

Un poco de Java y +

Otra forma de hablar de nuestro día a día...

JAVA, SPRING, TESTING, TUTORIALES

¿Qué es Chaos Monkey para Spring Boot?

Fecha: 24 enero 2020 Autor/a: LuisMi Gracia ☐ 0 Comentarios

Chaos Monkey for Spring Boot

inspired by Chaos Engineering at Netflix



Chaos Monkey for Spring Boot es una herramienta que introduce los principios de **Chaos Engineering** en nuestras aplicaciones Spring Boot, en esencia añadiendo latencias a nuestro endpoints REST, lanzando errores, matando apps,...para ver cómo se comporta nuestro sistema completo...como podéis ver algo muy importante en la era de los microservicios 😊

Como vamos a ver, usarlo en una aplicación Spring Boot es muy sencillo

Cómo configurarlo

Primero añadimos la dependencia:

```
<dependency>
```

```
<groupId>de.codecentric</groupId>
```

```
<artifactId>chaos-monkey-spring-boot</artifactId>
```

```
<version>2.0.0</version>
```

```
</dependency>
```

Luego habilitamos el profile **chaos-monkey** (o sobre mi profile pongo `chaos.monkey.enabled=true`)

```
java -jar myapp.jar --spring.profiles.active=chaos-monkey
```

Chaos Monkey tiene 2 conceptos: Watchers y Assaults.

Watchers:

Un Watcher es un componente Spring Boot que vía AOP comprueba los métodos públicos de los Beans de tipo Componente, Controller, RestController, Service y Repository y en función de la configuración de la librería me permite:

- **Latency Assault**: añade una latencia aleatoria a la petición
- **Exception Assault** lanza RuntimeExceptions aleatorias
- **AppKiller Assault** –mata la app

Assaults:

Los Assaults son escenarios en los que queremos testar mi aplicación sobre alguno de los escenarios de test de arriba, así que en función de lo que quiero probar:

Se configuran así:

chaos:

monkey:

assaults:

level: 8

latencyRangeStart: 1000

latencyRangeEnd: 10000

exceptionsActive: true

killApplicationActive: true

watcher:

repository: true

restController: true

Y para que funcionen con Spring Boot Actuator:

management:

endpoint:

chaosmonkey:

enabled: true

endpoints:

web:

exposure:

include: health,info,chaosmonkey

¿Entonces qué? ¿Lo probamos? 😊

Leer más

© 2020 UN POCO DE JAVA Y +

