

# Zipkin

## SpringBoot Microservices Monitöring



Microservices mimarisinin karmaşıklığı arttıkça, servis ve sunucu örnekleri arasında tekdüze gözlemlenebilirlik kritik hale gelir. Her servis isteği birden çok hizmet katmanından geçer, gecikme veya çalışma zamanı sorunlarını tanımlamak geleneksel hata ayıklama teknikleri kullanılarak giderek zorlaşır. Dağıtılmış izleme, birden çok hizmet üzerinden geçiş yapan isteklerin bütüncül bir görünümünü sağlayarak sorunların anında tanımlanmasını sağlar.

Zipkin dağıtılmış bir izleme sistemidir. Microservice mimarilerindeki gecikme sorunlarını gidermek için gereken zamanlama verilerini toplamaya yardımcı olur. Bu verilerin toplanmasını ve aranmasını içerir. Zipkin'in tasarımı Google Dapper çalışmasına dayanmaktadır.

Zipkin Kurulumu:

```
wget -O zipkin.jar 'https://search.maven.org/remote_content?g=io.zipkin.java&a=zipkin-server&v=LATEST&c=exec'
```

- **Zipkin source** – <https://github.com/openzipkin/zipkin>
- **Docker** – <https://github.com/openzipkin/docker-zipkin>

**Zipkin kullanıcı arayüzü şu özellikleri sunar:**

- Servis adına göre arama yapma
- Servisler arasında bağımlılığı izleme grafiği
- Bir Servise harcanan zamanın yüzdesi ve işlemlerin başarısız olup olmadığı verileri izleyebilirsiniz.

## SpringBoot Entegre:

### Pom.xml

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

### Örnek Kod:

```
@SpringBootApplication
@RestController
@CrossOrigin
public class UserGreetingService {

    @RequestMapping("/api/user/greet")
    public String greet() {
        String greetingMsg = "Hello";
        String userName = restTemplate().getForObject("http://localhost:3001/api/user/name", String.class);
        String userAddress = restTemplate().getForObject("http://localhost:3002/api/user/address", String.class);

        return greetingMsg + " " + userName + "!\n\n" + userAddress;
    }

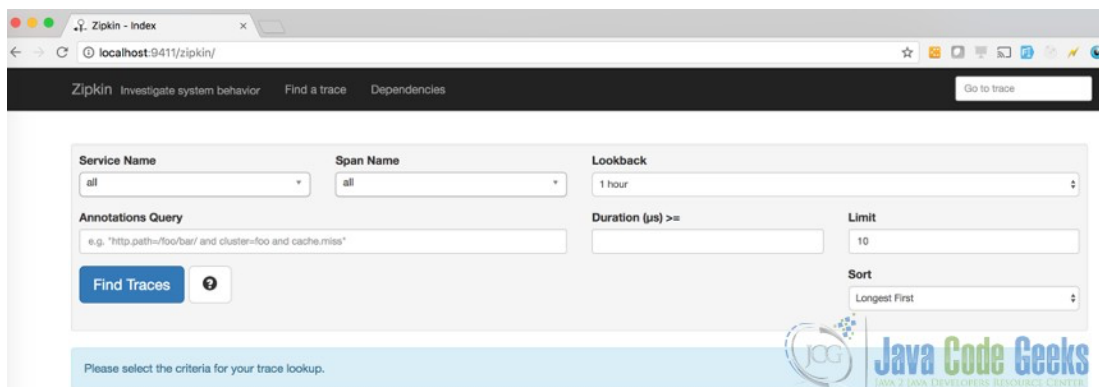
    @Bean
    RestTemplate restTemplate() {
        return new RestTemplate();
    }

    public static void main(String[] args) {
        SpringApplication.run(UserGreetingService.class,
            "--spring.application.name=user-greeting-service",
            "--server.port=3000"
        );
    }
}
```

### Zipkin Sunucusunu Başlatma:

Url: <http://localhost:9411/zipkin/>

### Zipkin kullanıcı arayüzü



Servis adı bölümünde seçim yapıldığında aşağıdaki gibi servis ayrıntısı gelecektir.

Service Name  
user-greeting-service

Span Name  
all

Lookback  
1 hour

Annotations Query  
e.g. "http.path=/foo/bar/ and cluster=foo and cache.miss"

Duration (µs) >=

Limit  
10

Find Traces ?

Sort  
Longest First

Showing: 1 of 1  
Services: user-greeting-service

JSON

3.603s 3 spans

user-greeting-service 100%

user-address-service x1 224ms user-greeting-service x3 3603ms user-name-service x1 1224ms



Örn:: 3 span seçeneği tıklanıldığında aşağıdaki gibi hizmetlerin yanıt süresi ve ayrıntıları görülebilmektedir.

Zipkin Investigate system behavior Find a trace Dependencies

Duration: 3.603s Services: 3 Depth: 2 Total Spans: 3

Expand All Collapse All Filter Service S...

user-address-service x1 user-greeting-service x3 user-name-service x1

Services		720.573ms	1.441s	2.162s	2.882s
- user-greeting-service	3.603s : http://api/user/greet	-	-	-	-
user-name-service	1.224s : http://api/user/name	-	-	-	-
user-address-service	-	-	2.324s : http://api/user/address	-	-

Ayrıca, her hizmet aralığına tıklayarak, her hizmet isteği için izleme kimlikleri, span kimlikleri, üst izleme kimlikleri, sunucu yönetim ortamı bilgileri ve diğer izleme ayrıntılarının ayrıntılarını alabiliriz. Bu, hatalı mikro hizmet veya gecikme sorunlarını gidermemize kolayca yardımcı olabilir.

Zipkin Investigate system behavior

Duration: 3.603s

Expand All Collapse All

user-address-service x1 user-greeting-service x3 user-name-service x1

Services

- user-greeting-service 3.603s
- user-name-service 1.224s
- user-address-service

user-name-service.http://api/user/name: 1.224s

AKA: user-greeting-service,user-name-service

Date Time	Relative Time	Annotation	Address
3/1/2018, 10:20:03 AM	28.000ms	Client Send	10.0.0.90:3000 (user-greeting-service)
3/1/2018, 10:20:04 AM	208.000ms	Server Receive	10.0.0.90:3001 (user-name-service)
3/1/2018, 10:20:05 AM	1.252s	Client Receive	10.0.0.90:3000 (user-greeting-service)
3/1/2018, 10:20:05 AM	1.252s	Server Send	10.0.0.90:3001 (user-name-service)

Key	Value
http.host	localhost
http.method	GET
http.path	/api/user/name
http.url	http://localhost:3001/api/user/name
mvc.controller.class	UserNameService
mvc.controller.method	getUserName
spring.instance_id	10.0.0.90:user-greeting-service:3000
spring.instance_id	10.0.0.90:user-name-service:3001

More Info

traceId	5a981a23f90863648501db99c3330f40
spanId	09d049cc9e687b61
parentId	8501db99c3330f40

