

环境介绍

开发环境:

jdk8
zookeeper3.4.14
activemq 5.15.9
camel 3.0.0-M2

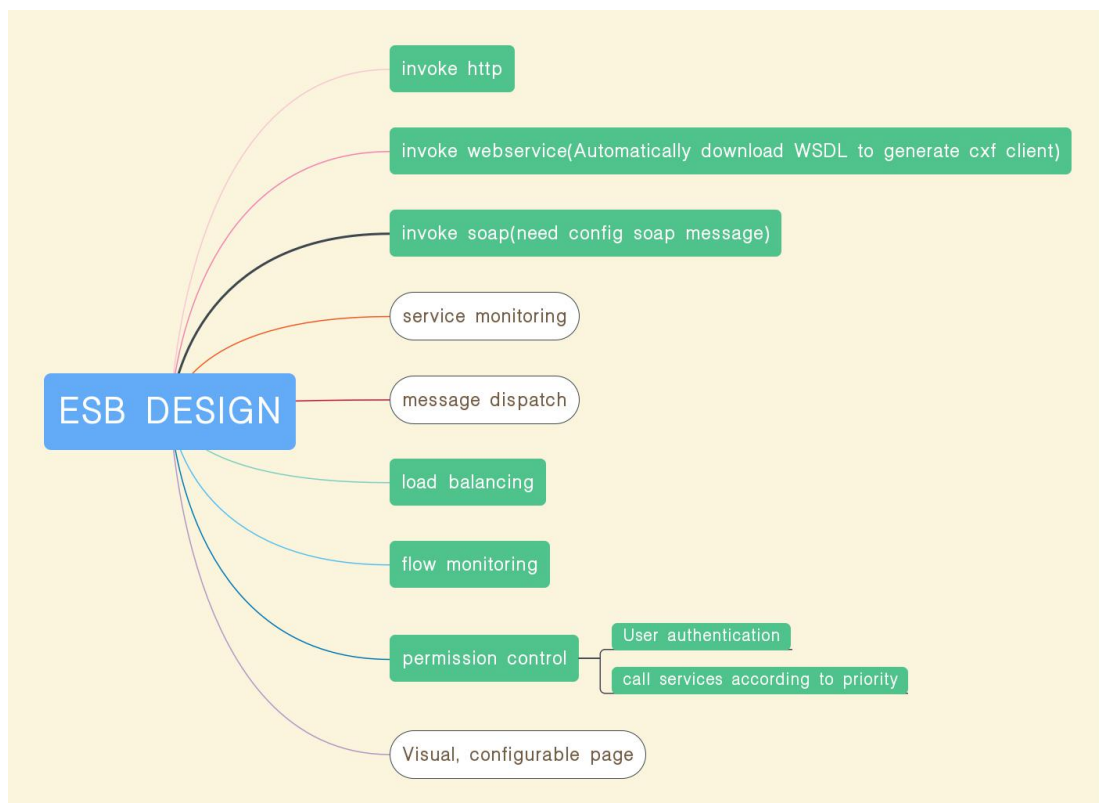
开发工具:

eclipse

框架:

基于 springcloud

功能脑图(绿色节点是已经完成的):



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

```
@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	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
	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+k15DKoMTkn4dAVY6Qtp7ra881ANxtavEFZW1+z+c4gQoiXQW89y0DCFpvpZdD6/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 ,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>			

	<div><key><![CDATA[param 2]]></key></div> <div><type>1:text;2:file</type></div> <div></param></div> <div></root></div> <div>put token:Authorization=token into head</div>		
Response	Direction	WEB Server ----> Terminal	
	transfer protocol	http +JSON	
	encrypt	No	
	return description	<div><root></div> <div><code>100</code></div> <div><desc>success</desc></div> <div></root></div> <div><root></div> <div><code>101</code></div> <div><desc>failure</desc></div> <div></root></div>	
	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+k15DKoMIkn4dAVY6Qtp7ra8B1ANXtavEFZW1+z+c4gQoiXQW89y0DCFvpP";
    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 = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAAKHDpXYwv93+k15DKoMIkn4dAVY6Qtp7ra8B1ANXtavEFZW1+z+c4gQoiXQw89y0DCFvpZdDG/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>			

	<pre><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</pre>		
Response	Direction	WEB Server ----> Terminal	
	transfer protocol	http +JSON	
	encrypt	No	
	return description	<pre><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></pre>	
	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 = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBABKHDpXYwv93+k15DKoMIkn4dAVY6Qtp7ra8B1ANXtavEFZW1+z+c4gQoiXQW89y0DCFvpPzd0G,
    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();
}
```

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

```
@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 = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAAKHDPXYwv93+k15DKoMIkn4dAVY6Qtp7ra8B1ANXtavEFZW1+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();
    }
}
```