Distributed Tracing

Lecture 92: Introduction to distributed Tracing

Centralized tracing mechanism: using sleuth & zipkins.
The concept is “attach a unique ID to each request to track which all micro services it passed through”
Spring Sleuth : This component attaches a unique ID to the system
Zipkins: A distributed Tracing mechanism .

How this is achieved , is the logs for this request (identified by the unique ID is put in a MQ implemented by Rabbit MQ , which is then transmitted to the Zipkins server.

Lecture 93: Add Spring Sleuth

Perform the below action to NZ-API Gateway Service, CES,CCS

→ Add dependency
→ Add Sampler (as a bean in the above mentioned Application) to tell which type/pattern of request you need to filter/track.
→ Add logger at respective methods in respective controllers(ZuleLoggingFilter.java,CurrencyExchangeController.java,CurrencyConversionController.java)

Launch NamingServer Service-
Launch CES-
Launch CCS-
Launch NZ-API Gw Service-

Give a call to CCS and u can see the logs in NZAPI GW,CES,CCS with a specific ID.
Traversing this through various console is difficult when there are 100s or m/s. Thus wee need to send this to a centralized Logging service.

Centralize the Logging server:

Note: ----------------
There are many centralized Logging Solutions available , most popular are :

1. ELK stack [Elastic Search, Logstash , Kibana]- managed by Elastic

   Lucene → indexing+ search
   ElasticSearch → distributed search capabilities
   Logstash → to all kind of Tanseries data
   Kibana → Visualization tool

Output => QBOX (Capabilities of Tablue, Jaspersoft etc)

Zipkins uses the above technology to centralize this . The topology we’ll use is as follows:

M/S --→ RabbitMQ ---→ Zipkins ---→ Database

Micro services will push all their logs to RabbitMQ , Zipkins will collect it from it & push it to its DB)

Step1: Install MQ
(Windows)
→ Pre-requsites : Install a version of **Erlang** . Which version to install , see the links in tutorial
→ download and install Rabbit MQ.(Any version is fine ,from **Github** or **Bintray**)
(Mac)
→ Use **Homebrew**.
→ make sure u download the latest version of Homebrew
→ add the folder where it is installed to your path

Now , Start MQ Server

Step2: Install ZipKins
--Config & Download the Zipkins from springs initializer: This is no more supported from spring Finchley : 2.0.0.M3
– We'll download from Zipkins from zipkins.io page
– Docker version
– Java version (Recomended)
– copy the link provided on the Quick start page and paste it on browser , this will download a zipkins.jar

Step3: Connect RabbitMQ to Zipkins
– (On windows):

```
prompt$> SET RABBIT_URI=amqp://localhost
```

prompt$> java ~~jar zipkin~~server-2.5.2-exec.jar

– http://localhost:9411/zipkin → verify that it is installed

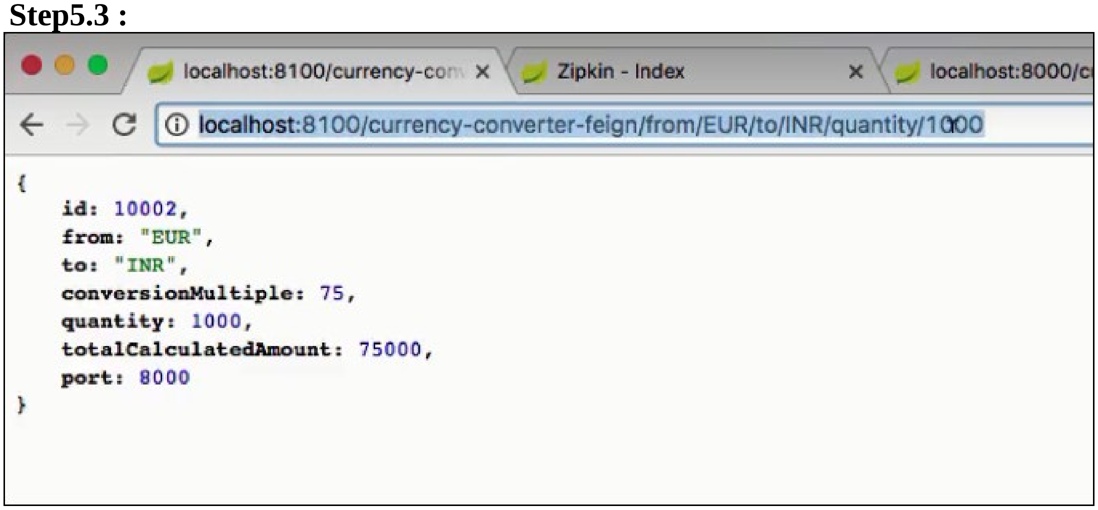
Step4: Add snippets to m/s to push logs to rabbit MQ
– add dependency in pom.xml of all these m/s
spring-cloud-sleuth-zipkin: tells spring to log messages in zipkins format
spring-cloud-starter-bus-amqp: tells springs that the default MQ is rabbit MQ.
Step5: Using Zipkins UI
Step 5.1 : Launching order :

1. Eureka Naming Server
2. Zipkins Distributed Tracing Server
3. CES
4. CCS
5. NZ-API Gateway Service

allow enough time between each application launch.

Step5.2 : Verify on Eureka Naming Server
http://localhost:8761/

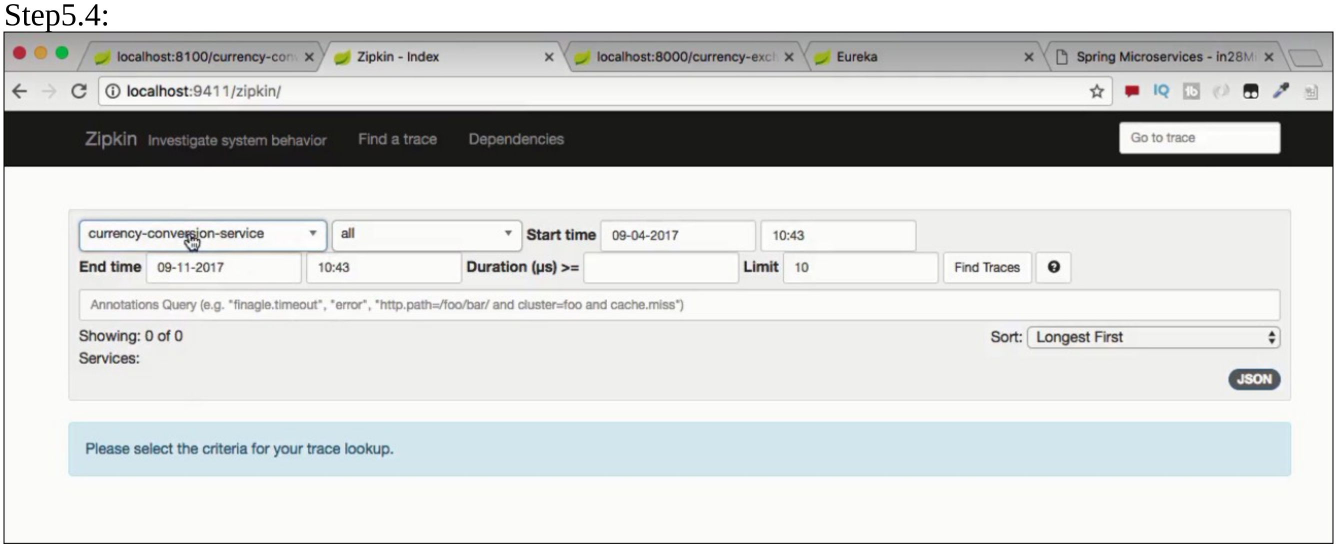
Step5.3 :



localhost:8100/currency-converter-feign/from/EUR/to/INR/quantity/1000

```
{
    id: 10002,
    from: "EUR",
    to: "INR",
    conversionMultiple: 75,
    quantity: 1000,
    totalCalculatedAmount: 75000,
    port: 8000
}
```

Step5.4:


localhost:8100/currency-con
Zipkin - Index
localhost:8000/currency-exc
Eureka
Spring Microservices - in28M
Zipkin Investigate system behavior
Find a trace
Dependencies
Go to trace
Find Traces
JSON

localhost:9411/zipkin/

currency-conversion-service | all | Start time 09-04-2017 | 10:43

End time 09-11-2017 | 10:43 | Duration (μs) >= | Limit 10

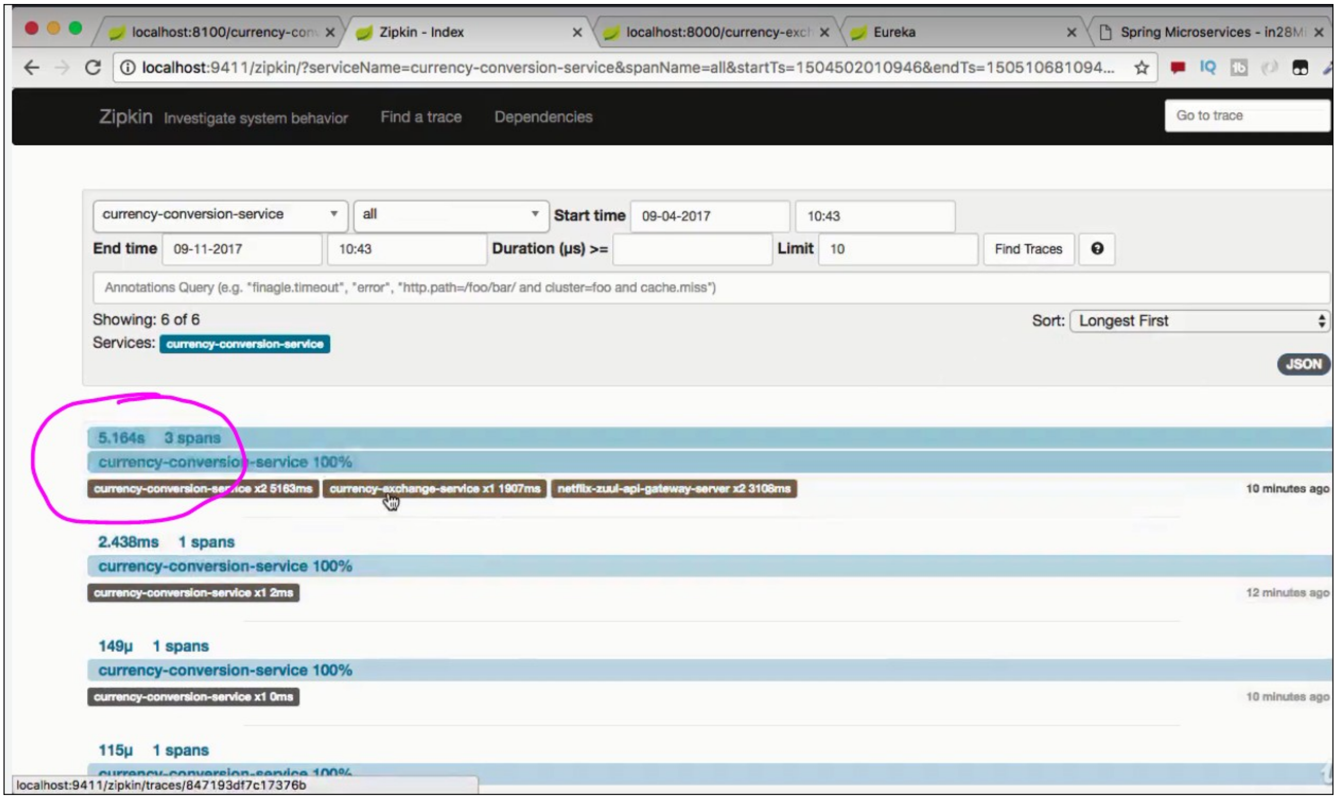
Annotations Query (e.g. "finagle.timeout", "error", "http.path=/foo/bar/ and cluster=foo and cache.miss")

Showing: 0 of 0

Services:

Sort: Longest First

Please select the criteria for your trace lookup.



localhost:8100/currency-con
Zipkin - Index
localhost:8000/currency-exch
Eureka
Spring Microservices - in28Mi
IQ
Zipkin Investigate system behavior
Find a trace
Dependencies
Go to trace
Find Traces
JSON

localhost:9411/zipkin/?serviceName=currency-conversion-service&spanName=all&startTs=1504502010946&endTs=150510681094...

currency-conversion-service | all | **Start time** 09-04-2017 | 10:43

End time 09-11-2017 | 10:43 | **Duration (μs) >=** | **Limit** 10

Annotations Query (e.g. "finagle.timeout", "error", "http.path=/foo/bar/ and cluster=foo and cache.miss")

Showing: 6 of 6

Sort: Longest First

Services: currency-conversion-service

5.164s 3 spans
currency-conversion-service 100%
currency-conversion-service x2 5163ms | currency-exchange-service x1 1907ms | netflix-zuul-api-gateway-server x2 3108ms
10 minutes ago

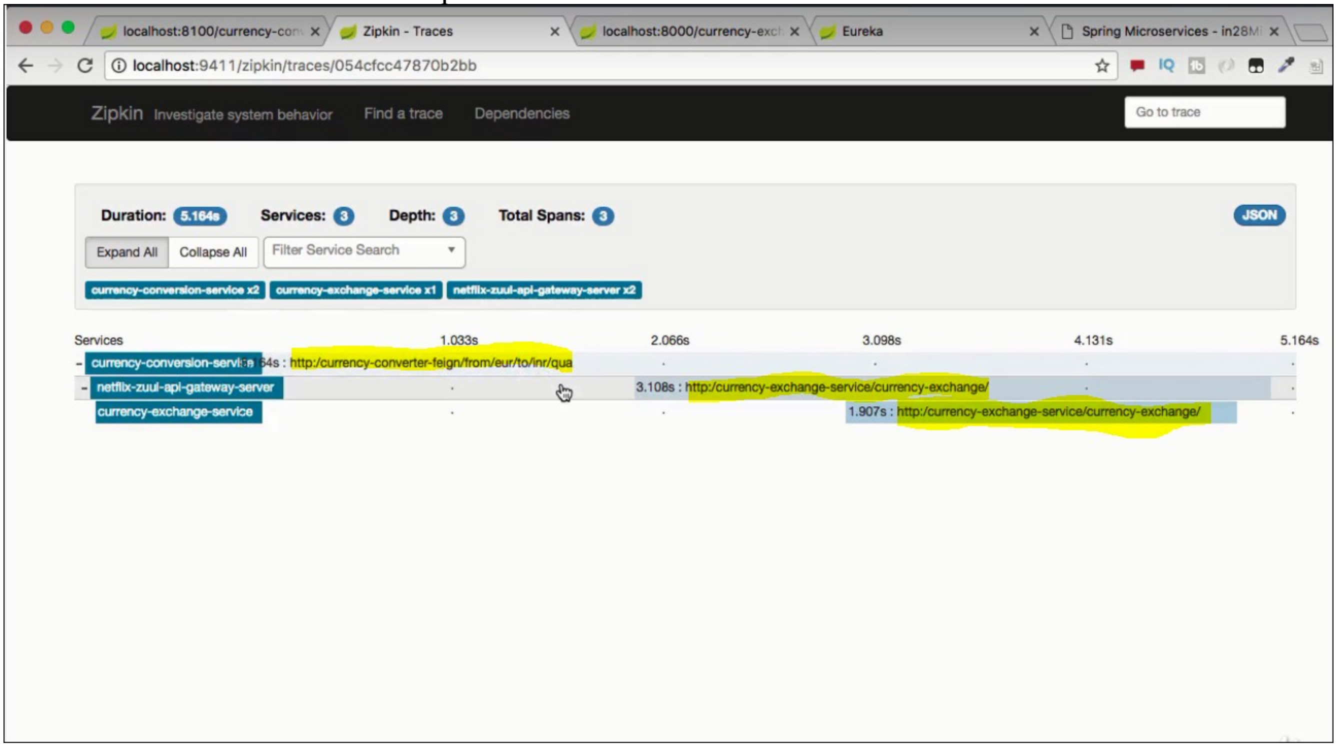
2.438ms 1 spans
currency-conversion-service 100%
currency-conversion-service x1 2ms
12 minutes ago

149μ 1 spans
currency-conversion-service 100%
currency-conversion-service x1 0ms
10 minutes ago

115μ 1 spans
currency-conversion-service 100%

localhost:9411/zipkin/traces/847193df7c17376b

Now u can see the trace of that request:



localhost:8100/currency-con
Zipkin - Traces
localhost:8000/currency-exch
Eureka
Spring Microservices - in28Mi
IQ
Zipkin Investigate system behavior
Find a trace
Dependencies
Go to trace
JSON
Expand All
Collapse All
Filter Service Search

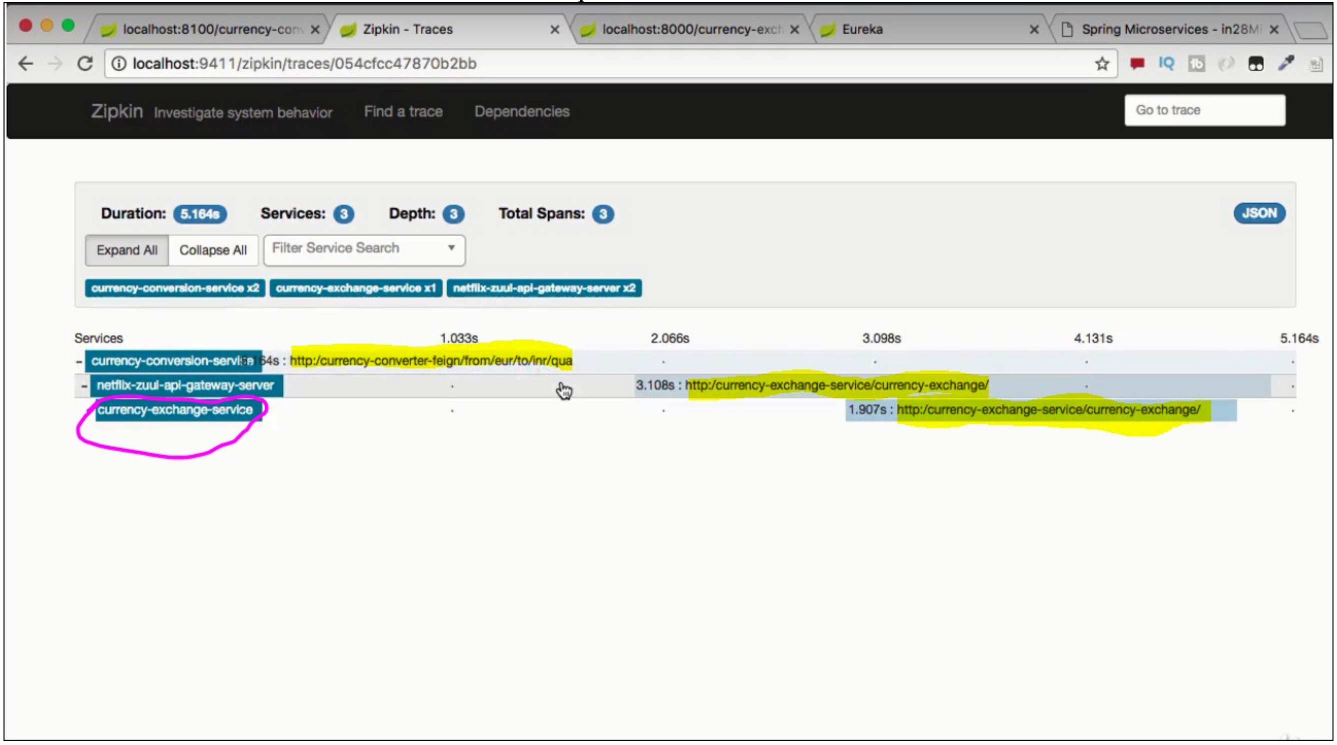
localhost:9411/zipkin/traces/054cfcc47870b2bb

Duration: 5.164s **Services:** 3 **Depth:** 3 **Total Spans:** 3

currency-conversion-service x2 | currency-exchange-service x1 | netflix-zuul-api-gateway-server x2

```
Services                          1.033s            2.066s            3.098s            4.131s            5.164s
- currency-conversion-service 5.164s : http:/currency-converter-feign/from/eur/to/inr/qua
 - netflix-zuul-api-gateway-server                   3.108s : http:/currency-exchange-service/currency-exchange/
   currency-exchange-service                                           1.907s : http:/currency-exchange-service/currency-exchange/
```

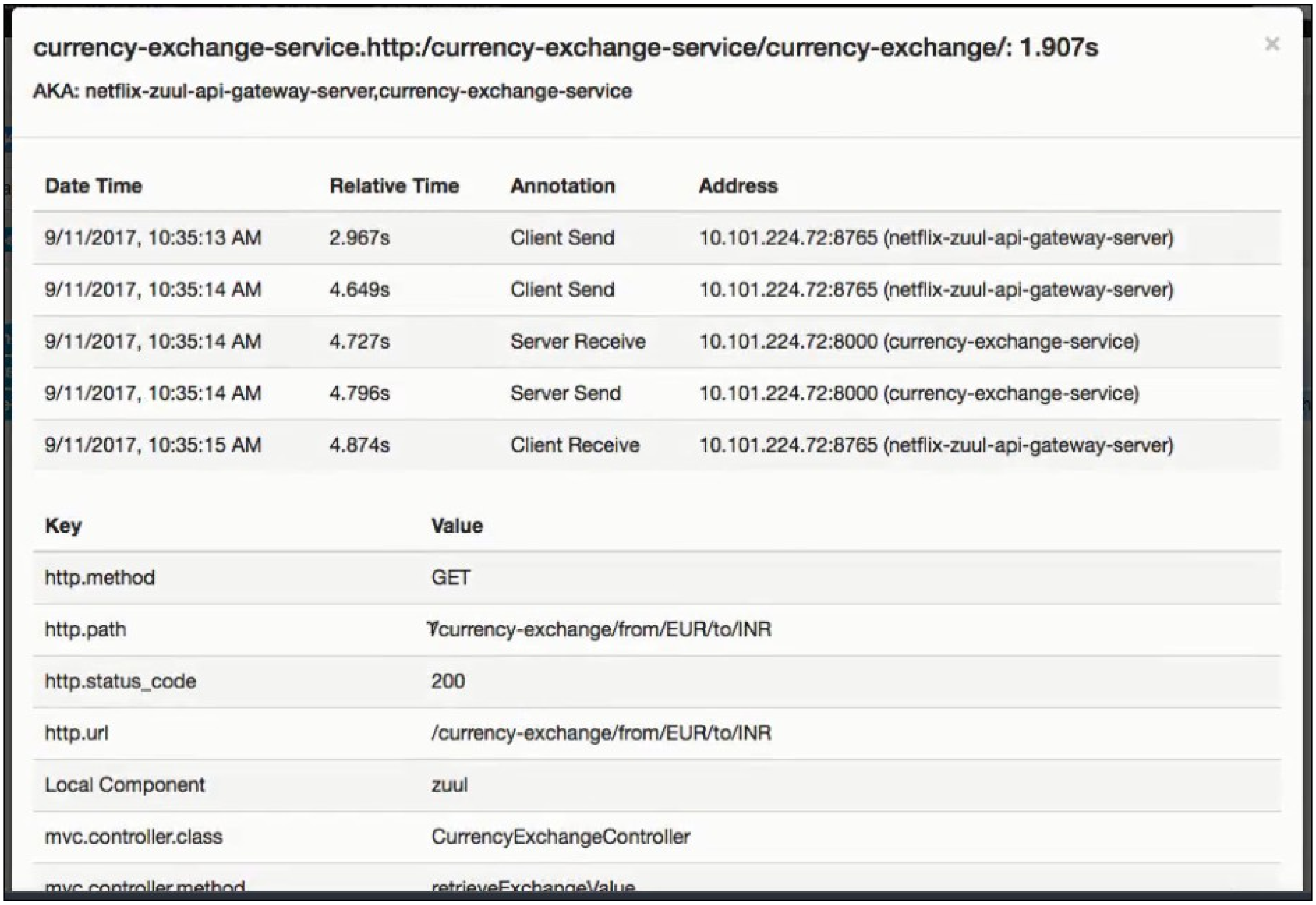
Click on the CES to see the details of this request inside the CES:


localhost:8100/currency-con
Zipkin - Traces
localhost:8000/currency-exch
Eureka
Spring Microservices - in28Mi
localhost:9411/zipkin/traces/054cfcc47870b2bb
Zipkin Investigate system behavior
Find a trace
Dependencies
Go to trace
JSON
Expand All
Collapse All
Filter Service Search
Services
1.033s
2.066s
3.098s
4.131s
5.164s

Duration: 5.164s Services: 3 Depth: 3 Total Spans: 3

currency-conversion-service x2 currency-exchange-service x1 netflix-zuul-api-gateway-server x2

- currency-conversion-service 5.164s : http:/currency-converter-feign/from/eur/to/inr/qua
- netflix-zuul-api-gateway-server 3.108s : http:/currency-exchange-service/currency-exchange/
currency-exchange-service 1.907s : http:/currency-exchange-service/currency-exchange/



currency-exchange-service.http:/currency-exchange-service/currency-exchange/: 1.907s

AKA: netflix-zuul-api-gateway-server,currency-exchange-service

Date Time	Relative Time	Annotation	Address
9/11/2017, 10:35:13 AM	2.967s	Client Send	10.101.224.72:8765 (netflix-zuul-api-gateway-server)
9/11/2017, 10:35:14 AM	4.649s	Client Send	10.101.224.72:8765 (netflix-zuul-api-gateway-server)
9/11/2017, 10:35:14 AM	4.727s	Server Receive	10.101.224.72:8000 (currency-exchange-service)
9/11/2017, 10:35:14 AM	4.796s	Server Send	10.101.224.72:8000 (currency-exchange-service)
9/11/2017, 10:35:15 AM	4.874s	Client Receive	10.101.224.72:8765 (netflix-zuul-api-gateway-server)

Key	Value
http.method	GET
http.path	/currency-exchange/from/EUR/to/INR
http.status_code	200
http.url	/currency-exchange/from/EUR/to/INR
Local Component	zuul
mvc.controller.class	CurrencyExchangeController
mvc.controller.method	retrieveExchangeValue