introduction

develop environment :

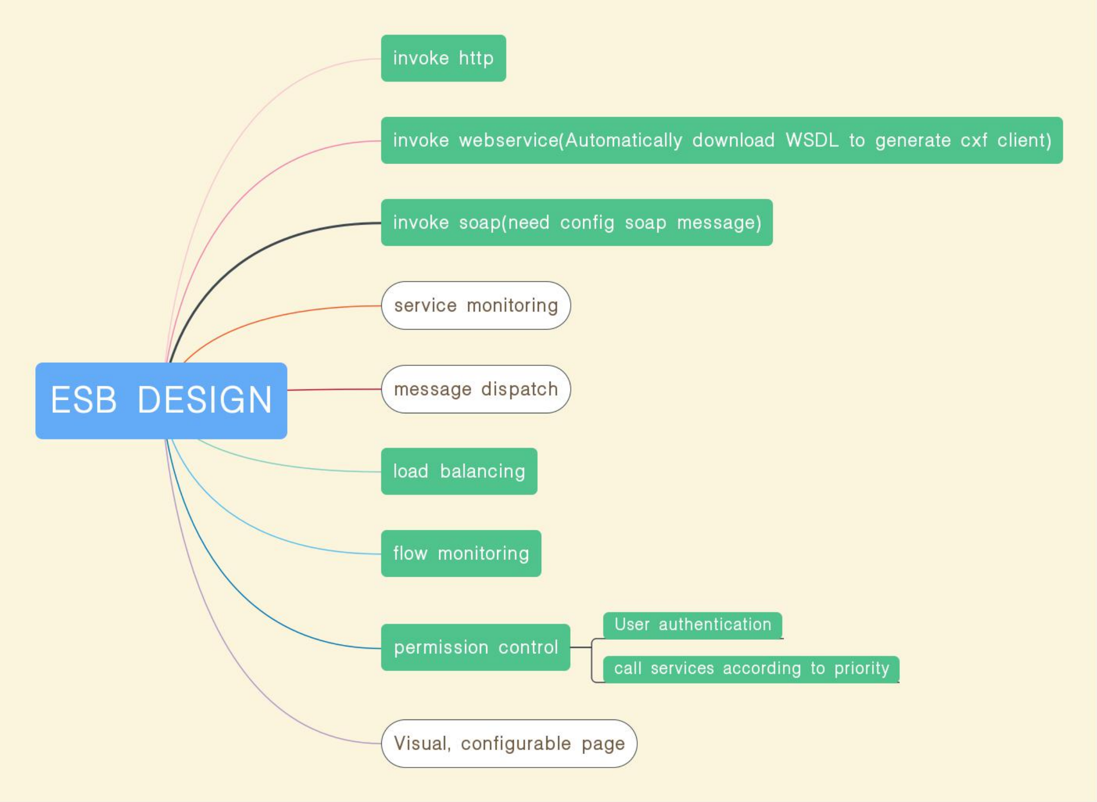
jdk8, zookeeper3.4.14 ,activemq 5.15.9,camel 3.0.0-M2

ESB framework:

base springcloud

functions provided:

green node have completed


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

Manual

user flow:

①get token by user register

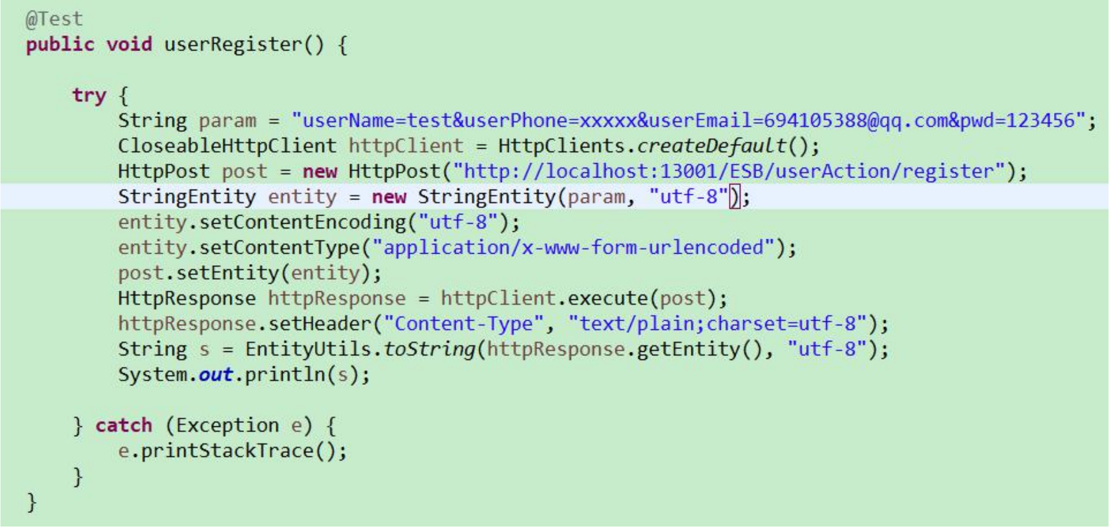
②register service

③invoke service

Get token by user register

	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



```
@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();
    }
}
```

register service

register service with json

	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
			Yes	int

	params	param key	No	string
	soap	it is necessary when type=3 ,Parameter replaced with “?”,	No	string
	encrypt	No		

	test address		

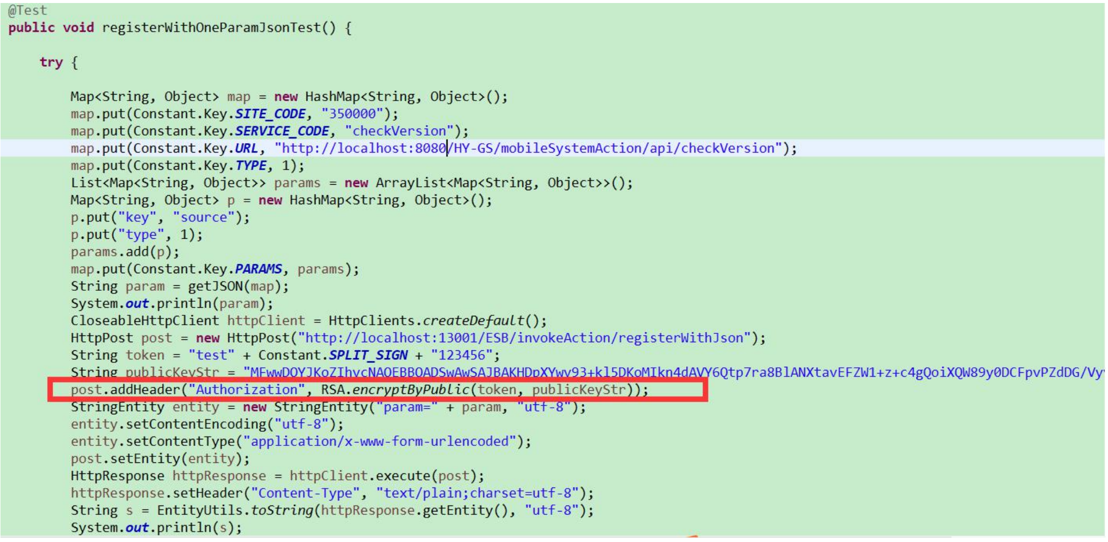
remark	param={
	"siteCode":"site code",
	"serviceCode":"service code",
	"url":resources",
	"type":1,//1:http;2:webservice;3:soap”
	params":[{"key":"","type":1},{}...{}]//1:text
	}
	put token:Authorization=token into head		

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



```
@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);
```

register service with xml

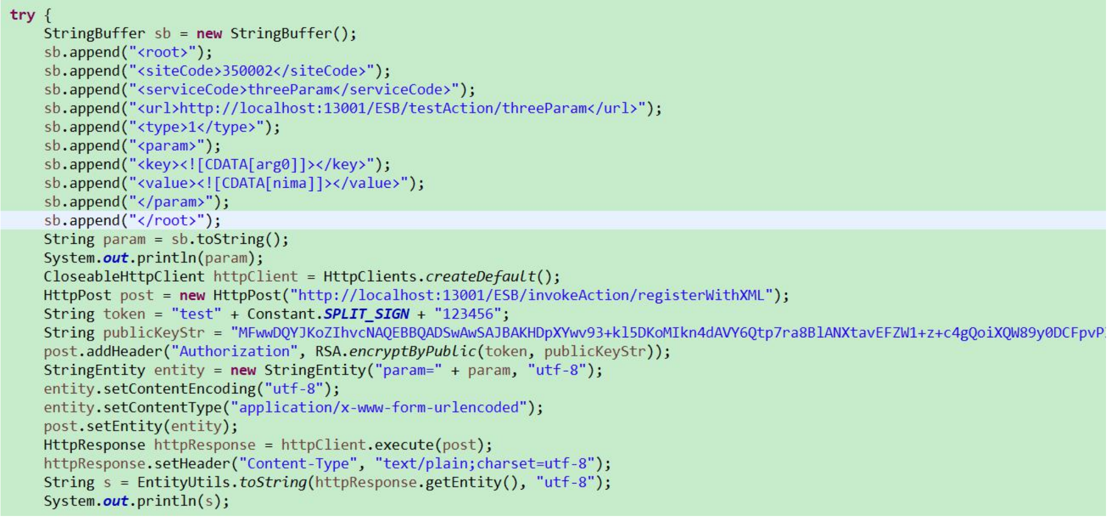
	register service with xml			
Request	Direction	Terminal ----> WEB Server		
	transfer protocol	http/post		
	transmission format	Json		
	interface name	/registerAction/registerWithXML		
	param	description	necessary	param type
	siteCode	site code	Yes	string
	serviceCode	service code	Yes	string
	url	resource;if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



```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);
```

invoke service

invoke service with json

	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			

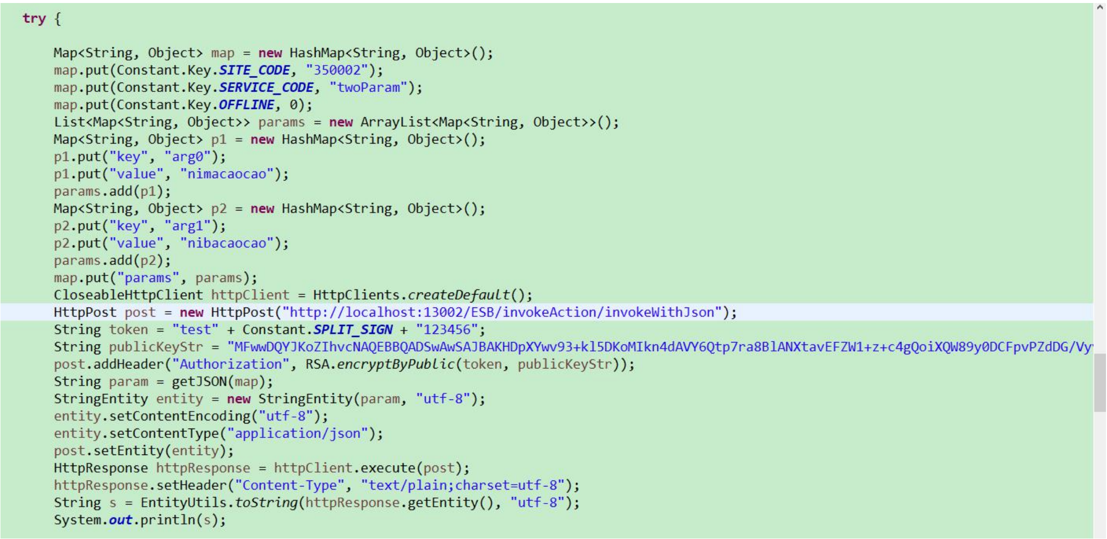
remark	param={
	"siteCode":"site code",
	"serviceCode":"service code",
	“offline”:1,
	"params":[{"value":"param 1",”key”:””},{"value":"param12",”key”:””}...{}]
	}
	put token:Authorization=token into head			

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



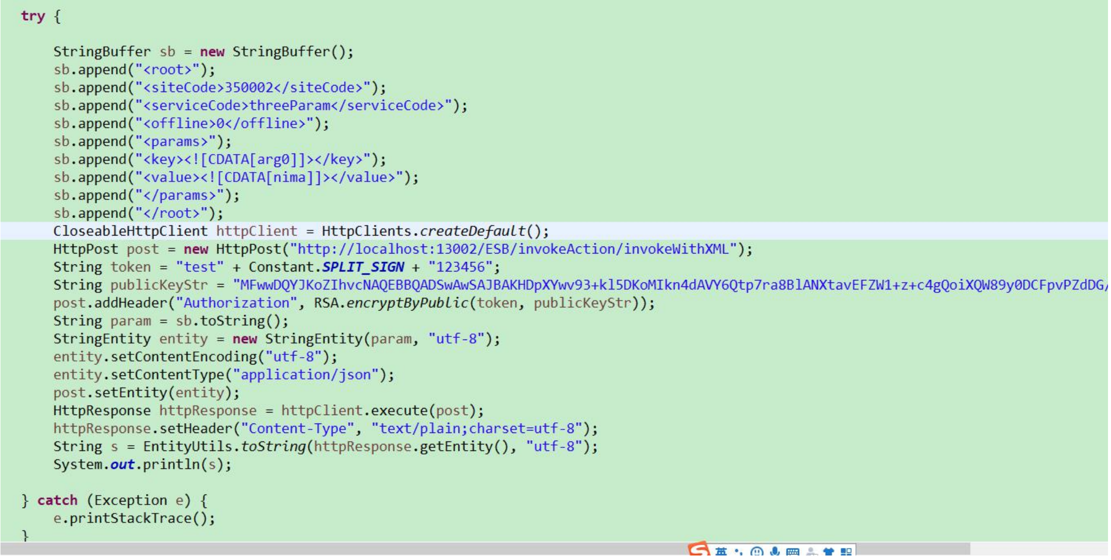
```java
  try {

      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);
```

invoke service with XML

	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



```
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();
}
```

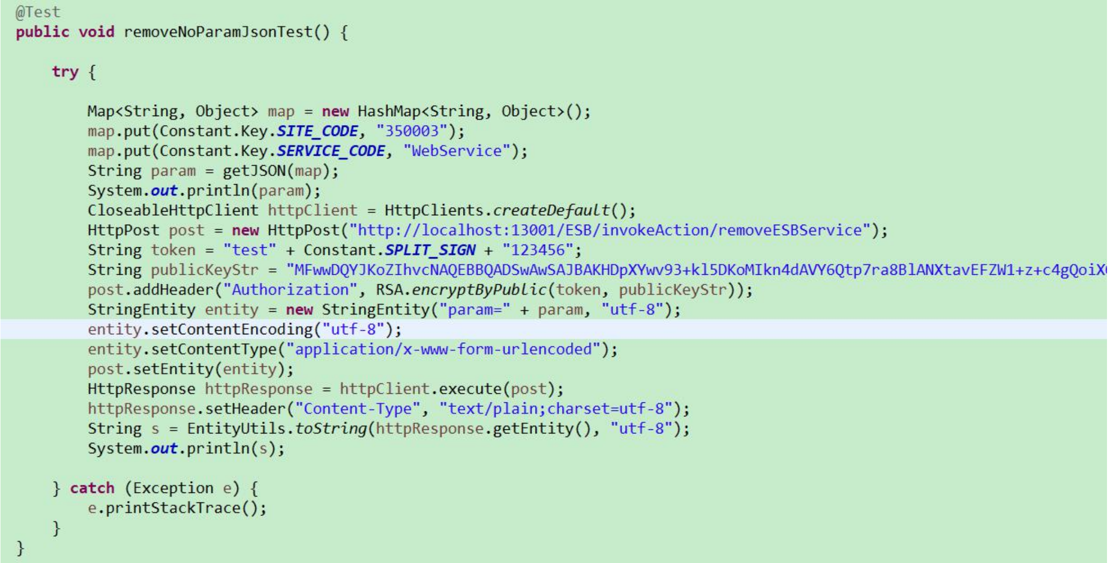
remove service

	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



```
@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();
    }
}
```