



استاد : دکتر امید عباس زاده

دانشجو : احمد اسدی و کیارش رحمانی

شماره دانشجویی : ۹۹۴۶۳۱۰۷ و ۹۹۴۶۳۱۳۹

پروژه Kong

**گام ۱: Docker**

لینک دانلود: <https://desktop.docker.com/win/main/amd64/Docker%20Desktop%20Installer.exe?utm_source=docker&utm_medium=webreferral&utm_campaign=dd-smartbutton&utm_location=module>  
سایت داکر : <https://www.docker.com/products/docker-desktop/>

**گام ۲ : نصب KONG**  
**Pull the Docker image**

docker pull kong/kong-gateway:2.7.0.0-alpinet

**Create a Docker network**

docker network create kong-net

**prepare Postgres DB**

docker run -d --name kong-database --network=kong-net -e “POSTGRES\_USER=kong” -e “POSTGRES\_DB=kong” -e “POSTGRES\_PASSWORD=kong” -p 5432:5432 postgres:9.6

**start Postgres**

docker run --rm --network=kong-net -e “KONG\_DATABASE=postgres” -e “KONG\_PG\_HOST=kong-database” -e “KONG\_PG\_PASSWORD=kong” kong:latest kong migrations bootstrap

**Start kong**

docker run -d --name kong --network=kong-net -e “KONG\_DATABASE=postgres” -e “KONG\_PG\_HOST=kong-database” -e “KONG\_PG\_PASSWORD=kong” -e “KONG\_PROXY\_ACCESS\_LOG=/dev/stdout” -e “KONG\_ADMIN\_ACCESS\_LOG=/dev/stdout” -e “KONG\_PROXY\_ERROR\_LOG=/dev/stderr” -e “KONG\_ADMIN\_ERROR\_LOG=/dev/stderr” -e “KONG\_ADMIN\_LISTEN=0.0.0.0:8001, 0.0.0.0:8444 ssl” -p 8000:8000 -p 8443:8443 -p 8001:8001 -p 8444:8444 -p 8002:8002 kong:latest

**KONG UI**

[**http://127.0.0.1:8002**](http://127.0.0.1:8002)

helper website : <https://medium.com/@dijin123/kong-api-gateway-docker-container-9d4bada843a7>

**گام ۳ : درست کردن Service**

**نصب flask برای ایجاد Service**

pip install flask

**کد Service**

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

    print("Received a request!")

    return 'This is rate limiting Plugin from KONG !'

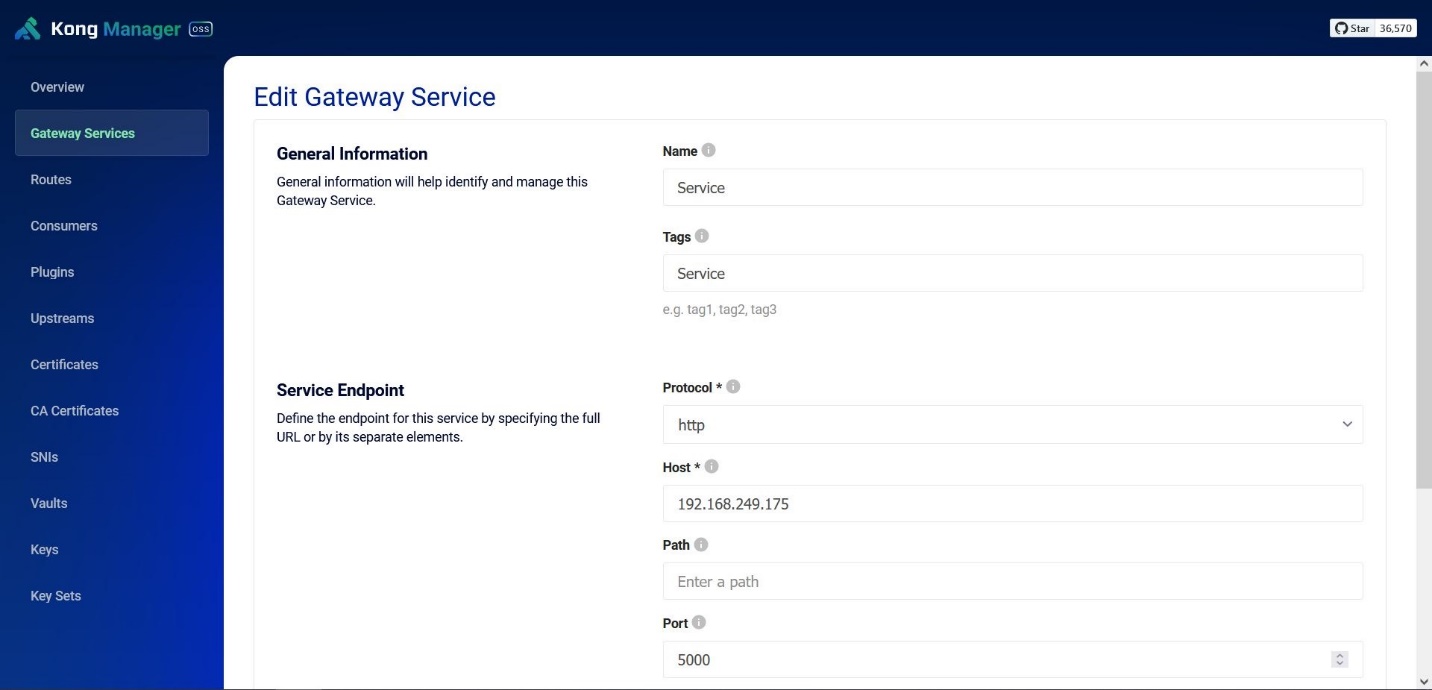
if \_\_name\_\_ == '\_\_main\_\_':

    app.run(host='0.0.0.0', port=5000)

**گام ۴ : درست کردن Plugin های KONG**

**تنظیمات اولیه :**

**درست کردن Gateway Service**

****

**Host : IP شما**

ip config

    Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :

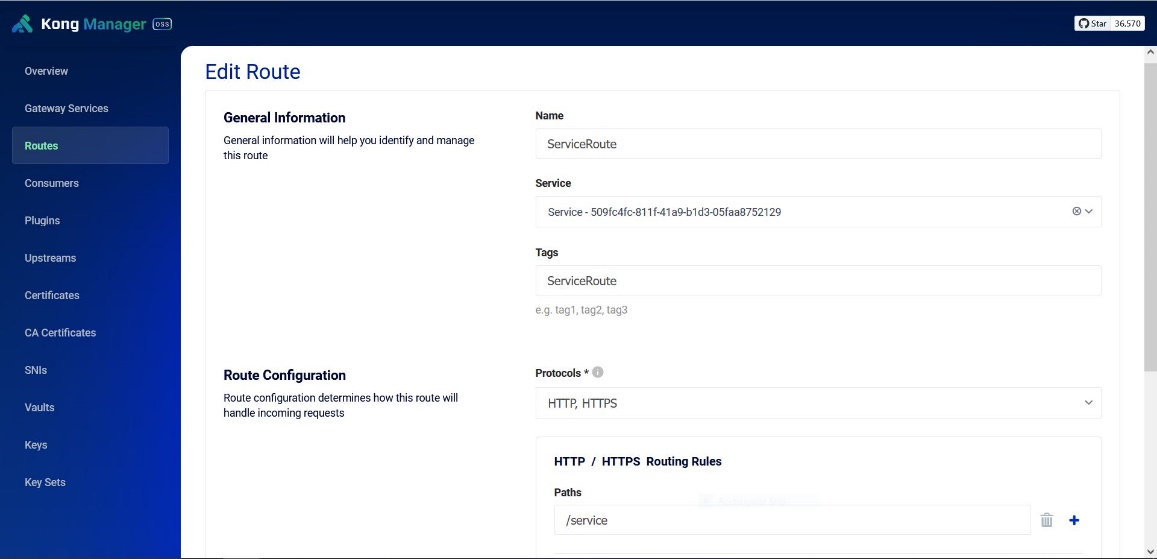
   Link-local IPv6 Address . . . . . : fe80::b6f2:ce43:bf1b:7ade%16

   IPv4 Address. . . . . . . . . . . : 192.168.249.175

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 192.168.249.166

**درست کردن Route**

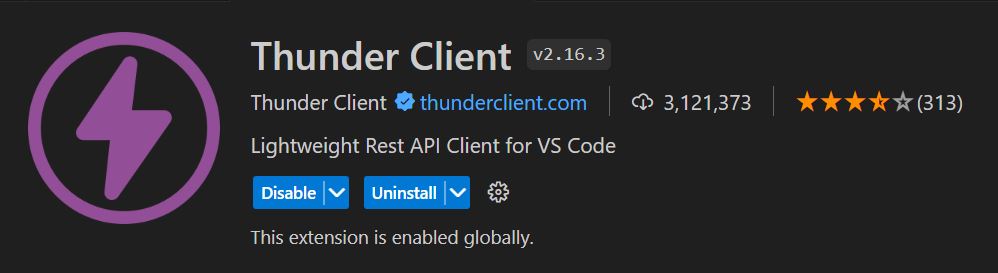
****

**Name : یک اسم دلخواه برای Route**

**Service : اسم Service ی که درست کرده بودیم.**

**نصب یک API Platform :**

**باید یک api platform نصب کنید که ما به شما postman یا extention در vscode به نام Thunder Client را پیشنهاد می دهیم.**

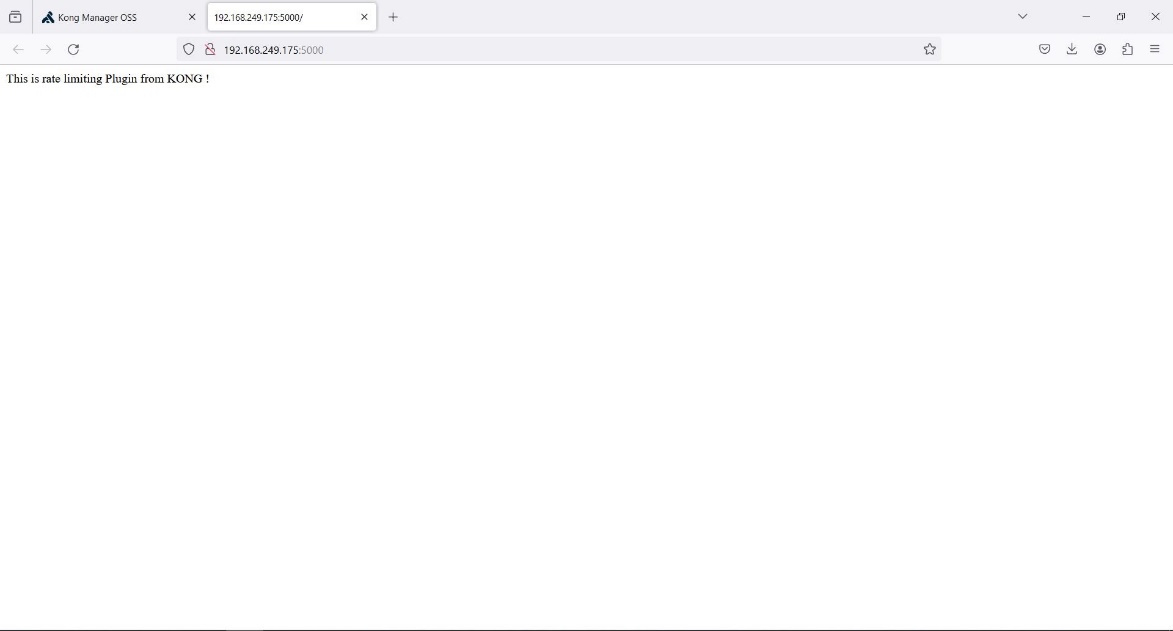
****

**Plugin 6: Rate Limiter**

**گام ها :**

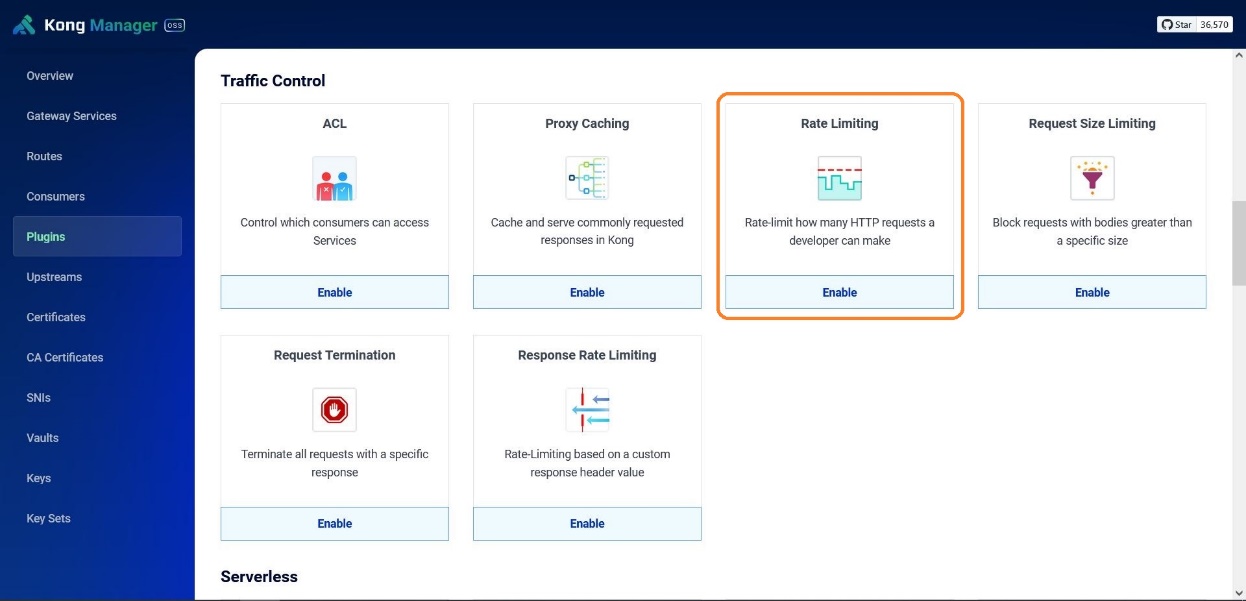
**گام ۱: ابتدا یک سرویس درست کنید (در توضیحات بالا یک سرویس ساختیم) و آن را run کنید.**

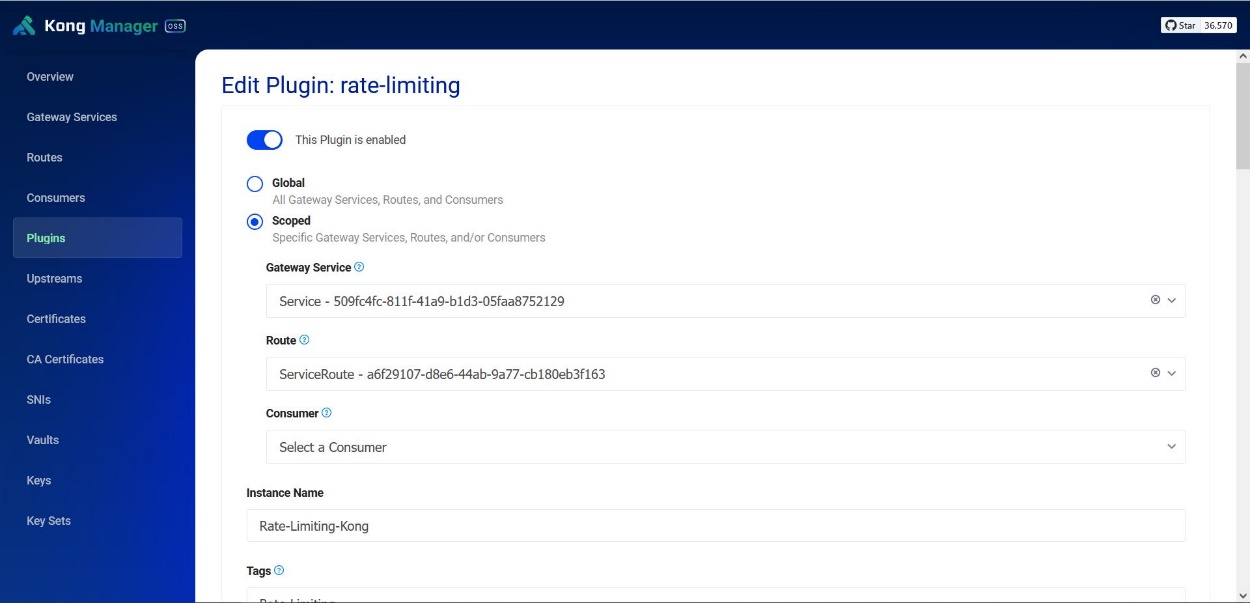
**سرویس شما روی این آدرس در حال اجرا است :** [**http://192.168.249.175:5000**](http://192.168.249.175:5000)

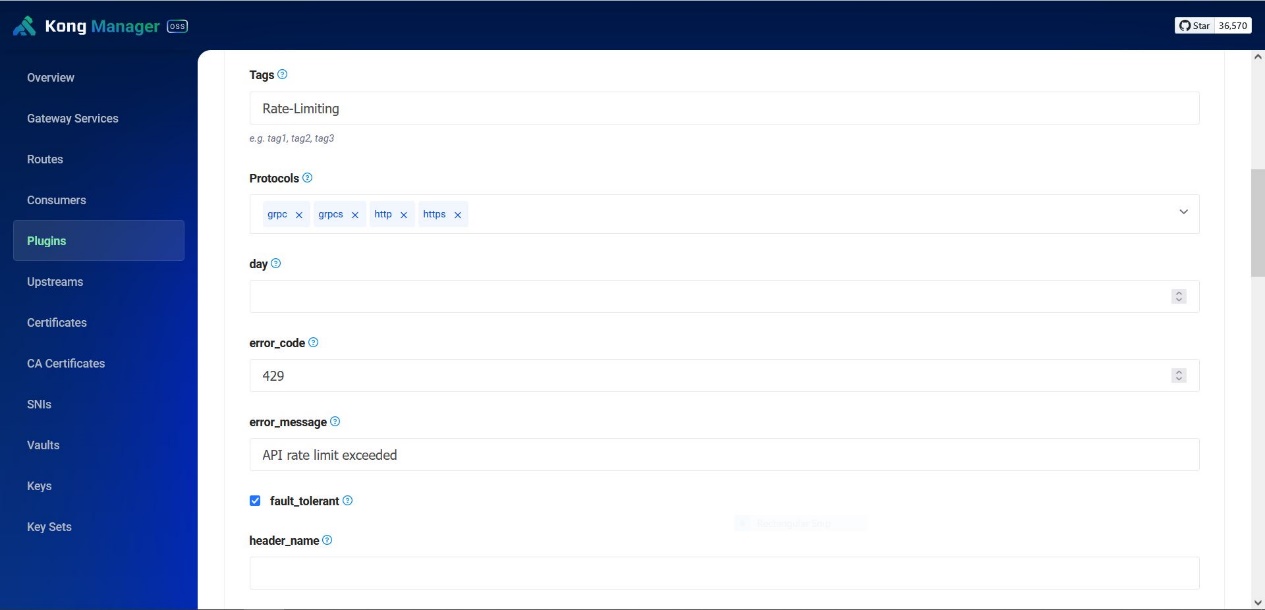
****

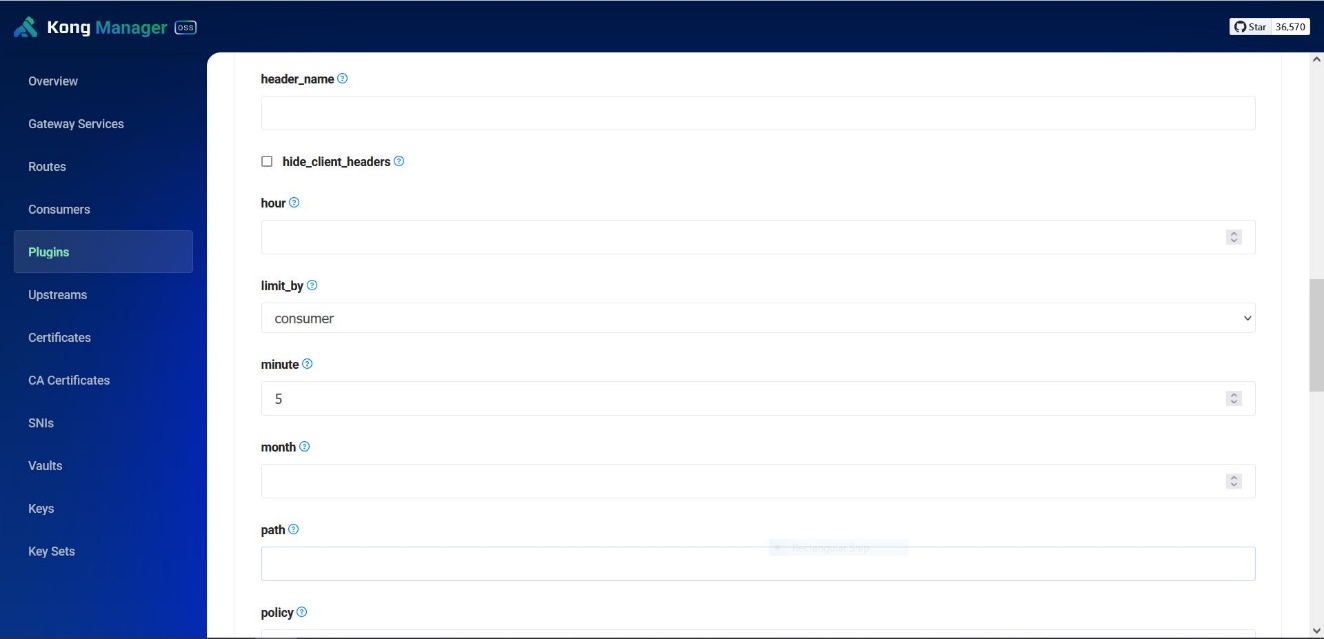
**گام ۲ : سپس یک Gateway Service و یک Router جدید درست کنید (اگر قبلا درست کرده اید لازم به ایجاد نیست)**

**گام ۳ : ایجاد پلاگین جدید :**

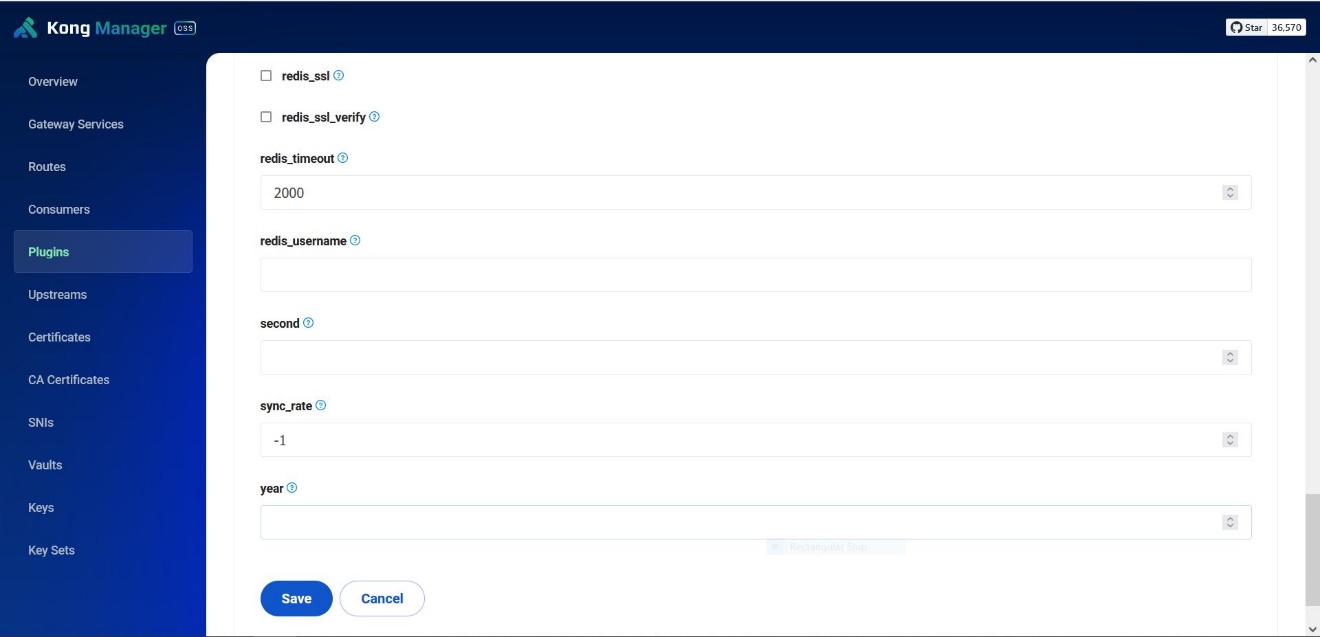
****

****

****

****

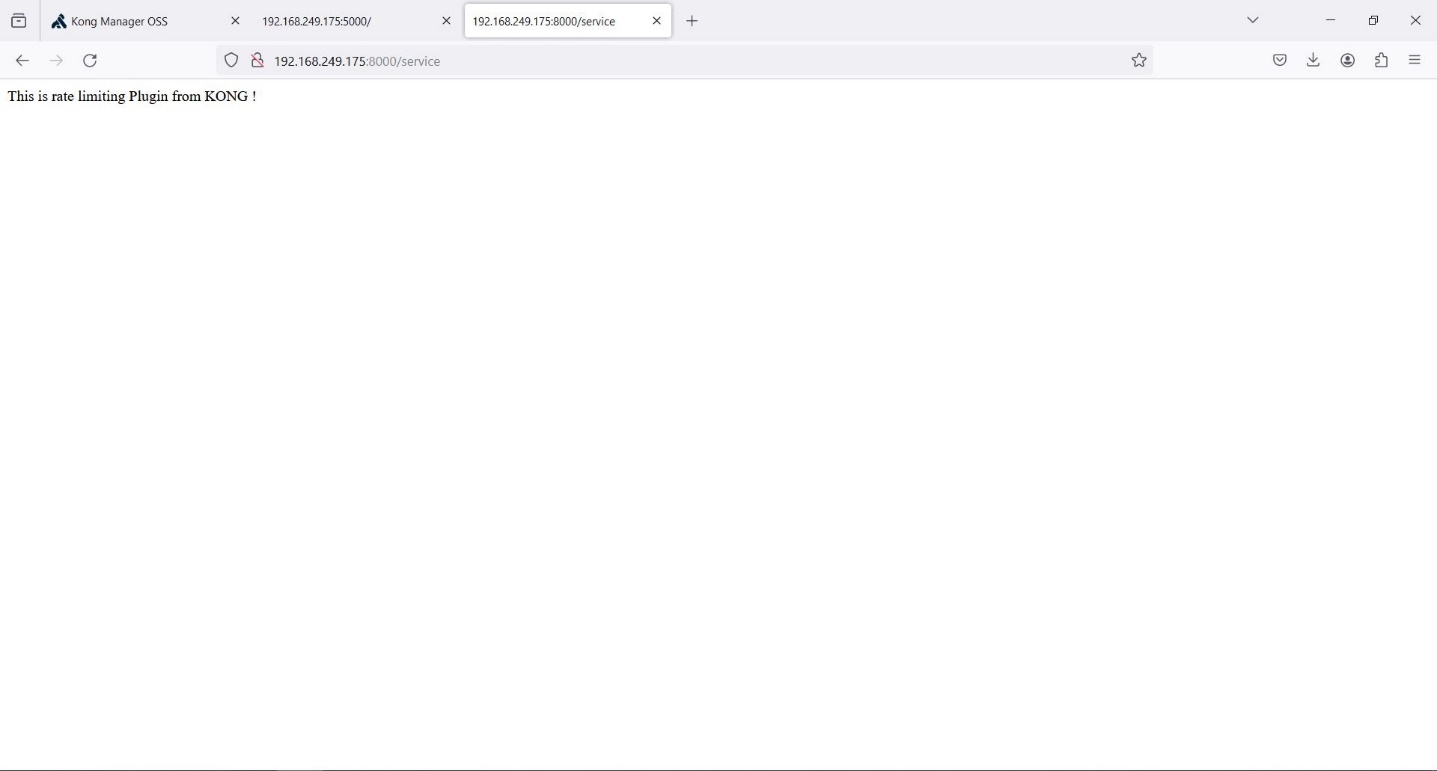
****

****

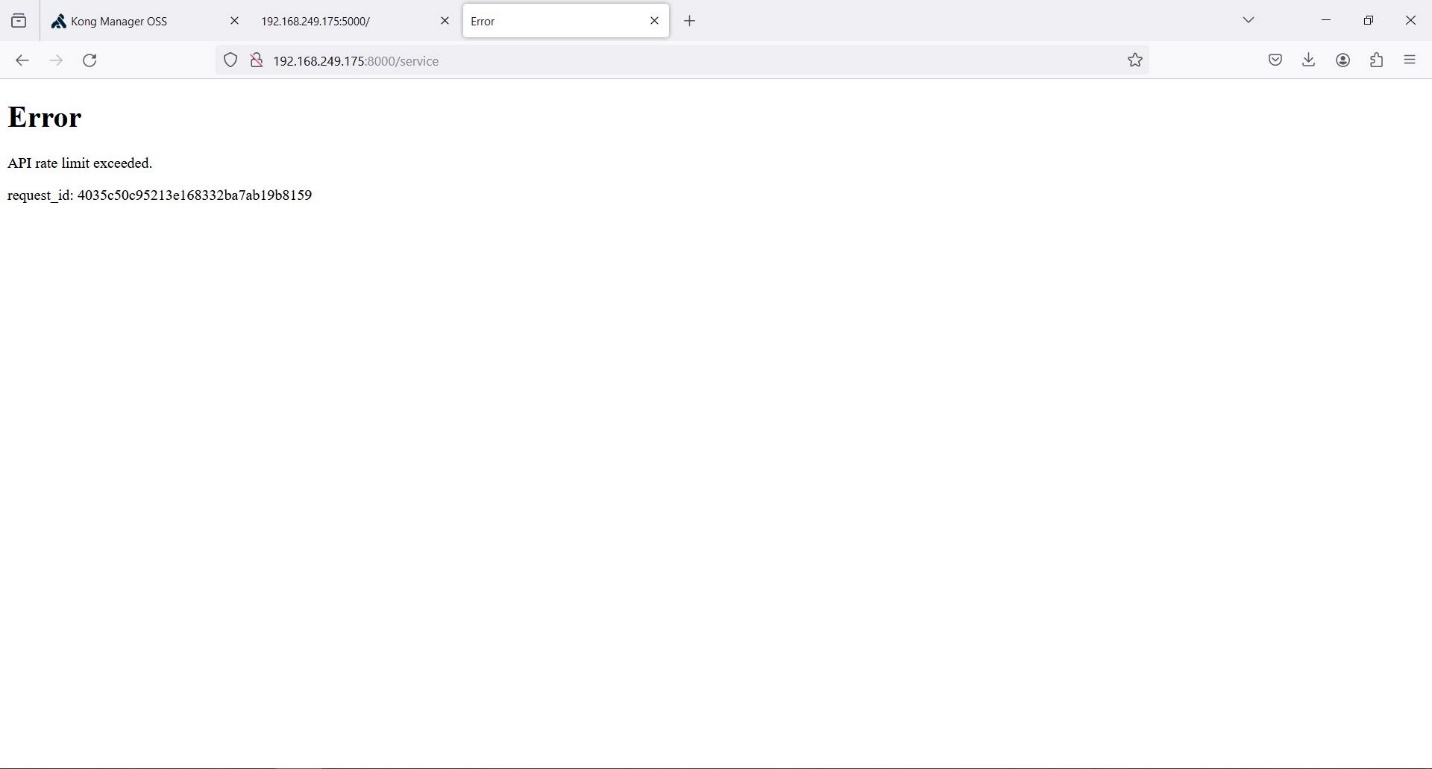
**توضیحات این پلاگین :  
در مرحله اول ما میتوانیم انتخاب کنیم که این Plugin برای همه gateway service ها انجام شود. یا فقط برای یک gateway service یا router خاصی انجام شود.  
در فیلد های بعدی انتخاب می‌کنیم. که این plugin به کدام gateway service یا router اختصاص یابد.  
در فیلد های اصلی این plugin زمان های مختلف نوشته شده که بعضی از آنها سال روز یا دقیقه است با انتخاب این فیلد ها و پر کردن این اعداد نشان میدهیم که به ازای چند Request در آن مدت زمان user میتوان Request بدهد. مثلا به ازای ۵ تا در دقیقه user بعد از دادن ۵ request هیچ response ای دریافت نمی کند. دوباره بعد از گذاشتن یک دقیقه سرویس به کاربر اجازه میدهد تا ۵ request دیگر به server ارسال کند.**

**گام ۴ : به این IP بروید :** [**http://192.168.91.175:8000/service**](http://192.168.91.175:8000/service)

**یک پیغام به نمایش در میاید : This is rate limiting Plugin from KONG !**

****

**حالا اگر ۵ بار در دقیقه کاربر Request بفرستد. دیگر به کاربر response داده نمی‌شود. و این خطا نمایش داده می شود.**

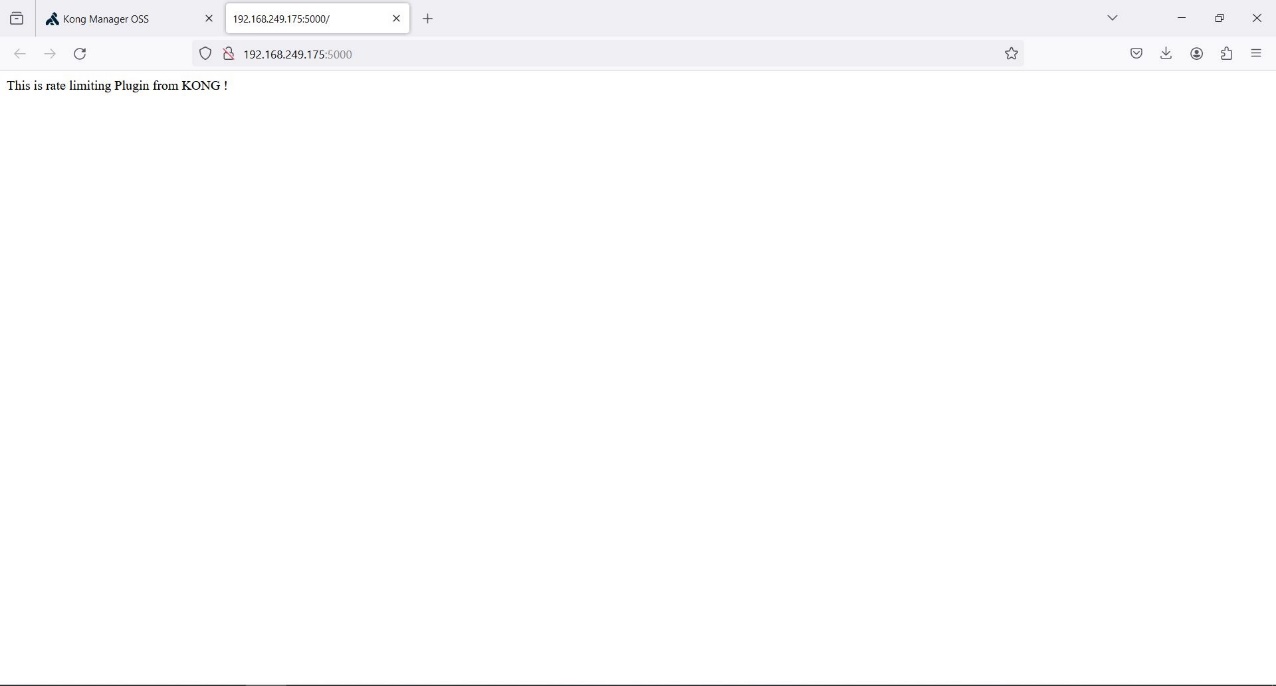
****

**حالا اگر بعد از کمی استراحت کردن سرور دوباره کاربر Request بفرستد سرور پاسخ میدهد.**

**Plugin 7: Bot Detection**

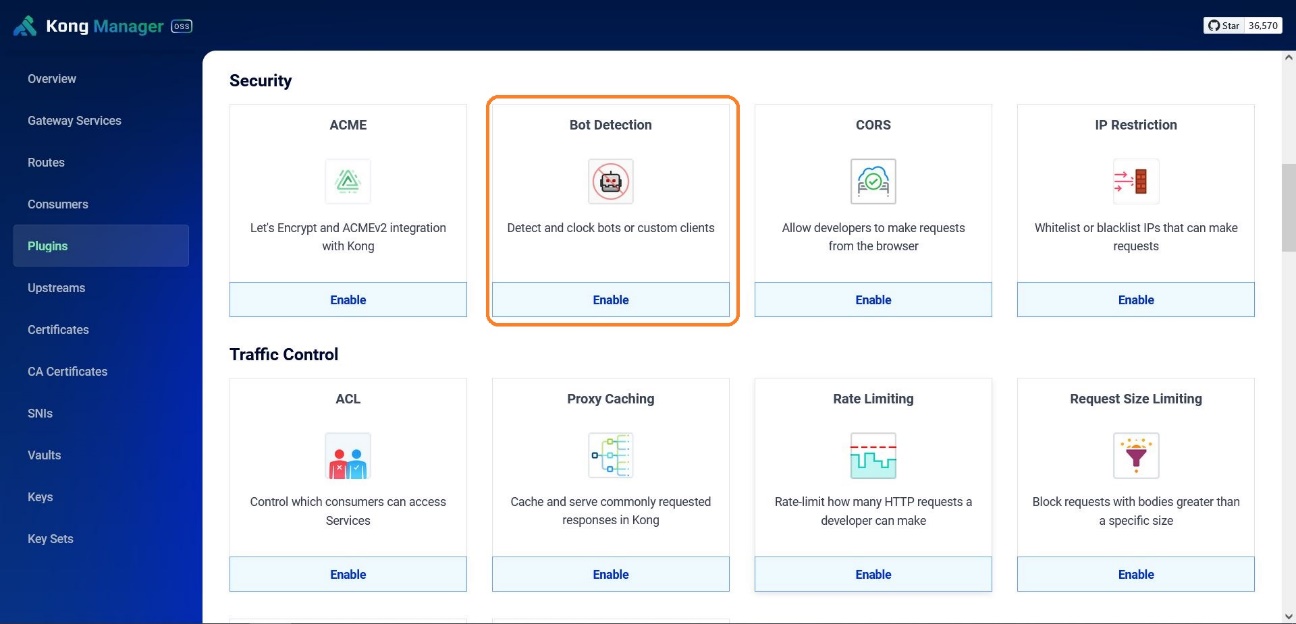
**گام ها:  
گام ۱: ابتدا یک سرویس درست کنید (در توضیحات بالا یک سرویس ساختیم) و آن را run کنید.**

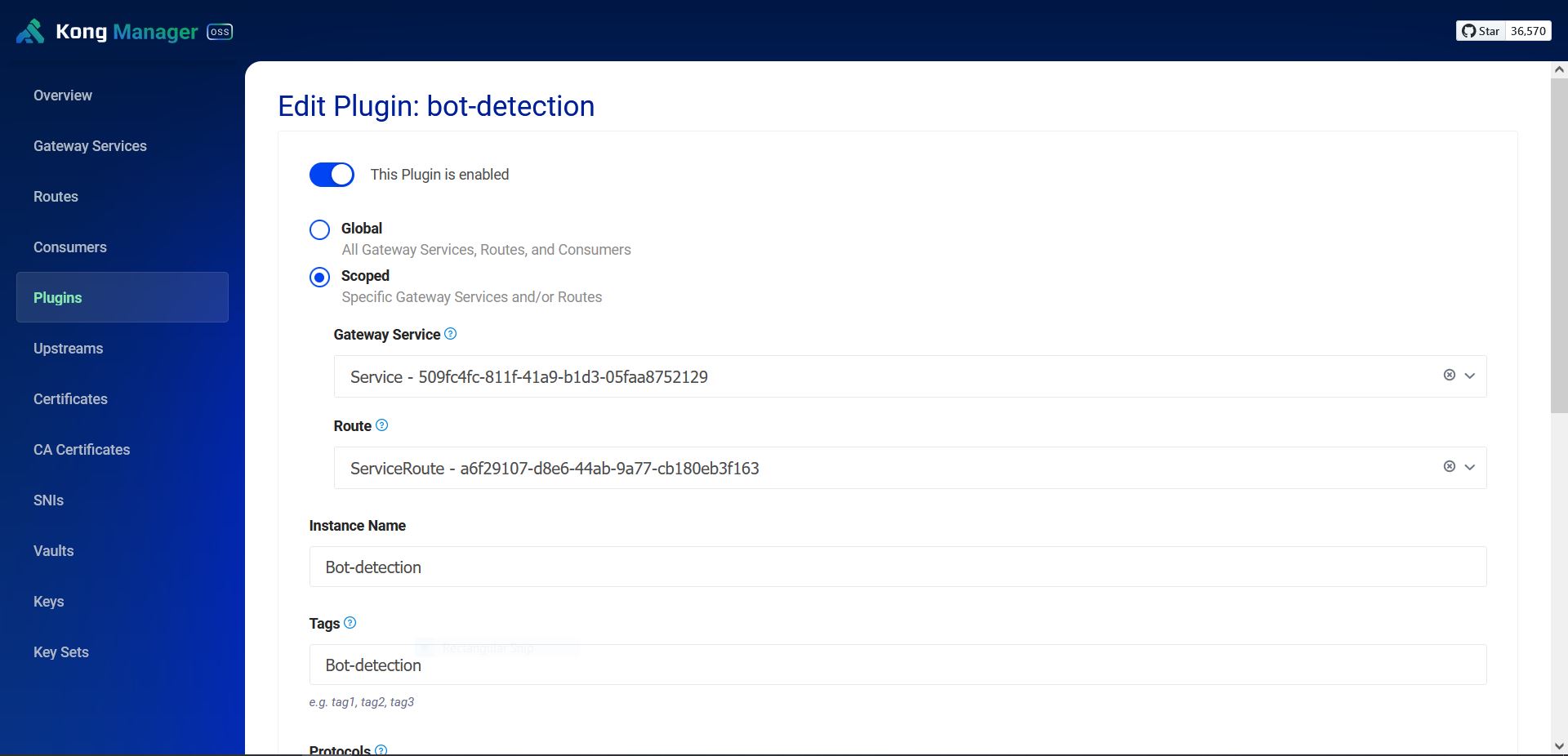
**سرویس شما روی این آدرس در حال اجرا است :** [**http://192.168.249.175:5000**](http://192.168.249.175:5000)

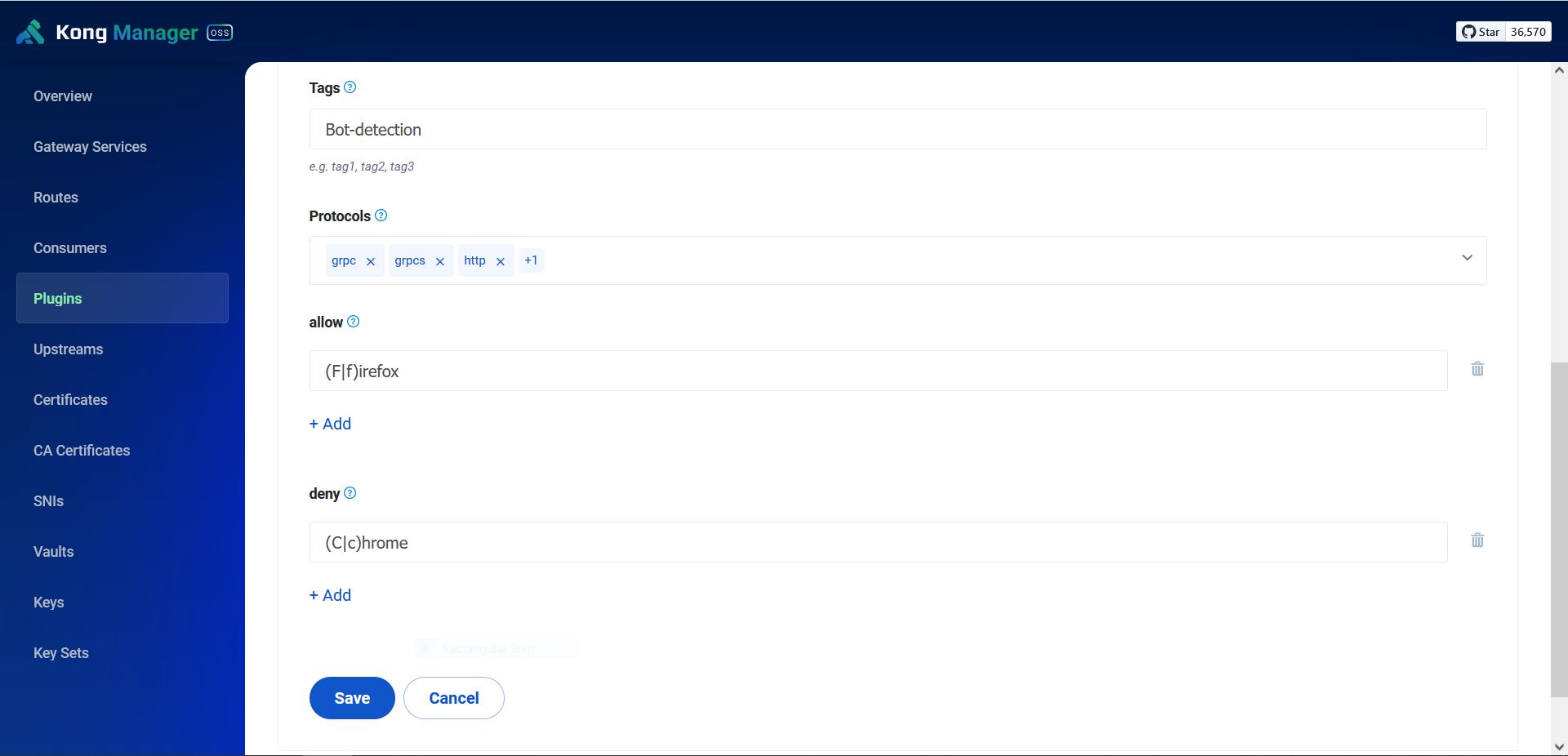
****

**گام ۲ : سپس یک Gateway Service و یک Router جدید درست کنید (اگر قبلا درست کرده‌اید لازم به ایجاد نیست)**

**گام ۳ : ایجاد پلاگین جدید :**

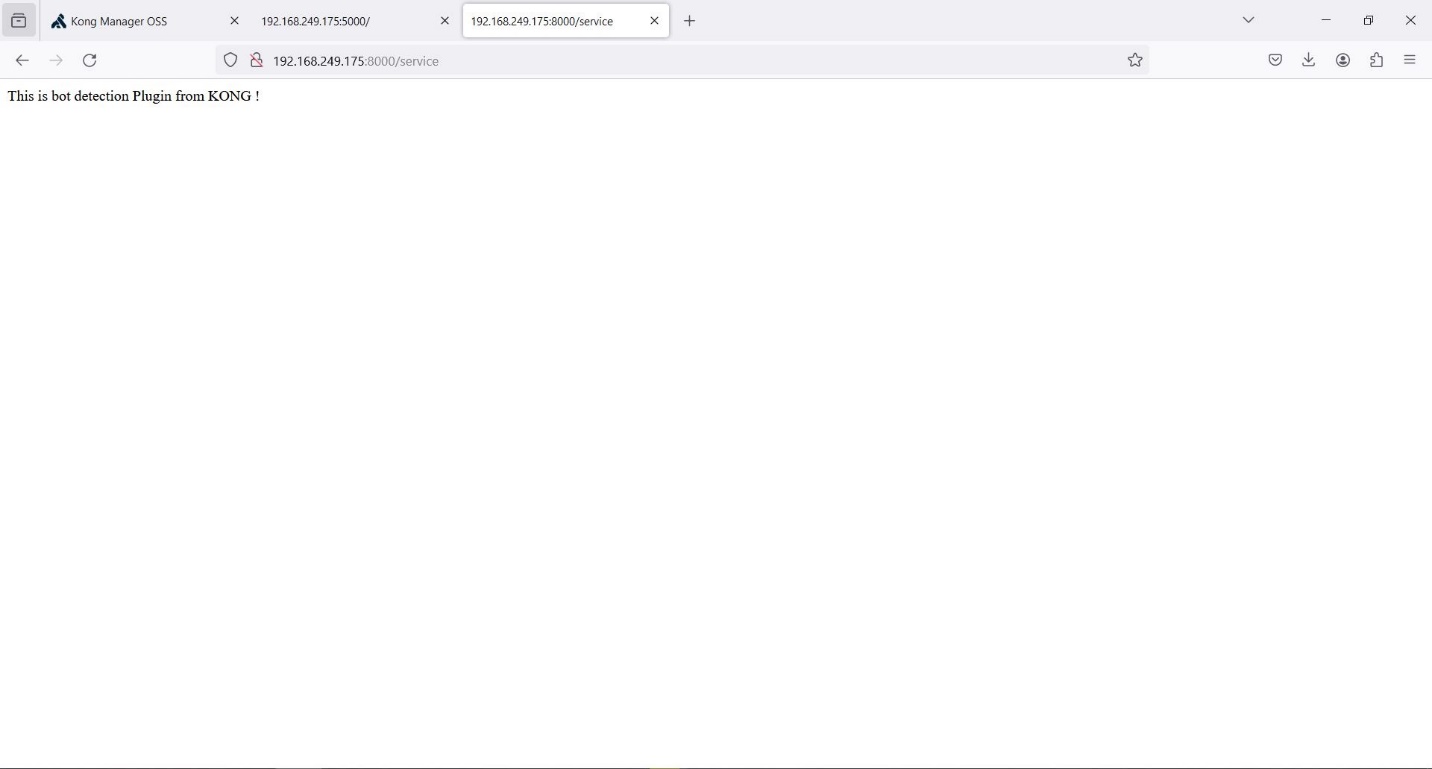
****

****

****

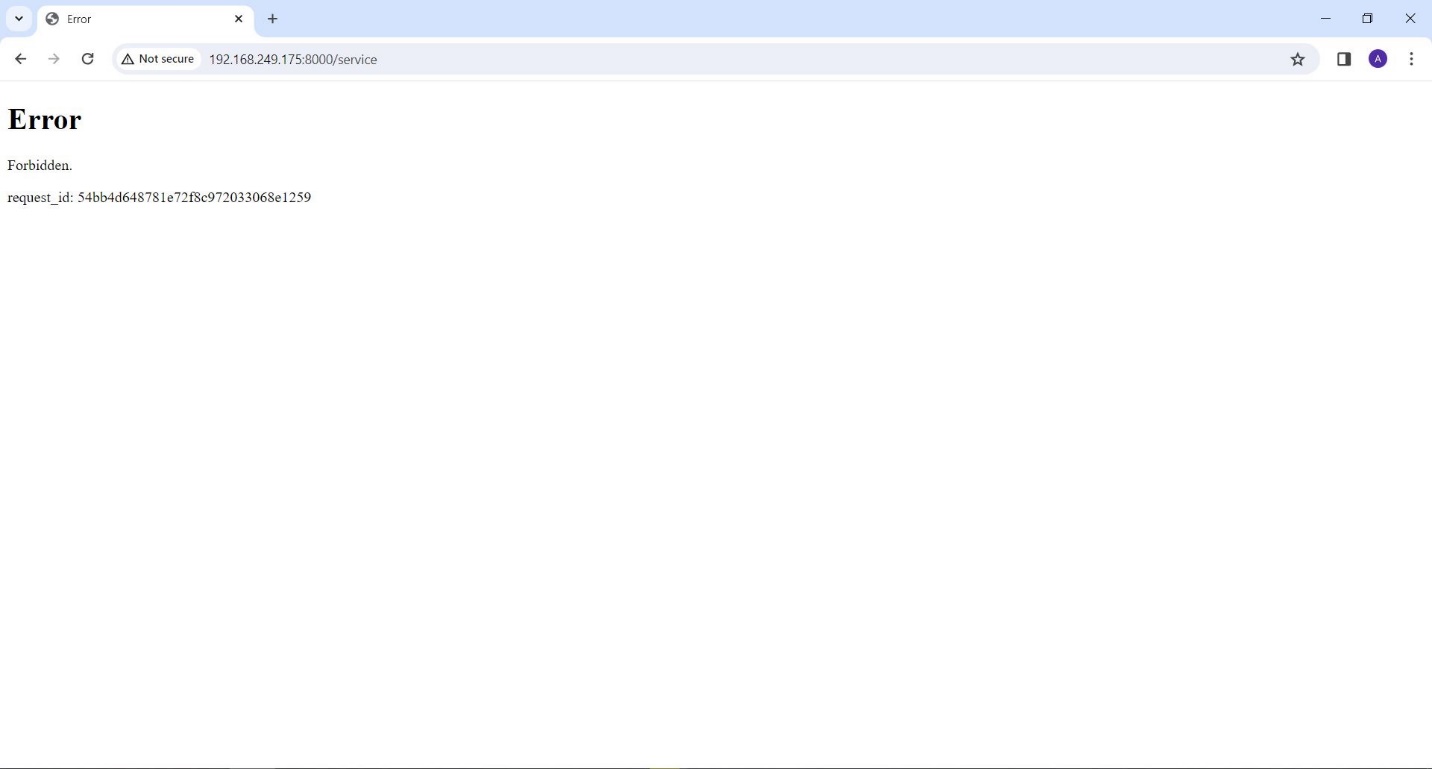
**توضیحات این پلاگین :  
در مرحله اول ما میتوانیم انتخاب کنیم که این Plugin برای همه gateway service ها انجام شود. یا فقط برای یک gateway service یا router خاصی انجام شود.  
در فیلد های بعدی انتخاب می‌کنیم. که این plugin به کدام gateway service یا router اختصاص یابد.  
در فیلد allow می نویسیم که کاربر با چه browser ای قابلیت استفاده را دارد.  
و در فیلد deny می نویسیم که کاربر با چه browser ای قابلیت استفاده ندارد.**

**گام ۴ : به این URL با browser FireFox بروید :** [**http://192.168.91.175:8000/service**](http://192.168.91.175:8000/service)

****

**یک پیغام به نمایش در میاید : This is Bot Detection Plugin from KONG !**

**با browser Chrome بروید :**

****

**در اینجا کاربر Deny شده است.**

### Plugin 8:Request Size Limiting

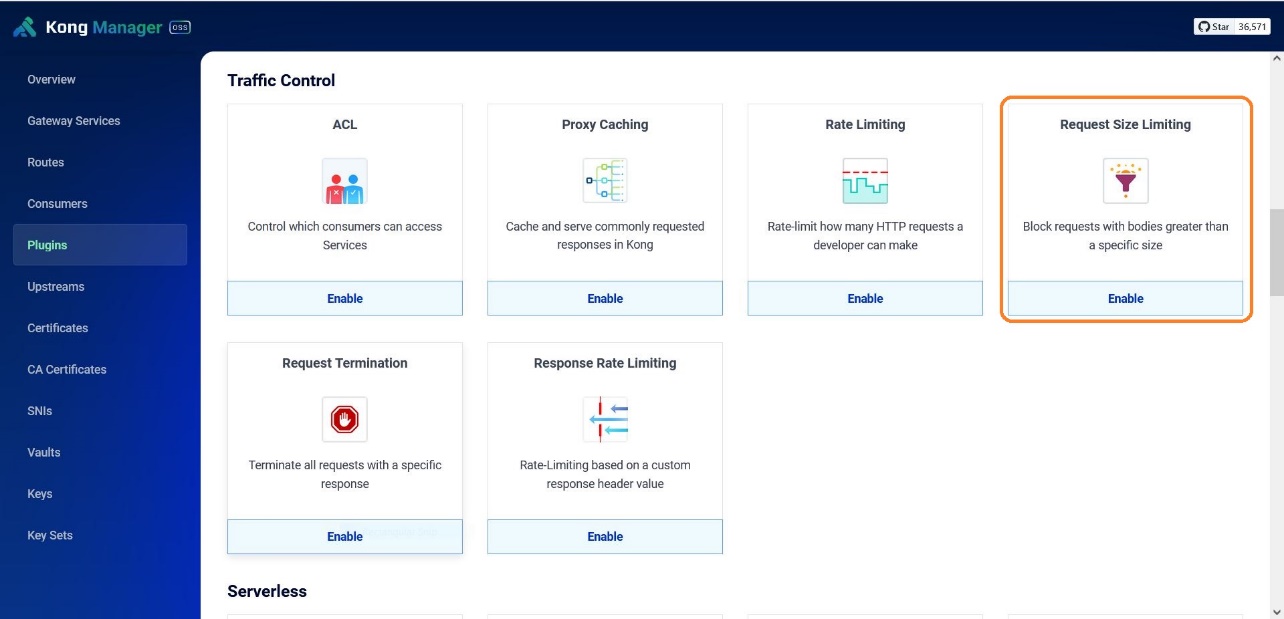
**گام ها:  
گام ۱: ابتدا یک سرویس درست کنید (در توضیحات بالا یک سرویس ساختیم) و آن را run کنید.**

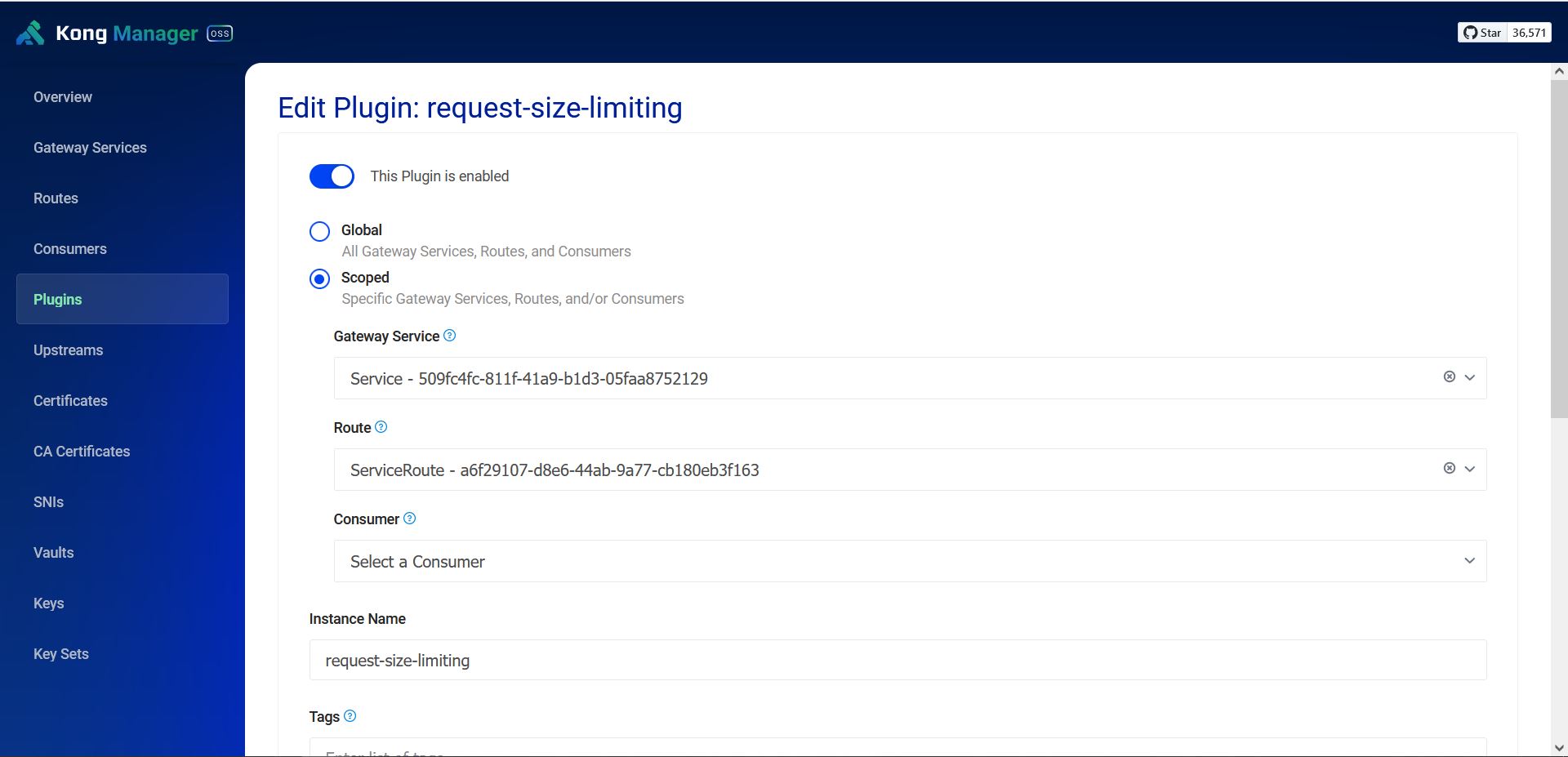
**سرویس شما روی این آدرس در حال اجرا است :** [**http://192.168.249.175:5000**](http://192.168.249.175:5000)

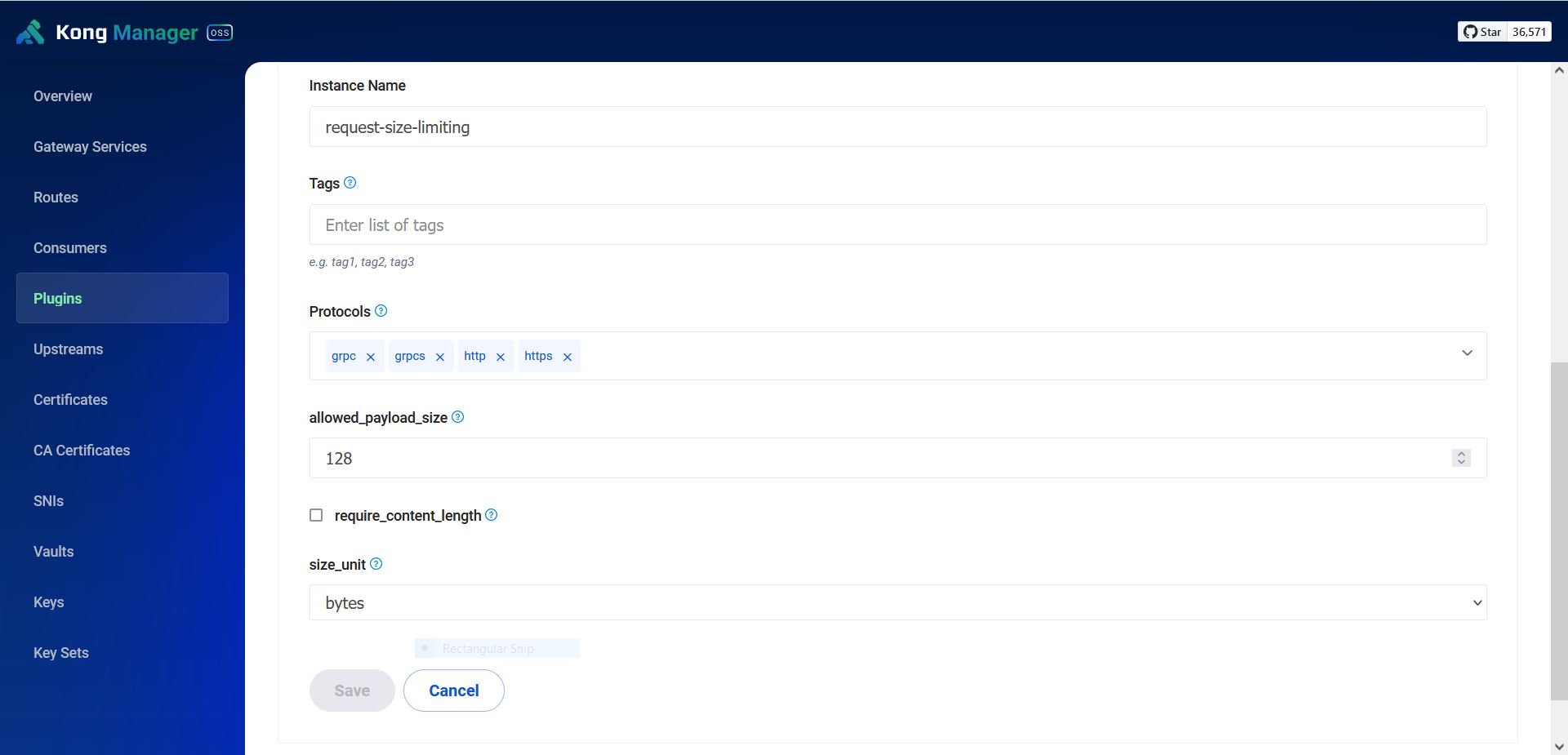
### 

**گام ۲ : سپس یک Gateway Service و یک Router جدید درست کنید (اگر قبلا درست کرده‌اید لازم به ایجاد نیست)**

**گام ۳ : ایجاد پلاگین جدید :**

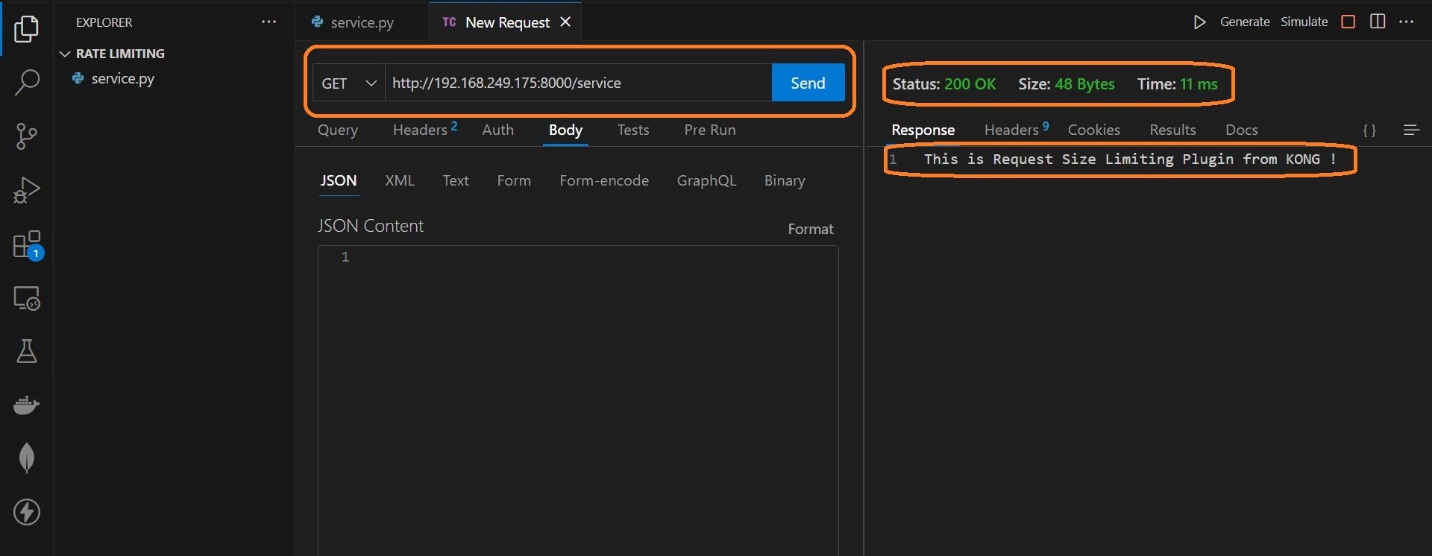
****

****

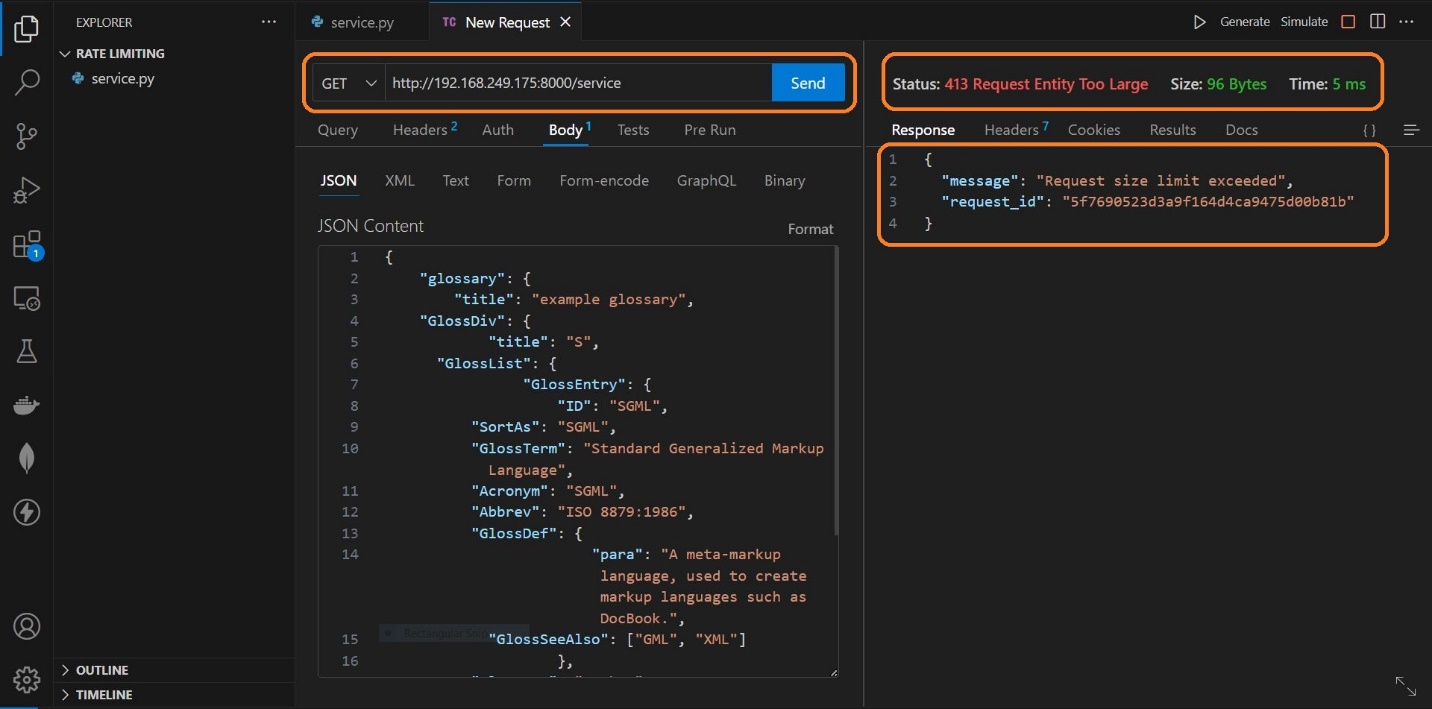
****

**توضیحات این پلاگین :  
در مرحله اول ما میتوانیم انتخاب کنیم که این Plugin برای همه gateway service ها انجام شود. یا فقط برای یک gateway service یا router خاصی انجام شود.  
در فیلد های بعدی انتخاب می‌کنیم. که این plugin به کدام gateway service یا router اختصاص یابد.  
در قسمت allowed payload size میزان حجم داده در request را مشخص میکنم و در size\_unit واحد payload size را مشخص می کنیم که در اینجا 128Byte میزان فضای مجاز برای payload است.**

**گام ۴ : حالا باید با استفاده از اکستنشن Thunder Client میزان سایز محدود شده Request را بسنجیم. URL مربوطه را در URL BAR وارد می‌کنیم.**

****

**همانطور که می بینیم status 200 است.**

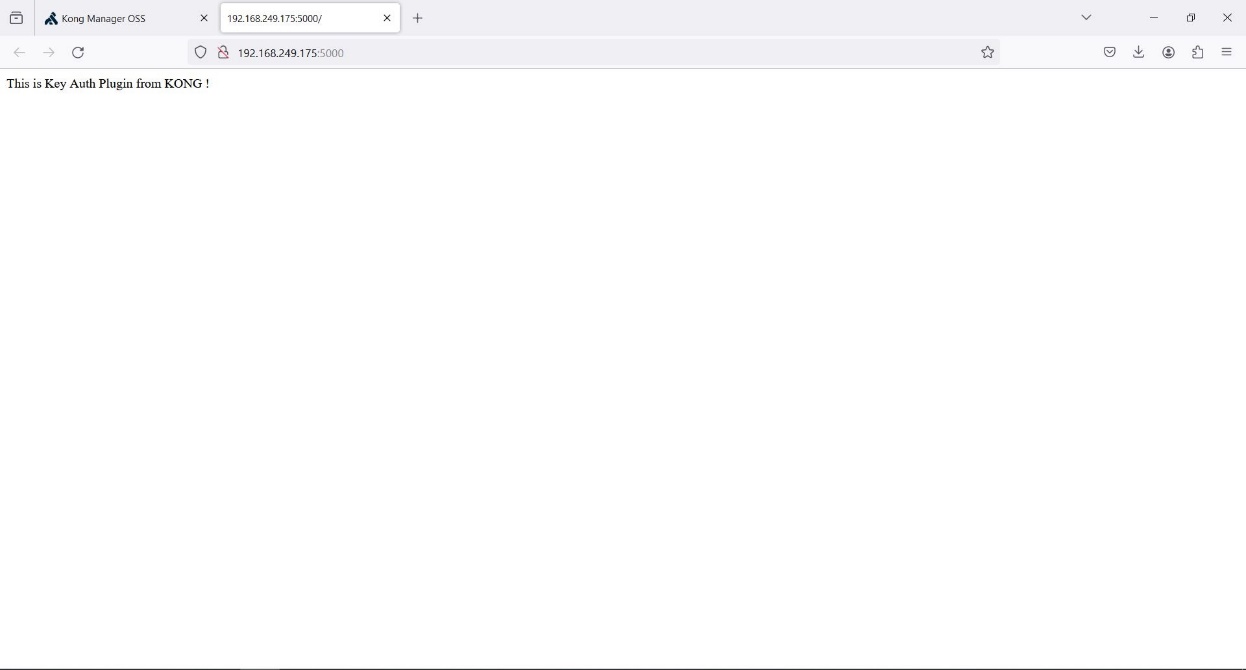
****

### گام ۵ : حال payload یا body request را با فایل json پر میکنیم. همانطور که می بینیم مقدار payload زیاد شده و خطای 413 را دریافت می کنیم.

### Plugin 9: key Auth

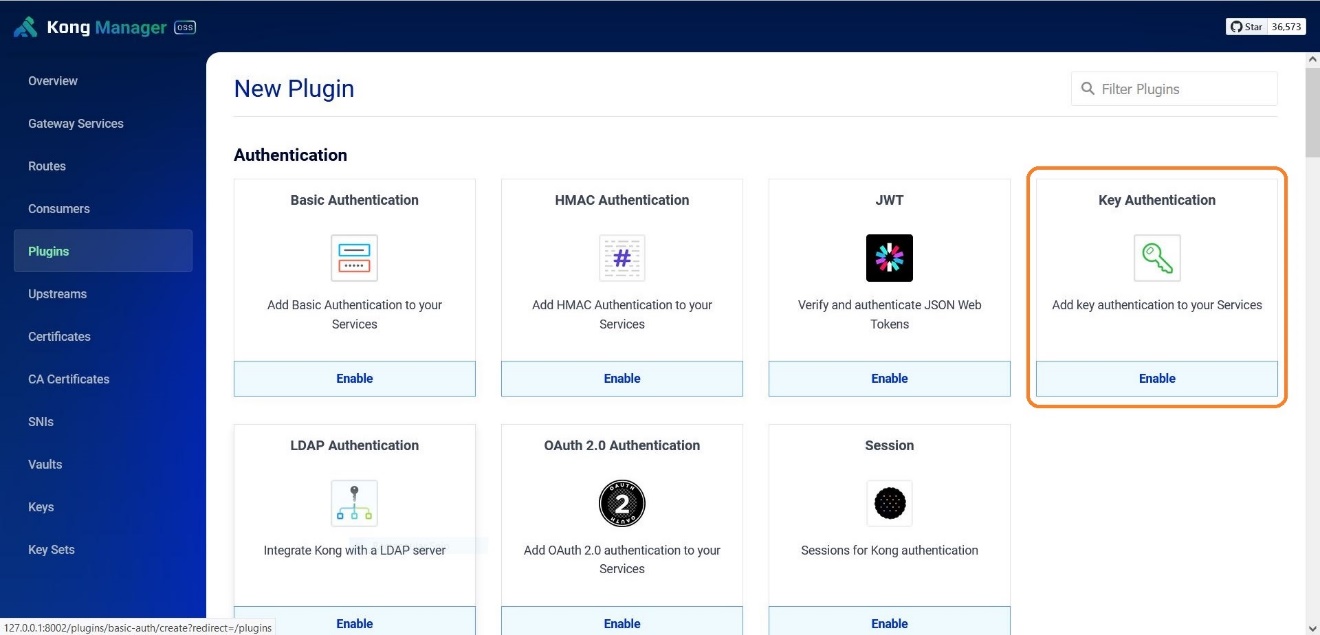
**گام ها:  
گام ۱: ابتدا یک سرویس درست کنید (در توضیحات بالا یک سرویس ساختیم) و آن را run کنید.**

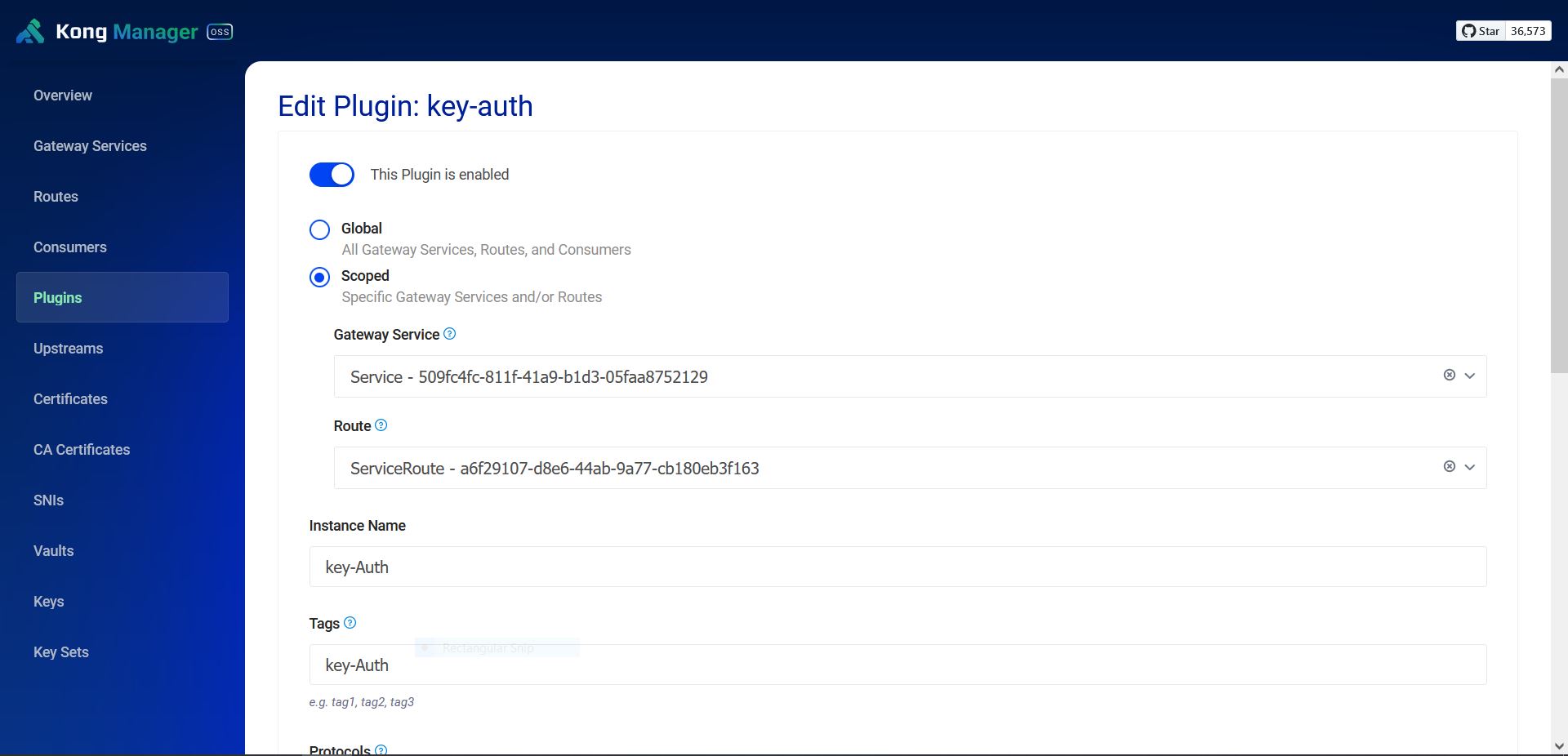
**سرویس شما روی این آدرس در حال اجرا است :** [**http://192.168.249.175:5000**](http://192.168.249.175:5000)

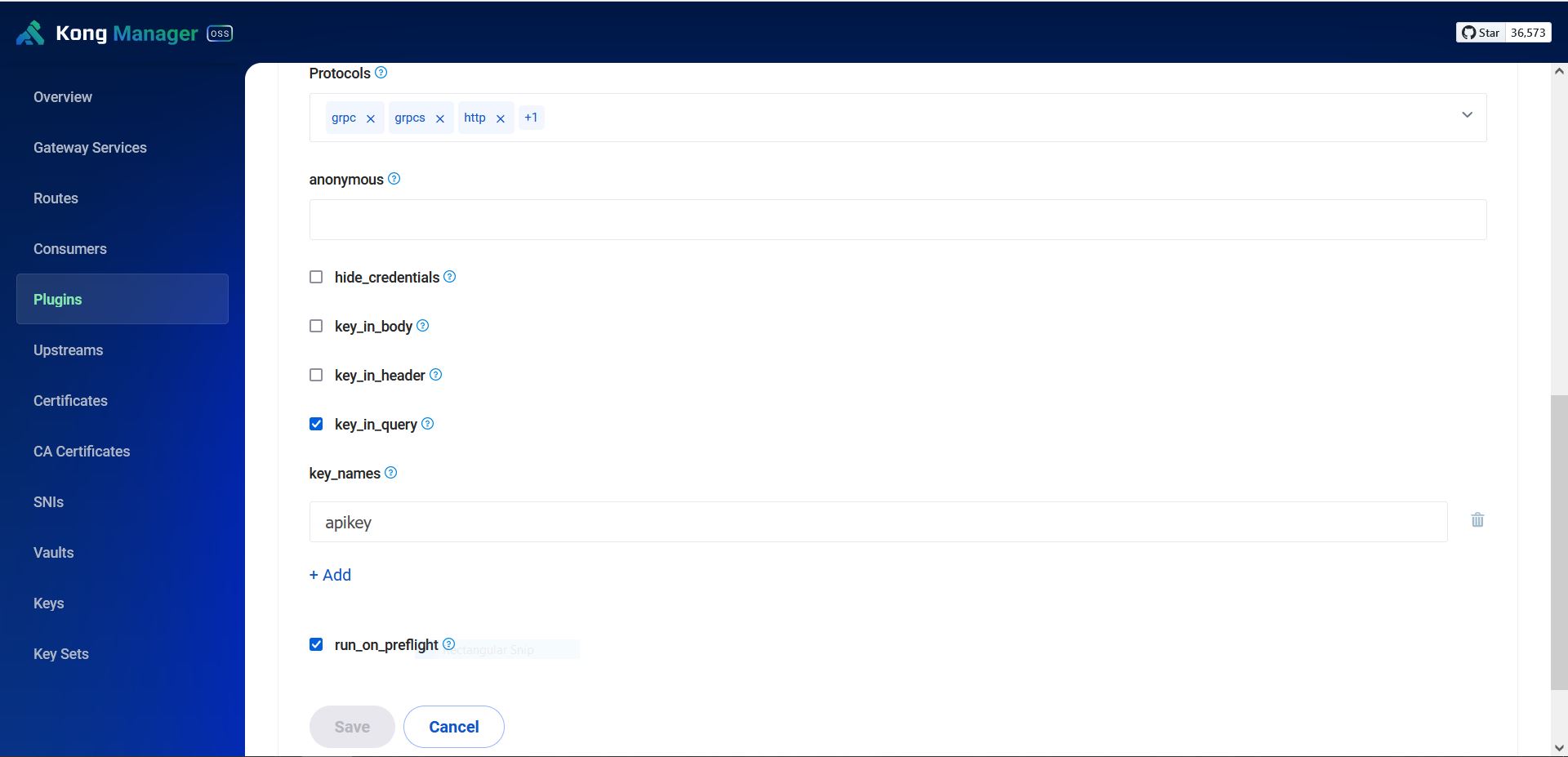
****

**گام ۲ : سپس یک Gateway Service و یک Router جدید درست کنید (اگر قبلا درست کرده‌اید لازم به ایجاد نیست)**

**گام ۳ : ایجاد پلاگین جدید :**

****

****

****

**توضیحات این پلاگین :  
در مرحله اول ما میتوانیم انتخاب کنیم که این Plugin برای همه gateway service ها انجام شود. یا فقط برای یک gateway service یا router خاصی انجام شود.  
در فیلد های بعدی انتخاب می‌کنیم. که این plugin به کدام gateway service یا router اختصاص یابد.  
در قسمت key-names باید اسم فیلد apikey را بنویسیم که در request کد بتواند key را تشخیص بدهد. همینطور باید انتخاب کنیم که key در کدام قسمت request قرار بگیرد.**

**در اینجا ما انتخاب کردیم که key‌در querry قرار بگیرد.**

**گام ۴ : حالا باید با استفاده از اکستنشن Thunder Client این url را send کنیم.**