

1 使用说明

能开平台接入只需要修改 原先的 请求url,并在消息头中加入 服务相关的信息即可，不影响原先的请求消息体。

2 接入服务

能开支持 接入 http/https 服务，对报文格式没有要求

2.1 http/https+json 接入

有如下服务接入

```
POST /test/post/json/json HTTP/1.1
Host: localhost:8010
Content-Type: application/json
Content-Length: 55
```

```
{
  "orderId": "333",
  "orderName": "json_json"
}
```

- 请求地址 <http://localhost:8080/test/post/json/json>
- 请求方式 post
- 报文格式 application/json
- 报文内容

```
{
  "orderId": "333",
  "orderName": "json_json"
}
```

2.1.1 新建服务

- 协议：指的是能开平台开放出去的接口协议 可选 http/https
- 报文格式： 请求报文的解析相关，此处选json
- 请求方式： 能开开放出去的接口请求方式 可选 get/post

* 服务名称

请输入服务名称

服务名称不能为空

* 服务编码

API_CODE_16497701407286233878

协议

RESTFUL

HTTP

HTTPS

CSF

Eureka

报文格式

JSON

XML

TEXT

FORM

* 请求方式

GET

POST

PUT

DELETE

超时时间 (毫秒)

60000

* 生效时间

🕒 2022-04-12 21:21:15

* 失效时间

🕒 2099-12-31 00:00:00

2.1.2 版本配置

- 服务类型：单服务
- 访问路径：选 <http://localhost:8080> 后面的内容 `/test/post/json/json`
- 请求方式：下游真实的请求方式 post

*** 版本号**

*** 版本名称**

*** 服务类型** 单服务 编排服务

*** 访问路径**

*** 访问方式** GET POST PUT DELETE

超时时间 (毫秒)

请求报文大小

版本描述

2.1.3 节点配置

- 协议：下游真实接口的协议 http
- 报文格式：下游真实接口的报文格式 json
- 节点集合：接入服务的节点，可以配置多个 能开负责负载均衡策略
- 客户端路径：用于适配定制化服务接入，不用填

添加节点分组
×

节点分组名称

* 节点分组编码

* 协议

RESTFUL

HTTP

HTTPS

CSF

Eureka

报文格式

JSON

XML

TEXT

FORM

* 节点集合

?

客户端路径

?

确定

取消

2.1.4 请求网关的路径

- http调用

请求路径 `http://能开地址:21100/http`

请求头

```
appCode 真实填写的代码
apiCode 真实填写的代码
apiVersion 真实填写的版本version
format json
```

- https 调用

请求路径 `https://能开地址:8443/https`

请求头

```
appCode 真实填写的代码
apiCode 真实填写的代码
apiVersion 真实填写的版本version
format json
```

2.2 http/https+formdata 接入

假设有如下 formdata 服务接入

```
POST /test/post/form/json HTTP/1.1
Host: localhost:8010
Content-Length: 226
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="orderId"

222
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="orderName"

post_form
----WebKitFormBoundary7MA4YWxkTrZu0gW
```

- 请求地址 <http://localhost:8080/test/post/form/json>
- 请求方式 post
- 报文格式 multipart/form-data
- 报文内容

```
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="orderId"

222
----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="orderName"

post_form
----WebKitFormBoundary7MA4YWxkTrZu0gW
```

2.2.1 新建服务

- 协议：指的是能开平台开放出去的接口协议 可选 http/https
- 报文格式： 请求报文的解析相关，此处选 form
- 请求方式： 能开开放出去的接口请求方式 可选 get/post

* 服务名称

请输入服务名称

* 服务编码

API_CODE_1649930550012179043

协议

RESTFUL

HTTP

HTTPS

CSF

Eureka

报文格式

JSON

XML

TEXT

FORM

* 请求方式

GET

POST

PUT

DELETE

超时时间 (毫秒)

60000

* 生效时间

🕒 2022-04-14 17:53:55

* 失效时间

🕒 2099-12-31 00:00:00

2.2.2 版本配置

- 服务类型：单服务
- 访问路径：选 <http://localhost:8080> 后面的内容 `/test/post/form/json`
- 请求方式：下游真实的请求方式 post

* 版本号	<input type="text"/>
* 版本名称	<input type="text"/>
* 服务类型	<input checked="" type="radio"/> 单服务 <input type="radio"/> 编排服务
* 访问路径	<input type="text" value="/test/post/form/json"/>
* 访问方式	<input type="radio"/> GET <input checked="" type="radio"/> POST <input type="radio"/> PUT <input type="radio"/> DELETE
超时时间 (毫秒)	<input type="text"/>
请求报文大小	<input type="text"/>
版本描述	<input type="text"/>

2.2.3 节点配置

- 协议：下游真实接口的协议 http
- 报文格式：下游真实接口的报文格式 form
- 节点集合：接入服务的节点，可以配置多个 能开负责负载均衡策略
- 客户端路径：用于适配定制化服务接入，不用填

节点分组名称

*** 节点分组编码**

*** 协议**

报文格式

*** 节点集合** ?

客户端路径 ?

2.2.4 请求网关的路径

- http调用

请求路径 `http://能开地址:21100/http`

请求头

`appCode` 真实填写的code

`apiCode` 真实填写的code

`apiVersion` 真实填写的版本version

`format` form

- https 调用

请求路径 `https://能开地址:8443/https`

请求头

`appCode` 真实填写的code

`apiCode` 真实填写的code

`apiVersion` 真实填写的版本version

`format` form

2.3 http/https+xml 接入

假设有如下 xml 服务接入

```
POST /test/post/xml/xml HTTP/1.1
Host: localhost:8010
Content-Type: application/xml
Content-Length: 109
```

```
<STUDENT STUDENT_ID="002">
  <STUDENT_NAME>李四</STUDENT_NAME>
  <STUDENT_SEX>女</STUDENT_SEX>
</STUDENT>
```

- 请求地址 <http://localhost:8080/test/post/xml/xml>
- 请求方式 post
- 报文格式 application/xml
- 报文内容

```
<STUDENT STUDENT_ID="002">
  <STUDENT_NAME>李四</STUDENT_NAME>
  <STUDENT_SEX>女</STUDENT_SEX>
</STUDENT>
```

2.3.1 新建服务

- 协议：指的是能开平台开放出去的接口协议 可选 http/https
- 报文格式： 请求报文的解析相关，此处选 xml
- 请求方式： 能开开放出去的接口请求方式 可选 get/post

* 服务名称

请输入服务名称

* 服务编码

API_CODE_16499317521378288289

协议

RESTFUL

HTTP

HTTPS

CSF

Eureka

报文格式

JSON

XML

TEXT

FORM

* 请求方式

GET

POST

PUT

DELETE

超时时间 (毫秒)

60000

* 生效时间

🕒 2022-04-14 17:53:55

* 失效时间

🕒 2099-12-31 00:00:00

效时间

04-14

8:14

04-14

6:11

2.3.2 版本配置

- 服务类型：单服务
- 访问路径：选 <http://localhost:8080> 后面的内容 `/test/post/xml/xml`
- 请求方式：下游真实的请求方式 post

* 版本号

* 版本名称

* 服务类型 单服务 编排服务

* 访问路径

* 访问方式 GET POST PUT DELETE

超时时间 (毫秒)

请求报文大小

版本描述

2.3.3 节点配置

- 协议：下游真实接口的协议 http
- 报文格式：下游真实接口的报文格式 xml
- 节点集合：接入服务的节点，可以配置多个 能开负责负载均衡策略
- 客户端路径：用于适配定制化服务接入，不用填

添加节点分组



节点分组名称

请输入节点分组名称

* 节点分组编码

请输入节点分组编码

* 协议

RESTFUL

HTTP

HTTPS

CSF

Eureka

报文格式

JSON

XML

TEXT

FORM

* 节点集合

http://localhost:8080;

客户端路径

确定

取消

2.4.4 请求网关的路径

- http调用

请求路径 `http://能开地址:21100/http`

请求头

`appCode` 真实填写的code

`apiCode` 真实填写的code

`apiVersion` 真实填写的版本version

`format` xml

- https 调用

请求路径 `https://能开地址:8443/https`

请求头

`appCode` 真实填写的code

`apiCode` 真实填写的code

`apiVersion` 真实填写的版本version

`format` xml

2.4 http/https+text 接入

适配除了 form表单 类型之外的所有的请求类型
可以注册 json xml soap 等多种报文格式的服务

假设有如下服务接入，请求报文是个 **json数组** 不能通过json接入, 可以通过 text接入

```
POST /test/post/jsonArray/json HTTP/1.1
Host: localhost:8010
Content-Type: application/json
Content-Length: 79
```

```
[
  {
    "orderId": "333",
    "orderName": "json_json"
  }
]
```

- 请求地址 <http://localhost:8080/test/post/jsonArray/json>
- 请求方式 post
- 报文格式 application/json
- 报文内容

```
[
  {
    "orderId": "333",
    "orderName": "json_json"
  }
]
```

2.4.1 新建服务

- 协议：指的是能开平台开放出去的接口协议 可选 http/https
- 报文格式： 请求报文的解析相关，此处选 text

- 请求方式：能开开放出去的接口请求方式 可选 get/post

× 添加服务开放 ×

* 服务名称

* 服务编码

协议 RESTFUL HTTP HTTPS CSF Eureka

报文格式 JSON XML TEXT FORM

* 请求方式 GET POST PUT DELETE

超时时间 (毫秒)

* 生效时间

* 失效时间

2.4.2 版本配置

- 服务类型：单服务
- 访问路径：选 <http://localhost:8080> 后面的内容 `/test/post/jsonArray/json`

- 请求方式：下游真实的请求方式 post

信息填写

参数配置

* 版本号	<input type="text"/>
* 版本名称	<input type="text"/>
* 服务类型	<input checked="" type="radio"/> 单服务 <input type="radio"/> 编排服务
* 访问路径	<input type="text" value="/test/post/jsonArray/json"/>
* 访问方式	<input type="radio"/> GET <input checked="" type="radio"/> POST <input type="radio"/> PUT <input type="radio"/> DELETE
超时时间 (毫秒)	<input type="text"/>
请求报文大小	<input type="text"/>
版本描述	<input type="text"/>

2.4.3 节点配置

- 协议：下游真实接口的协议 http
- 报文格式：下游真实接口的报文格式 text
- 节点集合：接入服务的节点，可以配置多个 能开负责负载均衡策略
- 客户端路径：用于适配定制化服务接入，不用填

添加节点分组

节点分组名称

* 节点分组编码

* 协议

报文格式

* 节点集合 ?

客户端路径 ?

2.4.4 请求网关的路径

- http调用

请求路径 `http://能开地址:21100/http`

请求头

`appCode` 真实填写的code
`apiCode` 真实填写的code
`apiVersion` 真实填写的版本version
`format` text

- https 调用

请求路径 `https://能开地址:8443/https`

请求头

`appCode` 真实填写的code
`apiCode` 真实填写的code
`apiVersion` 真实填写的版本version
`format` text

3 请求示例

3.1 请求头描述

请求头	是否必填	类型	备注
appCode	是	String	应用编码
apiCode	是	String	服务编码
apiVersion	是	String	服务版本号
format	是	String	请求报文格式 可填 json text form xml
sign	否	String	消息签名，开启签名校验时必填
accessToken	否	String	接入认证信息，开启接入认证开关时必填

3.2 场景举例

1. 下游接入服务是 http/https + json ，通过 http 的方式请求能开平台

请求报文

```
curl --location --request POST 'http://192.168.0.1:21100/http' \  
--header 'appCode: testApp' \  
--header 'apiCode: testApi' \  
--header 'apiVersion: 1.0.0' \  
--header 'format: json' \  
--data-raw '{  
  "startTime":"2021-09-01 12:21:22",  
  "endTime":"2021-09-30 12:21:22"  
}'
```

2. 下游接入服务是 http/https + form ，通过 https 的方式请求能开平台

此处的form 指的是 multipart/form-data;

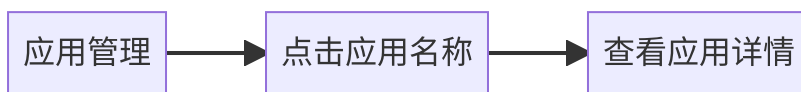
请求报文

```
curl --location --request POST 'http://192.168.0.1:8443/https' \  
--header 'appCode: testApp' \  
--header 'apiCode: testApi' \  
--header 'apiVersion: 1.0.0' \  
--header 'format: form' \  
--form 'userId="1"' \  
--form 'name="2"'
```

4 安全网关接入

4.1 获取授权信息

按照如下顺序找到应用详情信息，获取应用编码和应用密钥



测试应用2
app_16448903091161820972

应用名称 测试应用2	应用编码 app_16448903091161820972	状态 上线
应用密钥 c9037ab0aa888255dae835930708d1af		
加密类型 仅网关加密	加密算法 AES	签名算法 SHA256
授权回调地址		
RSA公钥 MGcwDQYJKoZIhvcNAQEBBQADVgAwUwJMAJYRtsaHlKBPKC5+AZstlyT1ta7P5OHUVCdbJUJ6MN5oqjWEv2WfuAMiTkV9q/MvWHIGrb6KBUgb03y9w+/g/vh1v0eCRd4LtvXpJQIDAQAB		
应用描述 测试应用2-说明		
授权类型 Oauth2Token模式	CRM工号	

4.2 准备aifgw-security服务

1. 按照网管安装手册，安装 aifgw-security 服务
2. 开启网关的token校验

- 请求地址
网关backend的31100端口

```
POST /api/filterStatus/update HTTP/1.1
Host: localhost:31100
Content-Type: application/json
```

```
[
  {
    "filterCode": "com.asiainfo.aifgw.controlchain.gw.filters.inbound.AppAuthFilter",
    "filterType": "global_filter",
    "isOpen": "N"
  }
]
```

4.3 调用演示

1. 获取accessToken

- 请求报文

```
POST /oauth2/oauth/token HTTP/1.1
Host: localhost:9001
Content-Type: application/x-www-form-urlencoded
Content-Length: 111
```

```
grant_type=client_credentials&client_id=app_16291992646709486308&client_secret=f6d901be9e31258315de65758d5d1f10
```

参数名称	key	是否必填	类型	备注
授权模式	grant_type	是	String	客户端授权模式，传client_credentials
App的应用编码	client_id	是	String	App的应用编码
App应用密钥	client_secret	是	String	App应用密钥

- 响应报文

```
{
  "access_token": "8ff468c2-1e2a-427b-9c57-28b36fcdf573",
  "token_type": "client_credentials",
  "expires_in": 7199,
  "scope": "all"
}
```

参数名称	key	是否必填	类型	备注
授权Token	access_token	是	String	访问令牌
Token类型	token_type	是	String	
有效期	expires_in	是	String	
授权范围	scope	是	String	

2. 请求网关时带上 accessToken

POST

http://{{ip}}:21100/http

Params ● Authorization Headers (14) Body ● Pre-request Script Tests Settings

<input checked="" type="checkbox"/>	apiCode	API_CODE_16291985439783020000	
<input checked="" type="checkbox"/>	apiVersion	1.0.0	
<input checked="" type="checkbox"/>	format	http+json	
<input checked="" type="checkbox"/>	accessToken	ed977f6c-9587-4800-a3b1-67f034f31c57	
<input checked="" type="checkbox"/>	userId	31b6bdb8df1e01eb9dc97f28888d739c	

Key

Value

Description