编码", "serviceCode":"服务号",			

	"offline":1, "files":[{"fileName":"文件名(包含后缀)","fileType:"文件类型""}], "params":[{"value":"参数 1","key":""},{"value":"参数 2","key":""}...{}] } 在 head 中增加 token:Authorization=token 若 type=4,文件的 key=file application/json		
Response	Direction	WEB Server ----> Terminal	
	传输协议	http +JSON	
	是否加密	否	

	返回说明	{"code":100,"success":true,"data":"成功的返回"}
		{"code":101,"success":false,"data":"失敗的異常"}	

	参数	说明	参数类型
	code	编码	string
	routeId	路由 id	string
	data	数据	string
备注	①code : 100:正确;101:错误;		

xml 格式文件上传(单个图片)

格式说明	调用			
Request	Direction	Terminal ----> WEB Server		
	传输协议	http/post,get		
	传输格式	Json		
	接口名	/invokeAction/invokeUploadWithXML		
	参数	说明	是否必须	参数类型
	siteCode	站点编码	必填	string
	serviceCode	服务号	必填	string
	offline	1 离线;0:即时(默认)	必填	int
	files	文件	必填	string
	param	参数 key	必填	string
	是否加密	否		
	测试地址			

备注	param=<root> <siteCode>站点编码</siteCode> <serviceCode>服务号</serviceCode> <offline></offline> <files> <fileName>文件名(包含后缀)</fileName> <fileType>文件类型</fileType> </files> <params> <value><![CDATA[参数 1]]></value> <key><![CDATA[参数 1]]></key> <params> <params> <value><![CDATA[参数 2]]></value> <key><![CDATA[参数 1]]></key> <params> <params> <value><![CDATA[参数 3]]></value> <key><![CDATA[参数 1]]></key> <params> </root> 在 head 中增加 token:Authorization=token 若 type=4,文件的 key=file application/json	
Response	Direction	WEB Server ----> Terminal
	传输协议	http +JSON
	是否加密	否
	返回说明	<root> <code>100</code> <desc>成功</desc> <routeId></routeId> <data><![CDATA[返回的数据]]></data> </root> <root>

		<data><![CDATA[异常]]></data> <routeId></routeId> <code>101</code> <desc>失败</desc> </root>	
	参数	说明	参数类型
	code	编码	string
	routeId	路由 id	string
	data	数据	string
备注	①code : 100:正确;101:错误;		

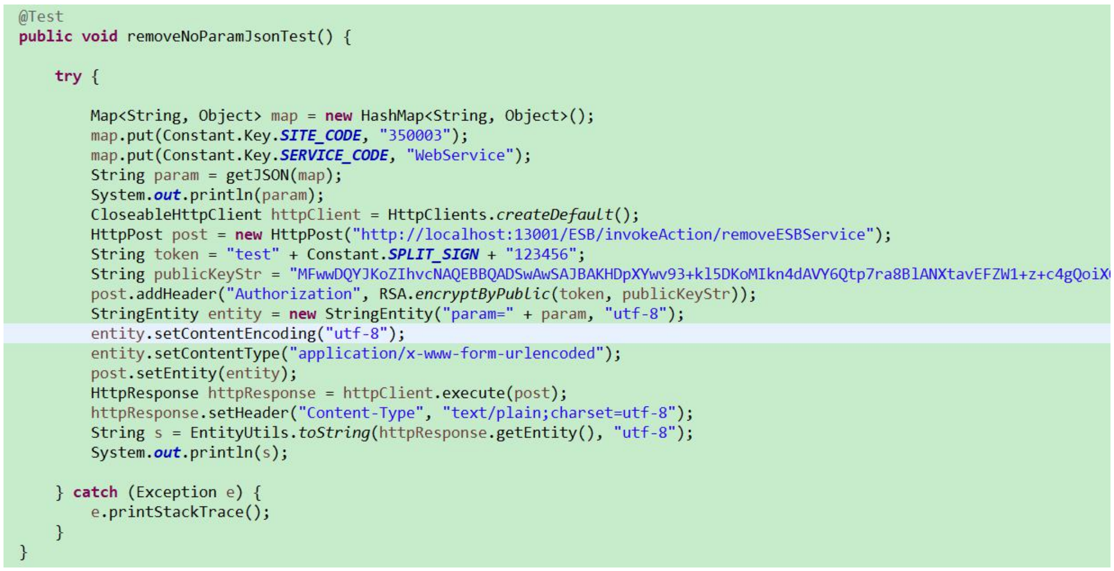
删除服务

	remove service with json			
Request	Direction	Terminal ----> WEB Server		
	transfer protocol	http/post		
	transmission format	Json		
	interface name	/invokeAction/removeESBService		
	param	description	necessary	param type
	siteCode	site code	Yes	string
	serviceCode	service code	Yes	string
	encrypt	No		
	test address			
remark	param={ “siteCode” :””, “serviceCode” :””]} put token:Authorization=token into head			
Response	Direction	WEB Server ----> Terminal		
	transfer protocol	http +JSON		
	encrypt	No		

	return description	{ “code” :100, “desc” :” success”]}
		{ “code” :101, “desc” :” failure” }]}		

remark	①code : 100:success;101:failure;

code path: test.com.camel.RmoveESBTest.java



```java
@Test
public void removeNoParamJsonTest() {

    try {

        Map<String, Object> map = new HashMap<String, Object>();
        map.put(Constant.Key.SITE_CODE, "350003");
        map.put(Constant.Key.SERVICE_CODE, "WebService");
        String param = getJSON(map);
        System.out.println(param);
        CloseableHttpClient httpClient = HttpClients.createDefault();
        HttpPost post = new HttpPost("http://localhost:13001/ESB/invokeAction/removeESBService");
        String token = "test" + Constant.SPLIT_SIGN + "123456";
        String publicKeyStr = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAKHDpXYwv93+kl5DKoMIkn4dAVY6Qtp7ra8BlANXtavEFZW1+z+c4gQoiX
        post.addHeader("Authorization", RSA.encryptByPublic(token, publicKeyStr));
        StringEntity entity = new StringEntity("param=" + param, "utf-8");
        entity.setContentEncoding("utf-8");
        entity.setContentType("application/x-www-form-urlencoded");
        post.setEntity(entity);
        HttpResponse httpResponse = httpClient.execute(post);
        httpResponse.setHeader("Content-Type", "text/plain;charset=utf-8");
        String s = EntityUtils.toString(httpResponse.getEntity(), "utf-8");
        System.out.println(s);

    } catch (Exception e) {
        e.printStackTrace();
    }
}
```