关于如何在openresty中修改响应体的数据。

了解：什么是openresty？什么是反向代理？如何修改响应体数据？

OpenResty

OpenResty® 是一个基于 Nginx 与 Lua 的高性能 Web 平台，其内部集成了大量精良的 Lua 库、第三方模块以及大多数的依赖项。用于方便地搭建能够处理超高并发、扩展性极高的动态 Web 应用、Web 服务和动态网关。

有些人使用 Lua 完成一些非常简单的任务，比如检查和修改某些请求头和响应体数据，而有些人则利用 Lua 创建非常复杂的 Web 应用、 CDN 软件和 API 网关等等。其中， [ngx\_lua](https://github.com/chaoslawful/lua-nginx-module) 在 nginx.conf 里面使用 lua 来在服务器端对数据做一些处理再丢给浏览器。

OpenResty 反向代理

反向代理（Reverse Proxy）是现今网络中一种非常重要的技术，它位于客户端和真正的服务器（即所谓的后端）之间，接受客户端的请求并转发给后端，然后把后端的处理结果返回给客户端。从客户端的角度来看，访问反向代理和真正的后端服务器两者并没有任何区别。（<https://wiki.shileizcc.com/confluence/pages/viewpage.action?pageId=47415810>）

Location / {

proxy\_pass houduan.ip;   # /的请求转发到 houduan.ip

}

修改响应体数据

<https://github.com/openresty/lua-nginx-module#body_filter_by_lua>

服务端单次返回的数据为 [ngx.arg](https://github.com/openresty/lua-nginx-module#ngxarg)[1] [ngx.arg](https://github.com/openresty/lua-nginx-module#ngxarg)[2]。其中[ngx.arg](https://github.com/openresty/lua-nginx-module#ngxarg)[1]表示响应体数据，他是一个lua的字符串类型。 [ngx.arg](https://github.com/openresty/lua-nginx-module#ngxarg)[2]为标记位eof，他是lua的boolean类型。如果返回数据太大，服务端分多次返回，当最后一次返回时，eof为true，表示数据传输结束。

Lua可以改变响应体数据[ngx.arg](https://github.com/openresty/lua-nginx-module#ngxarg)[1]的值重新给下游，从而实现修改响应体数据。

其中修改响应体数据中主要有三种方法：

body\_filter\_by\_lua <lua-script-str>:

body\_filter\_by\_lua\_block { lua-script-str }:

body\_filter\_by\_lua\_file <path-to-lua-script-file>:

例子：修改diviner.jd.local/diviner的返回体数据为hello world

1 在nginx.conf中配置一下代码即可实现：

location /diviner{

proxy\_pass <http://diviner.jd.local>;

header\_filter\_by\_lua\_block { ngx.header.content\_length = nil }

body\_filter\_by\_lua 'ngx.arg[1] = “hello world ” ';

}

2 通过lua脚本修改：

nginx.conf：

location /diviner{

proxy\_pass <http://diviner.jd.local>;

header\_filter\_by\_lua\_block { ngx.header.content\_length = nil }

body\_filter\_by\_lua\_file modify.lua;

}

modify.lua:

ngx.arg[1] = “hello world ”

拓展：如果响应体是json，实现指定key的修改时，需要将数据json.decode转换为json。

常见错误：

1 nginx配置更新失败，

sudo nginx -c /usr/local/openresty/nginx/conf/nginx.conf start/stop/restart

nginx不是默认路径时，每次重载需要指定路径更新。

2 curl带参数的url请求失败：

带有&的url在curl时要加入引号，curl “<http://localhost/p=1&q=2>”

4 响应数据转换为json失败

json模块导入失败：lua文件中配置package.path package.cpath才能正确引入json库。

json解码失败，解码成table后要编码成string再返回才正常显示。

返回失败，返回数据太大时，分块返回，在lua文件中整合后再修改返回。

参考：

简介：https://wdicc.com/intro-openresty/

<https://moonbingbing.gitbooks.io/openresty-best-practices>

<https://doc.openresty.com.cn/cn/>

<https://openresty.org/cn/>

<https://github.com/openresty/lua-nginx-module>

<http://nginx.org/en/docs/beginners_guide.html>

<https://wiki.shileizcc.com/confluence/display/openresty/OpenResty>

{"encode":"utf-8","success":true,"error\_msg":"","latency":69,

"impr":"//knicks.jd.com/log/server?t=rec\_common\_exp&v=type=rec.902008$src=rec$action=0$reqsig=ebc186e5cfc86a4ff19a88e74ed0a64a0e31703b$enb=1

$csku=56665720704,64447020186,68611897408,100006572499,40694959920,2144425,10021794605463,68374107690,71833625022,35063693428,71183223016,69905546690,65897486460,71278387941,52316481872,100013879670,68613936778,70465327725,71180038253,100015097506,100007265825,68622598619,59818206650,70022284787

$st=0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

$sku=8414512$p=902008$pin=$uuid=869234031078350-20f77c8b7865$expid=0$mexpid=$gm=FeatLogid%5Eebc186e5cfc86a4ff19a88e74ed0a64a0e31703b$rt=0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0$rid=4059803630092118233$ver=1$sig=82905f6f1ad98e0c7f9b4af237e09e20a2a9bc35",