

TP 6 - Circuit Breaker avec Hystrix

- Spring BOOT
- Spring Cloud Hystrix

Estimated Time :

- **30 minutes**

6.0 - Objectifs

Le présent TP se veut présenter comment mettre en place les technologies à partir de Spring cloud netflix Hystrix qui constituent les briques de bases d'un circuit breaker :

- Spring Cloud Hystrix
- Spring Boot Starter

Pour Chacun des aspects nous montrerons comment :

- Tester
- Mettre en place les bonnes pratiques

6.1 - Mise en place dans maven

###TODO 01

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-hystrix</artifactId>
</dependency>
```

6.2 - Configuration dans l'application

Dans `resanet-catalogue` pour la classe `com.zenika.microservices.resanet.catalog.services.TransportServiceImpl`, on a un service qui permet d'écrire les nouveaux transports dans ES.

```
public String ajouterTransport(Transport transport) {  
    Transport t = transportRepository.save(transport);  
    return t.getId();  
}
```

###TODO 02

Ajouter un boolean pour déclencher une exception si activé.

###TODO 03

Ajouter le support JMX

Dans Application.java > @EnableMBeanExport

et dans TransportServiceImpl ajouter @ManagedResource et @ManagedAttribute

TODO 04

Pour activer les circuits breaker dans resanet-catalogue, ajouter sur le starter

```
@EnableEurekaClient
```

Utiliser **@HystrixCommand**

```
@HystrixCommand(fallbackMethod = "alternateTransportAdd")  
public String ajouterTransport(Transport transport)
```

TODO 05

Implémenter la méthode alternateTransportAdd avec la même signature que ajouterTransport

TODO 06

Démarrer l'application et jouer avec JConsole ou JVisualVM pour changer la valeur qui va activer sur exception le circuit breaker.