kong网关+konga UI管理+JWT验证

笔记本:	kong		
创建时间:	2018/9/2 9:30	**更新时间:**	2018/9/2 10:14
作者:	754236623@qq.com		
URL:	http://52.83.182.209:7321/swagger/index.html		

1、部署kong和konga

- kong+konga: https://konghq.com/install/， https://www.cnblogs.com/zhoujie/p/kong5.html， https://github.com/pantsel/konga
- jwt plugin: https://docs.konghq.com/plugins/jwt/

2、访问konga

http://192.168.200.57:1337/#!/login， admin/adminadminadmin
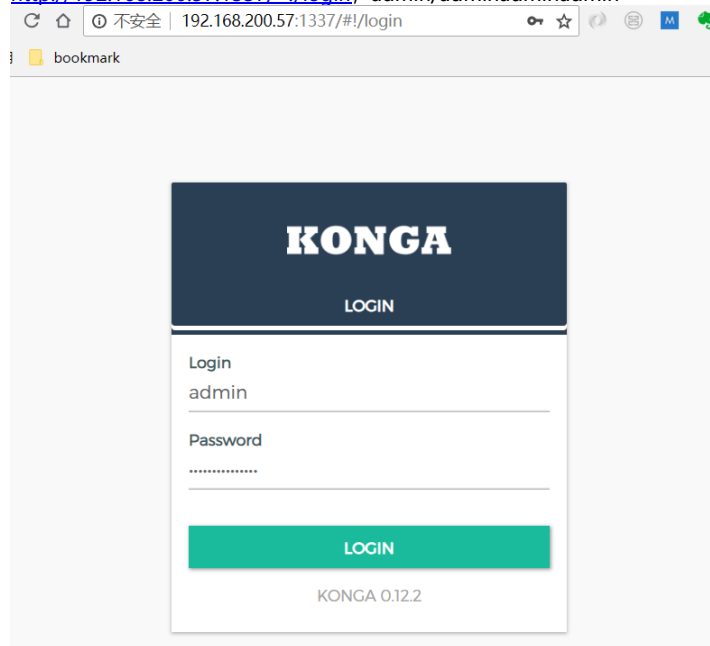
首页
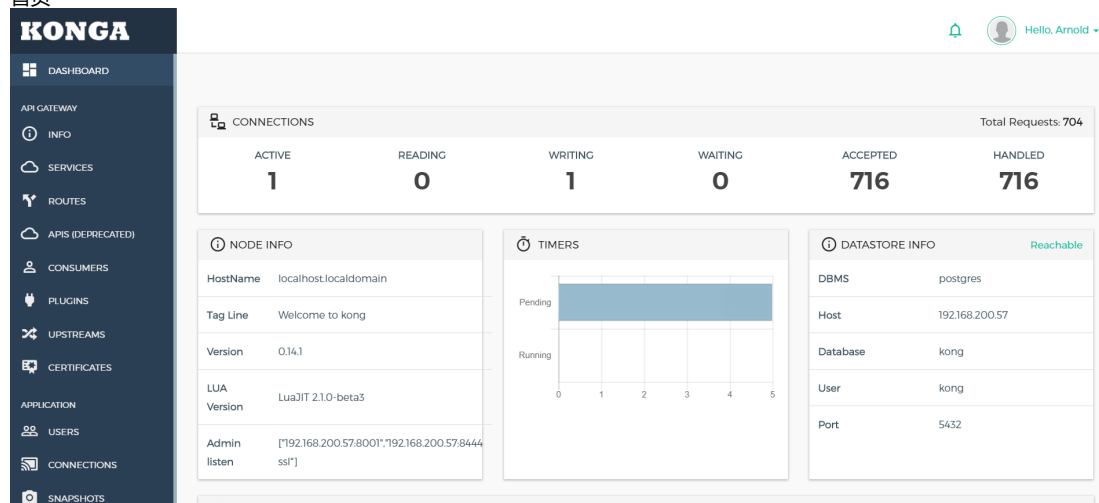
3、创建一个service，只要填写名称和Url即可，点击确认后，其它信息会自动补全

例如: http://52.83.182.209:7321/dme/api/algorithms/metadata/v1

Name
(optional)

名称

The service name.

Description
(optional)

An optional service description.

Tags
(optional)

Optionaly add tags to the service

Url
(shorthand-attribute)

(真实的endpoint完整地址)

Shorthand attribute to set `protocol`, `host`, `port` and `path` at once. This attribute is write-only (the Admin API never "returns" the url).

Protocol
(semi-optional)

The protocol used to communicate with the upstream. It can be one of `http` or `https`.

Host
(semi-optional)

The host of the upstream server.

Port
(semi-optional)

The upstream server port. Defaults to `80`.

Path
(optional)

The path to be used in requests to the upstream server. Empty by default.

Retries
(optional)

5

The number of retries to execute upon failure to proxy. The default is

自动补全

Service dme-service

services / show

- ⓘ Service Details
- 🖇 Routes
- 🔌 Plugins
- 👥 Eligible consumers `beta`

ⓘ Service details

Name
(optional)

dme-service

The service name.

Description
(optional)

An optional service description.

Tags
(optional)

Optionaly add tags to the service

Protocol
(semi-optional)

http

The protocol used to communicate with the upstream. It can be one of `http` or `https`.

Host
(semi-optional)

52.83.182.209

The host of the upstream server.

Port
(semi-optional)

7321

The upstream server port. Defaults to `80`.

Path
(optional)

/dme/api/algorithms/metadata/v1

The path to be used in requests to the upstream server. Empty by default.

Retries
(optional)

5

The number of retries to execute upon failure to proxy. The default is `5`.

Services

Service entities, as the name implies, are abstractions of each of your own upstream services. Examples of Services would be a data transformation microservice, a billing API, etc.

➕ ADD NEW SERVICE

🔍 search...

Results : 25 ▾

NAME	HOST	TAGS	CREATED ⌃	
👁 dme-service	52.83.182.209		Sep 1, 2018	🗑 DELETE

4、设置路由
在services模块中添加路由

Service dme-service

services / show

- Service Details
- **Routes**
- Plugins
- Eligible consumers `beta`

Routes　　　　　　　　　　　　　　　+ ADD ROUTE

search routes...

Hosts	Paths	Protocols	Methods	Regex priority	Created		
👁 -	/myroute	http, https	-	0	Sep 1, 2018	✏ EDIT	🗑 DELETE

- **Route Details**
- Plugins
- Eligible consumers `beta`

ⓘ Route details

* For hosts, paths, methods and protocols, press enter to apply every value you type

Hosts
(semi-optional)

A list of domain names that match this Route. For example: example.com. At least one of hosts, paths, or methods must be set.

Paths
(semi-optional)

/myroute ✕

A list of paths that match this Route. For example: /my-path. At least one of `hosts`, `paths`, or `methods` must be set.

Methods
(semi-optional)

A list of HTTP methods that match this Route. At least one of `hosts`, `paths`, or `methods` must be set.

Strip Path
(optional)

YES

When matching a Route via one of the `paths`, strip the matching prefix from the upstream request URL.

Preserve Host
(optional)

NO

When matching a Route via one of the `hosts` domain names, use the request `Host` header in the upstream request headers. By default set to `false`, and the upstream Host header will be that of the Service's `host`

Protocols
(semi-optional)

http ✕　https ✕

A list of the protocols this Route should allow. By default it is `["http", "https"]`, which means that the Route accepts both. When set to `["https"]`, HTTP requests are answered with a request to upgrade to HTTPS.

✓ SUBMIT CHANGES

5、设置jwt plugin

Route 31b81a6d-790a-4dda-acbd-2ae580074503

routes / route

- Route Details
- **Plugins**
- Eligible consumers `beta`

Assigned plugins　　　　　　　　　　+ ADD PLUGIN

search plugins...

		Name	Consumer	Created	
✴	●● 👁	jwt	All consumers	Sep 2, 2018	🗑 DELETE

EDIT JWT ✕

Verify requests containing HS256 or RS256 signed JSON Web Tokens (as specified in RFC 7519). Each of your Consumers will have JWT credentials (public and secret keys) which must be used to sign their JWTs. A token can then be passed through the Authorization header or in the request's URI and Kong will either proxy the request to your upstream services if the token's signature is verified, or discard the request if not. Kong can also perform verifications on some of the registered claims of RFC 7519 (exp and nbf).   **ENABLED**

secret is base64   **NO**

If true, the plugin assumes the credential's secret to be base64 encoded. You will need to create a base64 encoded secret for your consumer, and sign your JWT with the original secret.

cookie names

Tip: Press Enter to accept a value.

key claim name   iss   ⬅

The name of the claim in which the key identifying the secret must be passed.

anonymous

claims to verify

Tip: Press Enter to accept a value.

A list of registered claims (according to RFC 7519) that Kong can verify as well. Accepted values: exp, nbf.
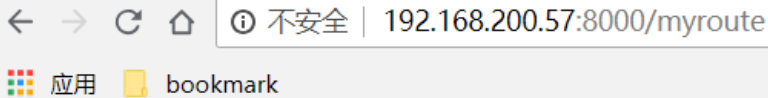
maximum expiration   0

run on preflight   **YES**

uri param names   access_token ✕   ⬅

PS：key claim name：iss，默认为：iss；uri param names：是设置requestparam的标签，例如：http://xxxx?access_token=xxxxx

6、如果不设置token，访问服务，则提示缺失未授权

← → C ⌂ ① 不安全 | 192.168.200.57:8000/myroute

⦂⦂ 应用   📁 bookmark

```
{
    message: "Unauthorized"
}
```

7、创建consumer 凭证

Consumers

The Consumer object represents a consumer - or a user - of an API. You can either rely on Kong as the primary datastore, or you can map the consumer list with your database to keep consistency between Kong and your existing primary datastore.

+ CREATE CONSUMER   🔍 search...   Results： 25

USERNAME	CUSTOM_ID	CREATED ∧	
👁 weifj	weifj	Sep 01 2018 @22:39	🗑 DELETE

CREATE JWT ✕

Create JWT for **weifj**

key
(optional)

A unique string identifying the credential. If left out, it will be auto-generated. However, usage of this key is **mandatory** while crafting your token, as specified in the next section.

algorithm
(optional)

HS256

The algorithm used to verify the token's signature. Can be `HS256` or `RS256`.

rsa_public_key
(optional)

If `algorithm` is `RS256`, the public key (in PEM format) to use to verify the token's signature.

secret
(optional)

If `algorithm` is `HS256`, the secret used to sign JWTs for this credential. If left out, will be auto-generated. If `algorithm` is `RS256`, this is the private key (in PEM format) to use to verify the token's signature.

✓ SUBMIT

只选择算法，其它会自动创建。

CONSUMER: weifj

consumers / edit consumer

| ⓘ Details | 👥 Groups | 🛡 Credentials | ☁ Accessible APIs | ☁ Accessible Services | ⑂ Accessible Routes | 🔌 Plugins |

BASIC
API KEYS
HMAC
OAUTH2
JWT

🔏 **JWT**　　+ CREATE JWT

```
{
    "created_at": 1535812800000,
    "id": "e2c140e6-27d3-4cc4-96f9-3b733c256d5e",
    "algorithm": "HS256",
    "key": "DxhxDnOOmOT3SsRhY1pPFXAg6v2nFaX8",
    "secret": "t9CiFkCGgMGfh2YL15sIvzoNV8h9MgUx",
    "consumer_id": "9dce23f6-b930-42ff-a611-11c5719d761c"
}
```

Created
Sep 1, 2018

🗑 DELETE

里面的key作为下面jwt plugin的key claim name：iss值

8、根据consumer的信息生成jwt
打开网址：https://jwt.io/，里面可以生成jwt信息。

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
pc3MiOiJEeGh4RG5PT21PVDNTc1JoWTFwUEZYQWc
2djJuRmFYOCJ9.3aYnusReOZVZZcfYJVybYDK-
8Je7FI9mFmrUDrM72gk
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
    "alg": "HS256",
    "typ": "JWT"
}
```
← 算法

PAYLOAD: DATA

```
{
    "iss": "DxhxDnOOmOT3SsRhY1pPFXAg6v2nFaX8"
}
```
← consumer生成的key

VERIFY SIGNATURE

```
HMACSHA256(
    base64UrlEncode(header) + "." +
    base64UrlEncode(payload),
    t9CiFkCGgMGfh2YL15sIvz
```
← consumer生成的secret
```
)
```
☐ secret base64 encoded

9、访问网关服务

1）第一种方式，在uri后面添加参数：？
access_token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEeGh4RG5PT21PVDNTc1JoWTFwUEZYQWc2djJuRmFYOCJ9.3aYnusReOZVZZcfYJVybYDK-8Je7FI9mFmrUDrM72gk

http://192.168.200.57:8000/myroute?
access_token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEeGh4RG5PT21PVDNTc1JoWTFwUEZYQWc2djJuRmFYOCJ9.3aYnusReOZVZZcfYJVybYDK-8Je7FI9mFmrUDrM72gk

效果图：
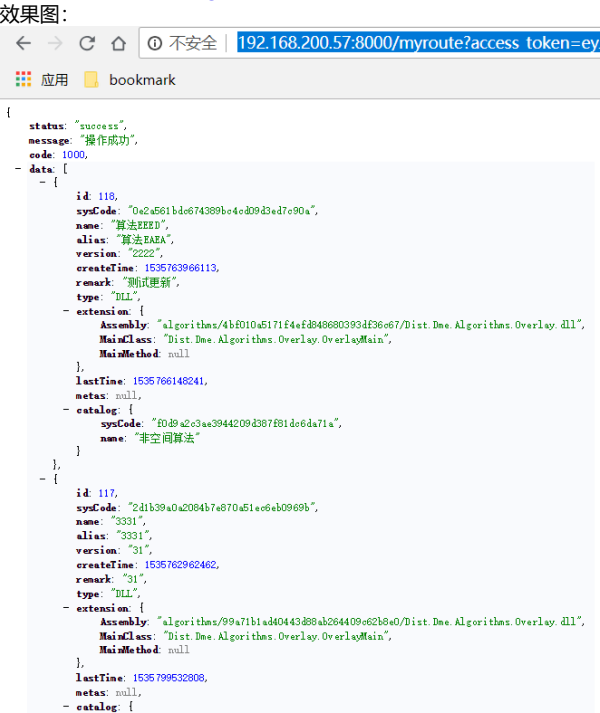
2）第二种方式，在请求头添加 authorization
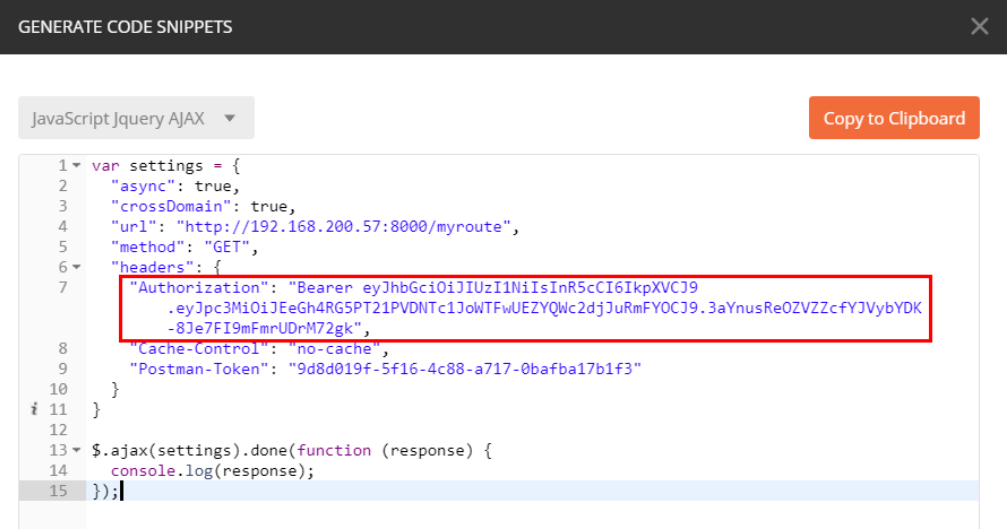
效果图：

GET ▾ | http://192.168.200.57:8000/myroute | Params | Send ▾ | Save ▾

Authorization ● | Headers (1) | Body | Pre-request Script | Tests | Cookies Coc

TYPE

Bearer Token ▾

The authorization header will be automatically generated when you send the request. Learn more about authorization

Preview Request

⚠ Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. ✕
Learn more about variables

Token | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJEeGh4RG5PT21PVDNTc1JoWTFwUEZYQWc2djJuRmFY...

Body | Cookies | Headers (8) | Test Results   Status: 200 OK   Time: 108 ms   Size: 4.92 KB

Pretty | Raw | Preview | JSON ▾

```
11        "version": "2222",
12        "createTime": 1535763966113,
13        "remark": "测试更新",
14        "type": "DLL",
15        "extension": {
16            "Assembly": "algorithms/4bf010a5171f4efd848680393df36c67/Dist.Dme.Algorithms.Overlay.dll",
17            "MainClass": "Dist.Dme.Algorithms.Overlay.OverlayMain",
18            "MainMethod": null
19        },
20        "lastTime": 1535766148241,
21        "metas": null,
22        "catalog": {
23            "sysCode": "f0d9a2c3ae3944209d387f81dc6da71a",
24            "name": "非空间算法"
25        }
26    },
27    {
```