Rate Limiting

GET /products
POST /products
GET /products/{id}
PUT /products/{id}

HTTP 429 Too many requests

Limit the calls per API

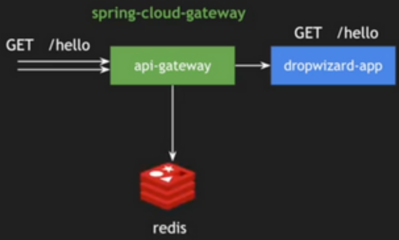
client

GET	/products	100 RPS
POST	/products	5 RPS
GET	/products/{id}	300 RPS
PUT	/products/{id}	5 RPS

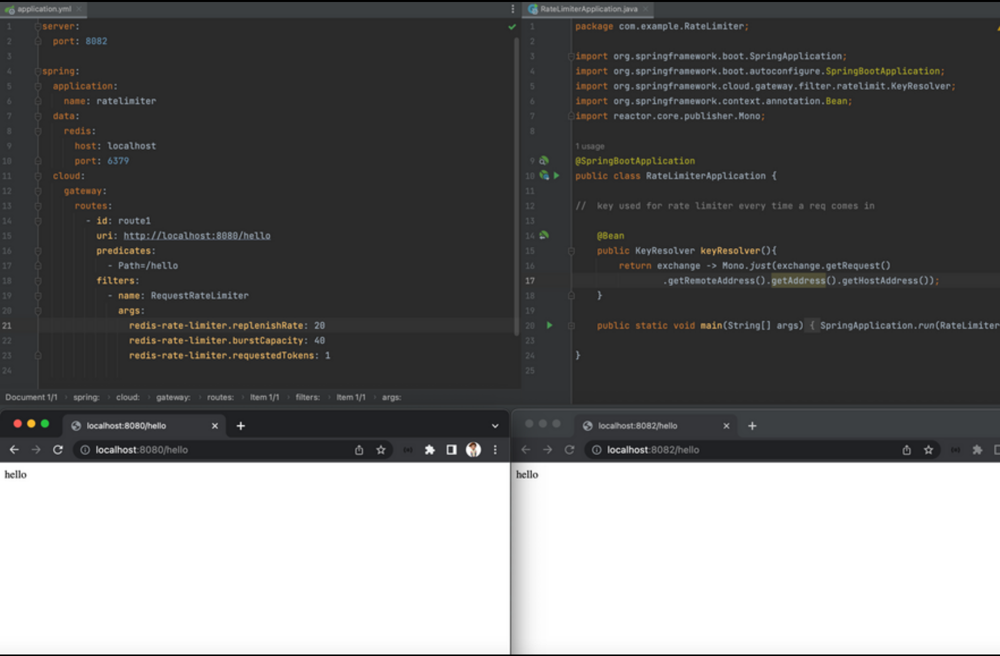
across-clients

GET	/products	3000 RPS
POST	/products	100 RPS
GET	/products/{id}	10000 RPS
PUT	/products/{id}	100 RPS

API level threshold to control incoming requests to the application



redisinsight

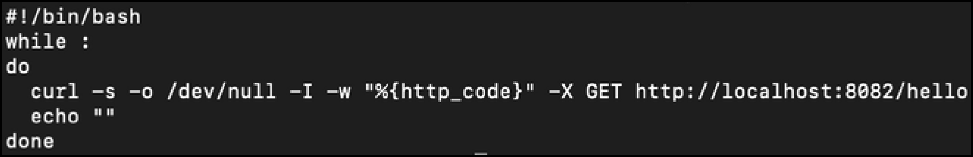

application.yml
RateLimiterApplication.java
Document 1/1 › spring: › cloud: › gateway: › routes: › Item 1/1 › filters: › Item 1/1 › args:
localhost:8080/hello
localhost:8082/hello

```yaml
server:
  port: 8082

spring:
  application:
    name: ratelimiter
  data:
    redis:
      host: localhost
      port: 6379
  cloud:
    gateway:
      routes:
        - id: route1
          uri: http://localhost:8080/hello
          predicates:
            - Path=/hello
          filters:
            - name: RequestRateLimiter
              args:
                redis-rate-limiter.replenishRate: 20
                redis-rate-limiter.burstCapacity: 40
                redis-rate-limiter.requestedTokens: 1
```

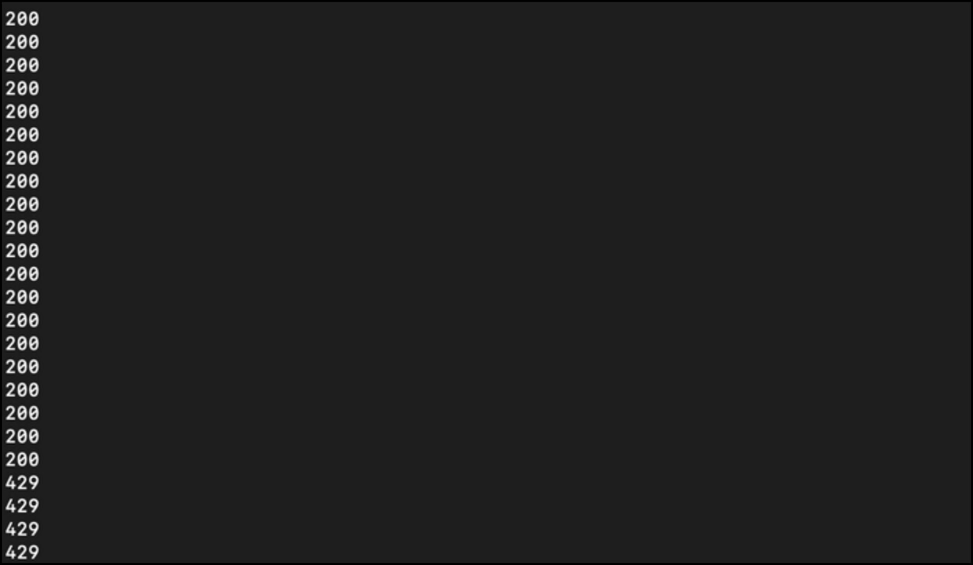
```java
package com.example.RateLimiter;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.gateway.filter.ratelimit.KeyResolver;
import org.springframework.context.annotation.Bean;
import reactor.core.publisher.Mono;

1 usage
@SpringBootApplication
public class RateLimiterApplication {

//  key used for rate limiter every time a req comes in

    @Bean
    public KeyResolver keyResolver(){
        return exchange -> Mono.just(exchange.getRequest()
                .getRemoteAddress().getAddress().getHostAddress());
    }

    public static void main(String[] args) { SpringApplication.run(RateLimiter

}
```

```
hello
```

```
hello
```

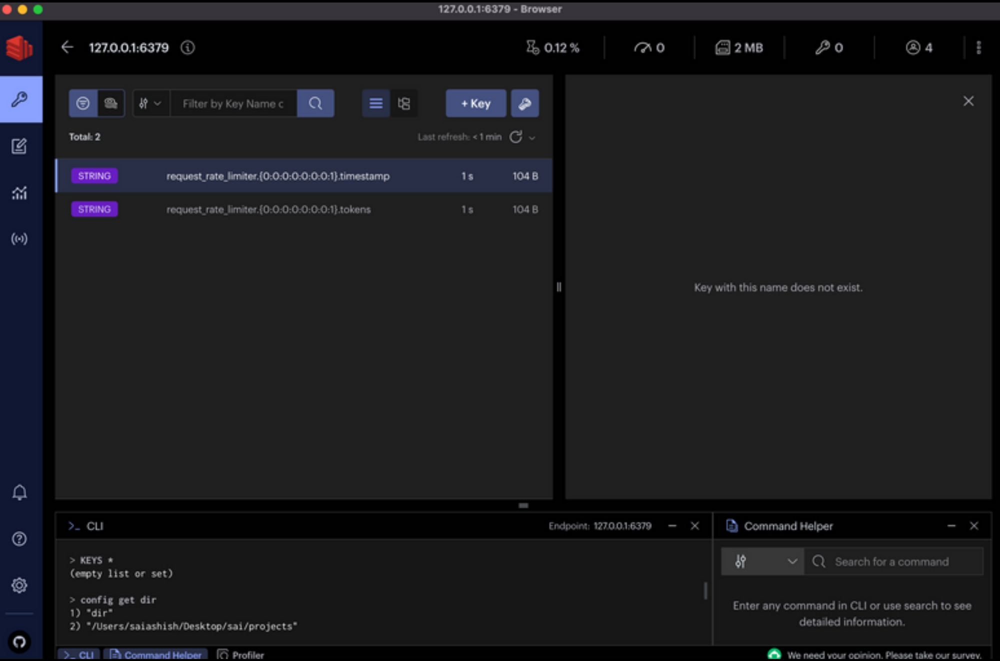


```
#!/bin/bash
while :
do
  curl -s -o /dev/null -I -w "%{http_code}" -X GET http://localhost:8082/hello
  echo ""
done
```



```
200
200
200
200
200
200
200
200
200
200
200
200
200
200
200
200
200
200
200
200
429
429
429
429
```

redis-rate-limiter.replenishRate: 20
redis-rate-limiter.burstCapacity: 40
200 Status Code : Success
429 Status Code : Too Many Requests
200 is appeared 20 times and 429 fourty times



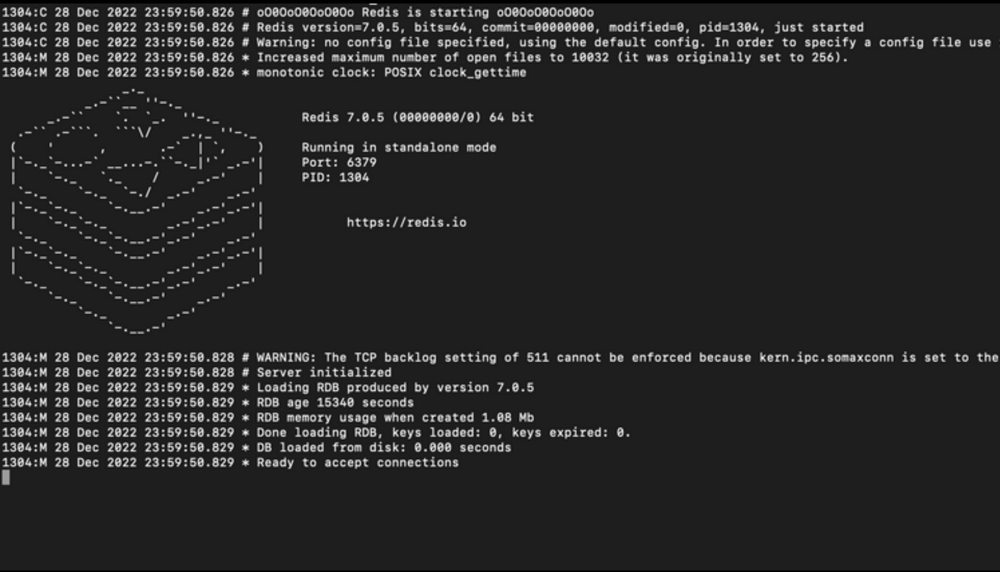
STRING	request_rate_limiter.{0:0:0:0:0:0:0:1}.timestamp	1 s	104 B
STRING	request_rate_limiter.{0:0:0:0:0:0:0:1}.tokens	1 s	104 B

Key with this name does not exist.

```
> KEYS *
(empty list or set)

> config get dir
1) "dir"
2) "/Users/saiashish/Desktop/sai/projects"
```

Enter any command in CLI or use search to see detailed information.

Rate Limiter Strings Were Appeared For Few Seconds At RedisInsight GUI At localhost:6379



```
1304:C 28 Dec 2022 23:59:50.826 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
1304:C 28 Dec 2022 23:59:50.826 # Redis version=7.0.5, bits=64, commit=00000000, modified=0, pid=1304, just started
1304:C 28 Dec 2022 23:59:50.826 # Warning: no config file specified, using the default config. In order to specify a config file use
1304:M 28 Dec 2022 23:59:50.826 * Increased maximum number of open files to 10032 (it was originally set to 256).
1304:M 28 Dec 2022 23:59:50.826 * monotonic clock: POSIX clock_gettime
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 7.0.5 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 1304
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           https://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

1304:M 28 Dec 2022 23:59:50.828 # WARNING: The TCP backlog setting of 511 cannot be enforced because kern.ipc.somaxconn is set to the
1304:M 28 Dec 2022 23:59:50.828 # Server initialized
1304:M 28 Dec 2022 23:59:50.829 * Loading RDB produced by version 7.0.5
1304:M 28 Dec 2022 23:59:50.829 * RDB age 15340 seconds
1304:M 28 Dec 2022 23:59:50.829 * RDB memory usage when created 1.08 Mb
1304:M 28 Dec 2022 23:59:50.829 * Done loading RDB, keys loaded: 0, keys expired: 0.
1304:M 28 Dec 2022 23:59:50.829 * DB loaded from disk: 0.000 seconds
1304:M 28 Dec 2022 23:59:50.829 * Ready to accept connections
```