

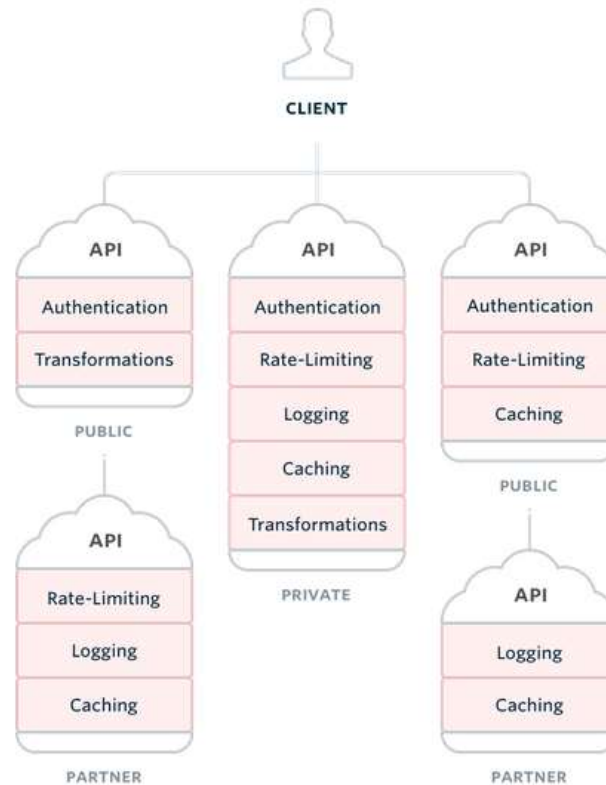


KONG

Kong is a scalable, open source **API Layer** (*also known as a API Gateway, or API Middleware*).

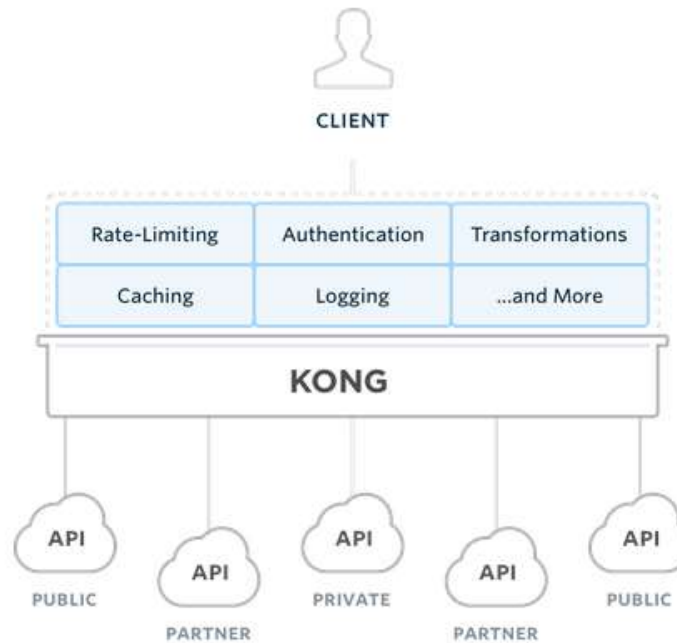
Kong runs in front of any RESTful API and is extended through Plugins, which provide extra functionalities and services beyond the core platform.

Current Architecture

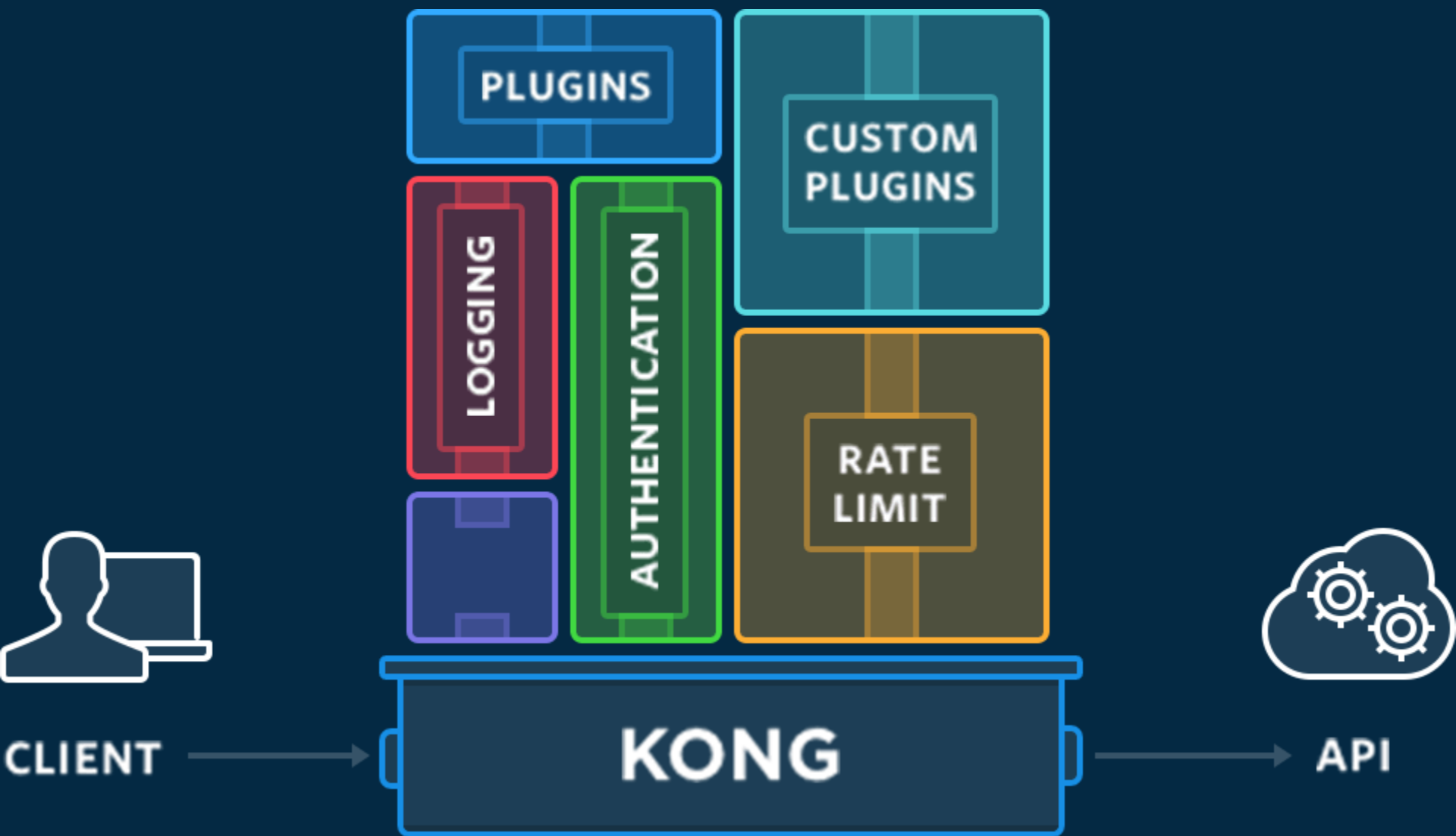


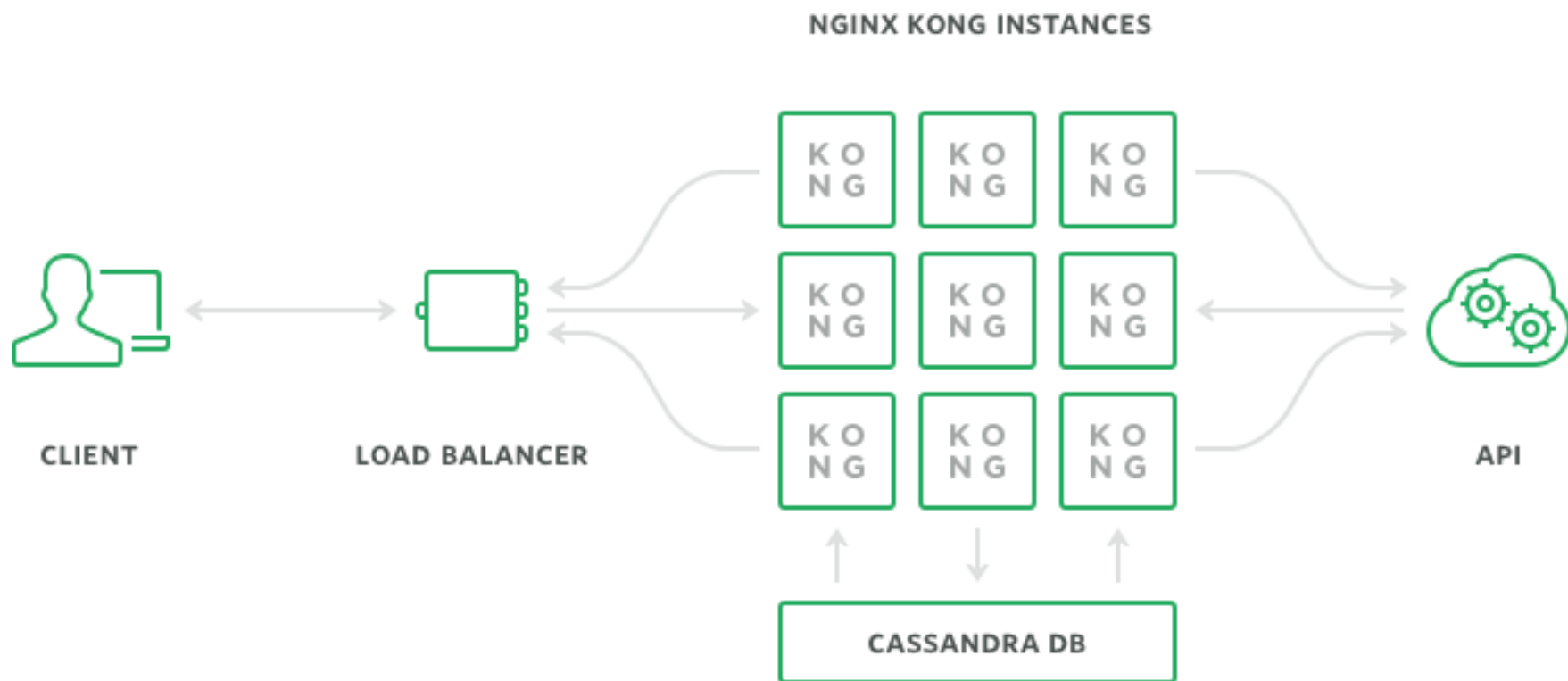
- ✗ Common functionality is duplicated across multiple services
- ✗ Systems tend to be monolithic and hard to maintain
- ✗ Difficult to expand without impacting other services
- ✗ Productivity is inefficient because of system constraints

Kong Architecture



- ✓ Kong centralizes and unifies functionality into one place
- ✓ Build efficient distributed architectures ready to scale
- ✓ Expand functionality from one place with a simple command
- ✓ Your team is focused on the product, Kong does the REST





Authentication



Basic Authentication

Add Basic Authentication to your APIs



Key Authentication

Add a key authentication to your APIs



OAuth 2.0 Authentication

Add an OAuth 2.0 authentication to your APIs



HMAC Authentication

Add HMAC Authentication to your APIs



JWT

Verify and authenticate JSON Web Tokens

Security



ACL

Control which consumers can access APIs



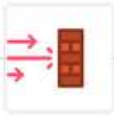
CORS

Allow developers to make requests from the browser



SSL

Add a SSL certificate for an underlying service



IP Restriction

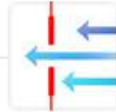
Whitelist or blacklist IPs that can make requests

Traffic Control



Rate Limiting

Rate-limit how many HTTP requests a developer can make



Response Rate Limiting

Rate-Limiting based on a custom response header value



Request Size Limiting

Block requests with bodies greater than a specific size

Transformation



Request Transformer

Modify the request before hitting the upstream server



Response Transformer

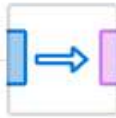
Modify the upstream response before returning it to the client

Logging



TCP

Send request and response logs to a TCP server



UDP

Send request and response logs to an UDP server



HTTP

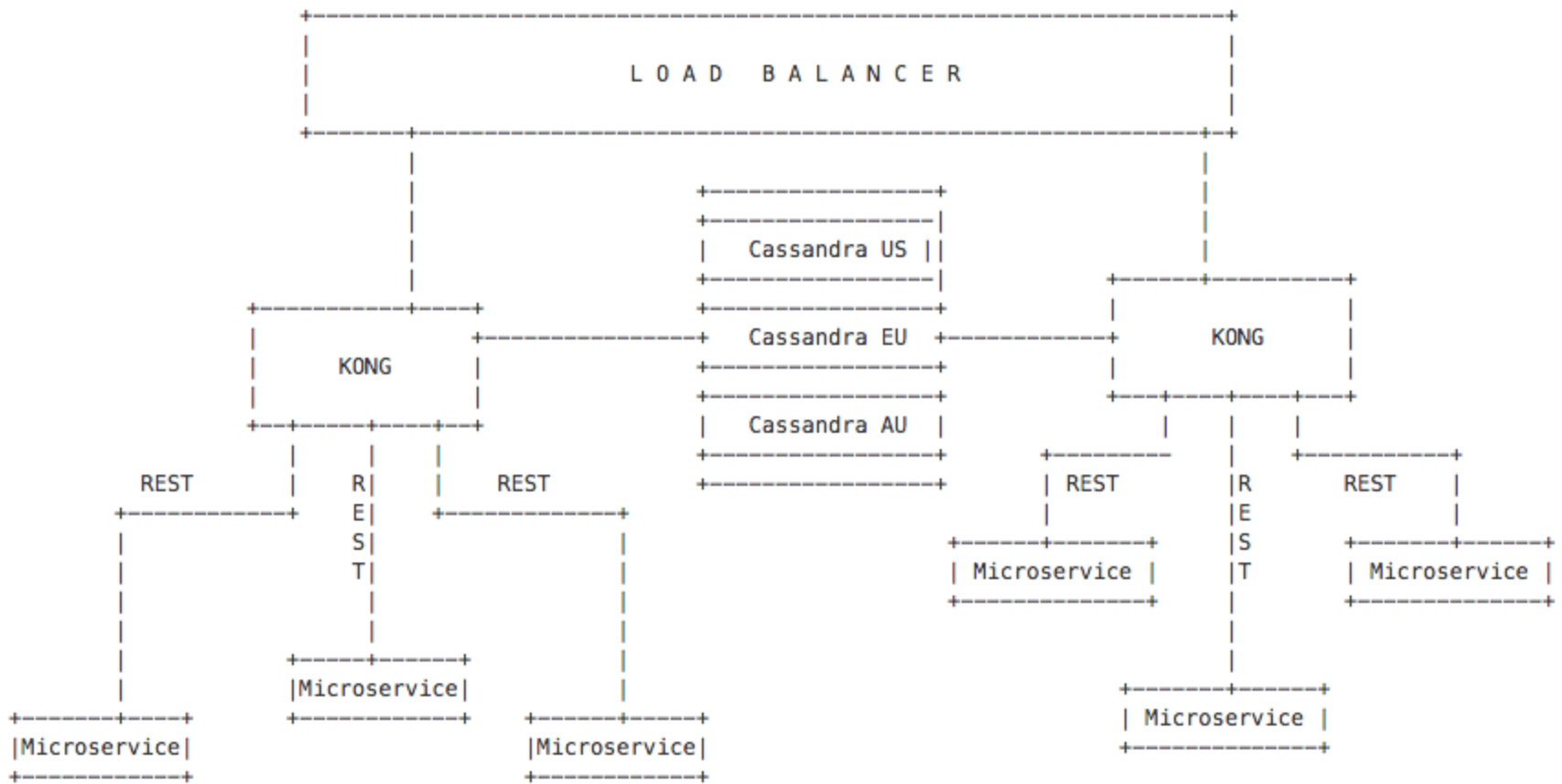
Send request and response logs to an HTTP server



File

Append request and response data to a log file on disk

Implement KONG



Example: Create API

```
$ curl -i -X POST
--url http://192.168.59.103:8001/apis/
--data 'name=mockbin'
--data 'target_url=http://mockbin.com/'
--data 'public_dns=mockbin.com'
```

HTTP/1.1 201 Created

Date: Mon, 22 Jun 2015 08:05:07 GMT

Content-Type: application/json; charset=utf-8

Transfer-Encoding: chunked

Connection: keep-alive

Server: kong/0.3.2

```
{ "public_dns": "mockbin.com", "target_url": "http://mockbin.c
om/", "id": "5656b41f-c3c7-4ea7-ce30-  
b187c779d9e5", "created_at": 1434960307000, "name": "mockbin" }
$
```

Example: Get API

```
$ curl -i -X GET \  
  --url http://192.168.59.103:8000/ \  
  --header 'Host: mockbin.com'  
HTTP/1.1 200 OK  
Date: Mon, 22 Jun 2015 08:05:33 GMT  
Content-Type: text/html; charset=utf-8  
Transfer-Encoding: chunked  
Connection: keep-alive  
Set-Cookie: __cfduid=d73f83d5ee63c47bea8274b9debe016421434960333;  
expires=Tue, 21-Jun-16 08:05:33 GMT; path=/; domain=.mockbin.com;  
HttpOnly  
Etag: W/"WjyUnylhiU0eFTCRGSBgnQ=="  
Vary: Accept-Encoding  
Via: kong/0.3.2  
Server: cloudflare-nginx  
CF-RAY: 1fa672e2192f045b-NRT  
:  
<!DOCTYPE html><html><head>...  
$
```

Example: Enable Plugins

```
$ curl -i -X POST \  
  --url http://192.168.59.103:8001/apis/mockbin/plugins/ \  
  --data 'name=keyauth'
```

```
HTTP/1.1 201 Created
```

```
Date: Tue, 23 Jun 2015 09:20:02 GMT
```

```
Content-Type: application/json; charset=utf-8
```

```
Transfer-Encoding: chunked
```

```
Connection: keep-alive
```

```
Server: kong/0.3.2
```

```
{ "api_id": "966e541a-d3c0-4a26-ca20-ce63be9da3a3", "id": "db497a57-  
3538-478a-c791-  
9321c7a36cc7", "value": { "key_names": [ "apikey" ], "hide_credentials":  
false }, "enabled": true, "created_at": 1435051202000, "name": "keyauth"  
}
```

```
$
```

Example: Get API

```
$ curl -i -X GET \  
  --url http://192.168.59.103:8000/ \  
  --header 'Host: mockbin.com'  
HTTP/1.1 403 Forbidden  
Date: Tue, 23 Jun 2015 09:20:37 GMT  
Content-Type: application/json; charset=utf-8  
Transfer-Encoding: chunked  
Connection: keep-alive  
Server: kong/0.3.2
```

```
{"message": "Invalid authentication  
credentials"}
```

```
$
```


Example: Create Key Auth

```
$ curl -i -X POST \  
  --url  
http://192.168.59.103:8001/consumers/teerapat/keyauth/ \  
  --data 'key=1234567890qwerty'  
HTTP/1.1 201 Created  
Date: Tue, 23 Jun 2015 09:24:31 GMT  
Content-Type: application/json; charset=utf-8  
Transfer-Encoding: chunked  
Connection: keep-alive  
Server: kong/0.3.2  
  
{ "created_at": 1435051471000, "consumer_id": "6e9eb3a9-f0cc-  
4295-c2d4-  
59f37078fb58", "key": "1234567890qwerty", "id": "83f4473c-  
3178-46bf-c6c5-2645999409ff" }  
  
$
```

Example: Create Consumer

```
$ curl -i -X POST \
  --url http://192.168.59.103:8001/consumers/ \
  --data 'username=teerapat'
HTTP/1.1 201 Created
Date: Tue, 23 Jun 2015 09:22:54 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Server: kong/0.3.2

{"username":"teerapat","created_at":1435051374000,"id":"6e9eb3a9-f0cc-4295-c2d4-59f37078fb58"}
$
```

Example: Get API

```
$ curl -i -X GET \  
  --url http://192.168.59.103:8000 \  
  --header 'Host: mockbin.com' \  
  --header 'apikey:  
1234567890qwerty'  
HTTP/1.1 200 OK  
Date: Tue, 23 Jun 2015 09:25:30 GMT  
Content-Type: text/html;  
charset=utf-8  
  
OK  
  
$
```

Example: Get API (wrong key)

```
$ curl -i -X GET \  
  --url http://192.168.59.103:8000 \  
  --header 'Host: mockbin.com' \  
  --header 'apikey: 1234567890qwert'  
HTTP/1.1 403 Forbidden  
Date: Tue, 23 Jun 2015 14:02:08 GMT  
Content-Type: application/json; charset=utf-8  
Transfer-Encoding: chunked  
Connection: keep-alive  
Server: kong/0.3.2  
  
{ "message": "Invalid authentication  
credentials" }  
$
```



KONG