

Title: ZuulException (SendErrorFilter) at first call

Post Body:

I am building an application by Spring Cloud ,Spring Boot and Docker. Entire application is working fine. I have couple of micro-services. Each of the project is running on Docker. When I try to consume my micro-services through Zuul API Gateway I am getting an error for the 1st call. But if I refresh the browser it's working fine. The error is given below--

```
2019-03-10 04:54:55.440 WARN [netflix-zuul-api-gateway-server,1855093598d4f99c,1855093598d4f99c true] 1 --- [nio-8765-exec-1] o.s.c.n.web.filter.ZuulErrorFilter: ZuulException (SendErrorFilter) at first call
```

I have already created the images for all my projects. And push it in the DockerHub. And the Docker-Compose file I also push it in the GitHub. Below is the path.

<https://github.com/numery009/DockerCompose/blob/master/docker-compose.yaml>

I also deploy it on the Docker Swarm on EC2. But When I try to consume my micro-services through Zuul it's not working at all. And I am getting the same 'Filter Error' for my every request.

Please help!!!.

Accepted Answer:

There should be 3 things we need to keep on the top of our head for the request which are gone through the Zuul

1) According to this Document -

https://cloud.spring.io/spring-cloud-static/spring-cloud-netflix/1.4.3.RELEASE/single/spring-cloud-netflix.html#_zuul_eager_application_context_loading

Zuul internally uses Ribbon for calling the remote url's and Ribbon clients are by default lazily loaded up by Spring Cloud on first call. This behavior can be changed for Zuul using the following configuration and will result in the child Ribbon related Application contexts being eagerly loaded up at application startup time.

application.yaml

```
zuul:
  ribbon:
    eager-load:
      enabled: true
```

application.properties

```
zuul.ribbon.eager-load.enabled= true
```

2) According to this Document - http://cloud.spring.io/spring-cloud-static/Edgware.SR3/single/spring-cloud.html#_service_discovery_configuration

Service Discovery Configuration --- If Zuul is using service discovery there are two timeouts you need to be concerned with, the Hystrix timeout (since all routes are wrapped in Hystrix commands by default) and the Ribbon timeout. The Hystrix timeout needs to take into account the Ribbon read and connect timeout PLUS the total number of retries that will happen for that service. By default Spring Cloud Zuul will do its best to calculate the Hystrix timeout for you UNLESS you specify the Hystrix timeout explicitly.

The Hystrix timeout is calculated using the following formula:

$$(\text{ribbon.ConnectTimeout} + \text{ribbon.ReadTimeout}) * (\text{ribbon.MaxAutoRetries} + 1) * (\text{ribbon.MaxAutoRetriesNextServer} + 1)$$

As an example, if you set the following properties in your application properties

application.yaml

```
ribbon:
  ReadTimeout:100
  ConnectTimeout:500
  MaxAutoRetries:1
  MaxAutoRetriesNextServer:1
```

application.properties

```
ribbon.ReadTimeout= 100 ribbon.ConnectTimeout= 500 ribbon.MaxAutoRetries= 1 ribbon.MaxAutoRetriesNextServer= 1
```

Then the Hystrix timeout (for all routes in this case) will be set to 2400ms.

In my zuul application config I have added the following properties. And it's working for my 1st call with out any error.

application.yaml

```
hystrix.command.default.execution.isolation.thread.timeoutInMilliseconds: 11000 ribbon:
  ConnectTimeout: 10000
  ReadTimeout: 10000
```

application.properties

```
hystrix.command.default.execution.isolation.thread.timeoutInMilliseconds= 11000 ribbon.ConnectTimeout= 10000 ribbon.ReadTimeout= 10000
```

3) This is the most easiest way. Disable the hystrix execution time out.

According to this Document - <https://github.com/Netflix/Hystrix/wiki/Configuration#executiontimeoutenabled>

Following property will disable the hystrix execution time out on Zuul

application.properties

```
hystrix.command.default.execution.timeout.enabled=false
```

If we keep remember these 3 scenarios then we can easily get the solution of ZuulException (SendErrorFilter).

Highest Rated Answer:

According to [this documentation](#):

Zuul internally uses Ribbon for calling the remote URLs. By default, Ribbon clients are lazily loaded by Spring Cloud on first call. This behavior can be changed for Zuul by using the following configuration, which results eager loading of the child Ribbon related Application contexts at application startup time.

The following example shows how to enable eager loading:

```
# application.yml zuul:  ribbon:    eager-load:      enabled: true
```

Or

```
# application.properties ribbon.eager-load.enabled = true
```

You might need to check the following related issues:

- [Ryan Baxter's comment](#)
- [First request through Zuul/Eureka will always timeout. Subsequent requests then behave as expected \(no timeouts\)](#)