Lab 4.1

In the file arc/main/resources/application.yaml replace the <TODO> to Define a route that maps any url ending in /mtn/* to a eureka service “service-one”

```
zuul:
  routes:
    echo:
      path: /mtn/**
      serviceId: service-one
```

Then define what service-one actually maps to in the RibbonServerList of this Zuul Service, by replacing the last <TODO> in the Ribbon config part of this file with;

```
service-one:
  ribbon:
    DeploymentContextBasedVipAddresses: service01
    NIWSServerListClassName: com.netflix.niws.loadbalancer.DiscoveryEnabledNIWSServerList
```

Note the name of the service, “edge” in the yaml

Lab 4.1

Launch your Zuul service (this project)

You have FIVE services running now; FOUR registered with Eureka

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
EDGE	n/a (1)	(1)	UP (1) - host.docker.internal:edge:8080
SERVICE01	n/a (1)	(1)	UP (1) - host.docker.internal:service01:8082
SERVICE02	n/a (2)	(2)	UP (2) - host.docker.internal:service02:8086 , host.docker.internal:service02:8085

Lab 4.1

Click on the link for the “edge” service

In the browser navigator, change the url from http://hostname:8080/info to http://hostname:8080/mtn/1 Do not worry if you get a timeout error, just refresh the browser. You will be routed to Service01 courtesy of your yaml configuration in the edge service. The filter in the Zuul project is logging the request as well.

```
// http://192.168.1.141:8080/mtn/1

{
  "id": 1,
  "name": "Annapurna I",
  "range": "Annapurna Himalaya",
  "country": "Nepal",
  "firstAscent": "1950"
}
```

```
INFO 18640 --- [nio-8080-exec-9]
com.student.filters.SimpleFilter         : GET request to
http://192.168.1.141:8080/mtn/1
INFO 18640 --- [nio-8080-exec-9]
com.student.filters.SimpleFilter         : PRE
```