**Spring Cloud Bus & Hysterix**

If there are multiple instances of **limits service** running ,then changing a value in the active profile of **Config server** ( in git) , we would need to refresh the actuator URL of each instance the limit service.That is painful.

<http://localhost:8080/refresh> ( Spring 1.x+)

[http://localhost:8080/actuator/refresh](http://localhost:8080/refresh) ( Spring 2.0.0+)

[http://localhost:8081/actuator/refresh](http://localhost:8080/refresh)

**Solution :** Implement Spring-Bus

**Pre requisite:** Start Rabbit MQ ( the AMPQ protocol )

**Step1**:Add Dependency in pom.xml of limit-services & Config-services

**<dependency>**

**<groupId>org.springframework.cloud</groupId>**

**<artifactId>spring-cloud-starter-bus-amqp</artifactId>**

**</dependency>**

**Step2**:Disable security in limit-service bootstrap.properties

**management.security.enabled=false**

**step3:** Update the values in the **limit-service-qa.properties** ( Config-service) & commit in git.

**Step4:**  Refresh the actuator URLof one instance , it will refresh every instance.

<http://localhost:8080/refresh> ( Spring 1.x+)

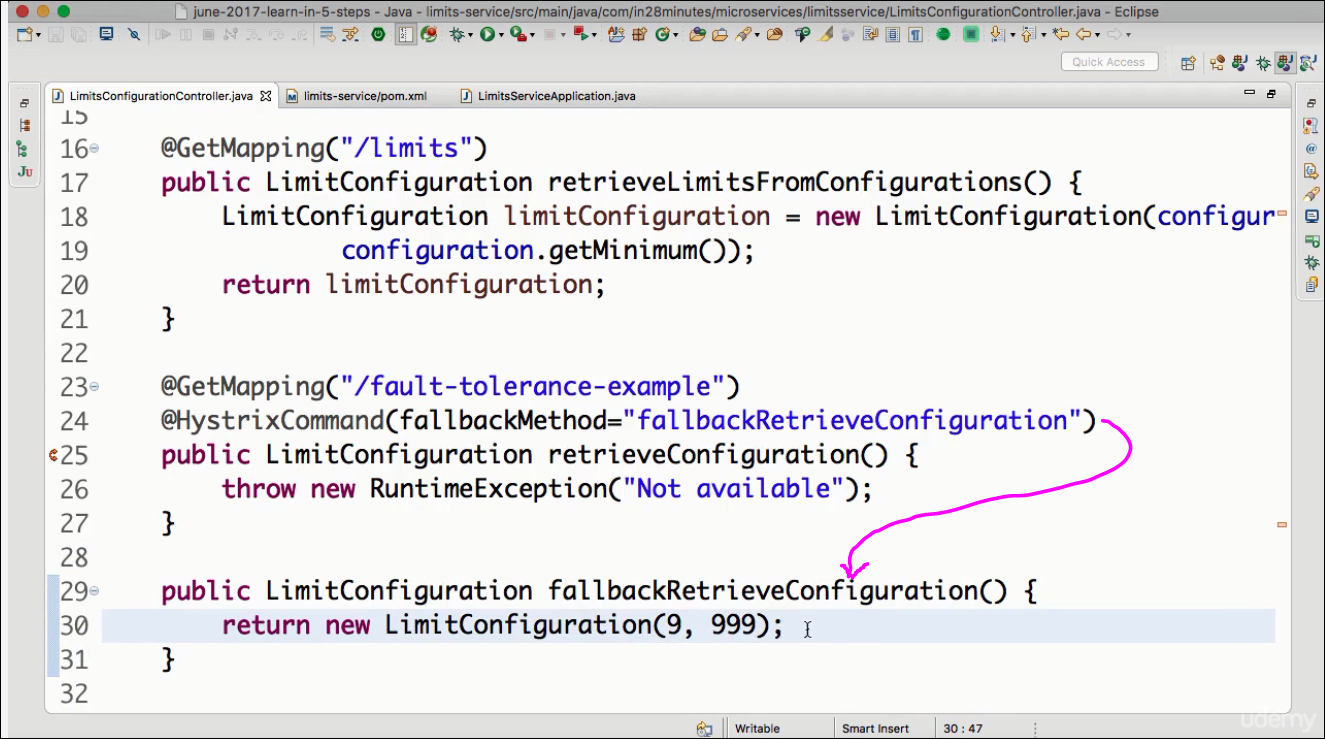
[http://localhost:8080/actuator/refresh](http://localhost:8080/refresh) ( Spring 2.0.0+)

**How it works:** As soon as ur application starts all instances are registered with the spring-cloud-starter-bus , when an actuator refresh request is called the change it finds is propagated to all the instance of the service on which refresh is called.

**Fault Tollerance with Hystrix**

**How to handle scenario when a dependent microservice is not available?**

**Solution:** Use Hysterix to call a fallback mechanism when they are not available

****