链滴

Eureka 添加安全验证的两种方式其一：使用 Spring Security(2022 最新版本)

作者：MingGH

原文链接：https://ld246.com/article/1672374544516

来源网站：链滴

原文发布于：Eureka添加安全验证的两种方式其一：使用Spring Security(2022最新版本)，欢迎使用SS订阅获取最新更新。

1. 使用Spring Security

已有做好的demo，大家可以自取,链接：https://github.com/MingGH/demo-eureka-server-auth

这里代码使用的依赖版本如下

spring-boot-starter-parent	3.0.1
spring-cloud.version	2022.0.0
java.version	17

开始我们的步骤

1.1 pom.xml中添加spring security的依赖

```
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

1.2 yaml配置中添加登录和接口请求需要的账号密码

```
server:
  port: 5005

spring:
  application:
    name: demo-eureka-server-auth
  security:
    user:
      name: develop # 账号
      password: develop # 密码
eureka:
  instance:
    hostname: localhost
    appname: ${spring.application.name}
  server:
    enable-self-preservation: true
    eviction-interval-timer-in-ms: 4000
  client:
    registerWithEureka: true # 这里我设置为true是因为把当前项目也注册到注册中心，就省了新建个client项目
    fetchRegistry: false
    service-url:
      defaultZone: http://${spring.security.user.name}:${spring.security.user.password}@${eureka.instance.hostname}:${server.port}/eureka #这里是有更改的，对应的内容是：http://develop:develop@localhost:5005/eureka
  environment: dev
```

在上述配置中需要注意的几个配置，

- spring.security.user账号密码
- eureka.client.registerWithEureka=true 这里我设置为true是因为把当前项目也注册到注册中心，省了新建一个client项目
- eureka.client.service-url.defaultZone=http://${spring.security.user.name}:${spring.security.user.password}@${eureka.instance.hostname}:${server.port}/eureka 对 defaultZone [进行特殊配是因为 spring.security](http://进行特殊配置是因为spring.security) 需要账号密码才能授权请求到应的接口

1.3 Spring Security关闭csrf

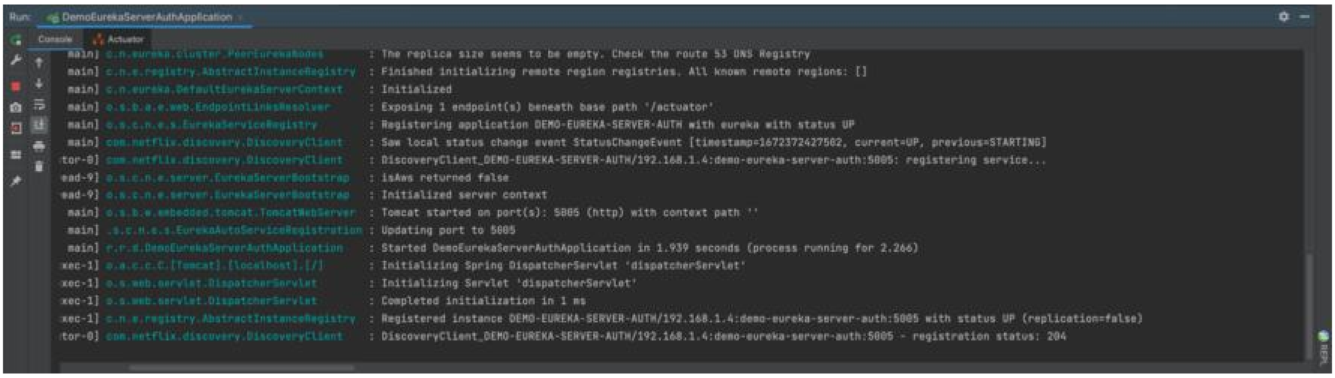
新建一个WebSecurityConfig, 注入spring容器中

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class WebSecurityConfig {

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http.csrf().disable()
                .authorizeRequests()
                .anyRequest()
                .authenticated()
                .and()
                .httpBasic();
        return http.build();
    }
}
```

最后别忘了在启动类上加上注解：@EnableEurekaServer

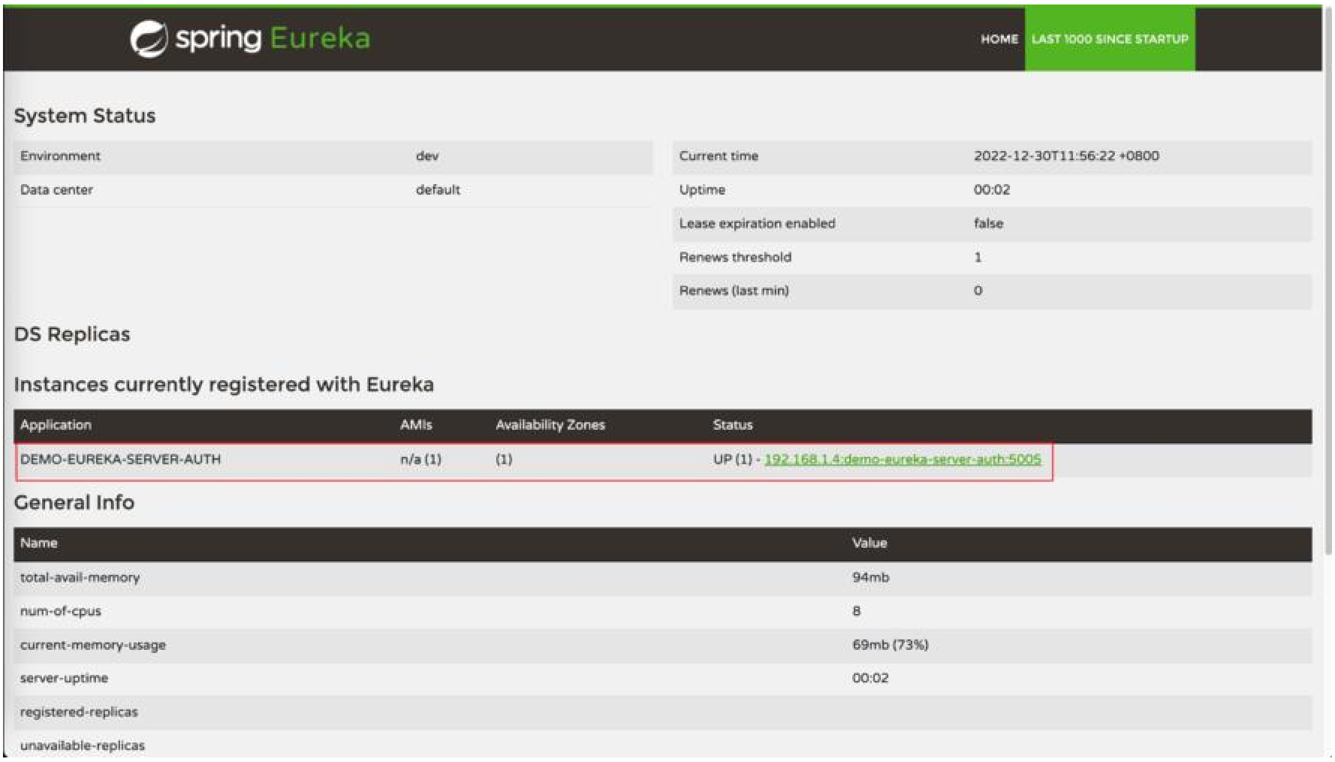
1.4 测试效果

启动正常



```
  main] c.n.eureka.cluster.PeerEurekaNodes        : The replica size seems to be empty. Check the route 53 DNS Registry
  main] c.n.e.registry.AbstractInstanceRegistry   : Finished initializing remote region registries. All known remote regions: []
  main] c.n.eureka.DefaultEurekaServerContext     : Initialized
  main] o.s.b.a.e.web.EndpointLinksResolver       : Exposing 1 endpoint(s) beneath base path '/actuator'
  main] o.s.c.n.e.s.EurekaServiceRegistry         : Registering application DEMO-EUREKA-SERVER-AUTH with eureka with status UP
  main] com.netflix.discovery.DiscoveryClient     : Saw local status change event StatusChangeEvent [timestamp=1672372427502, current=UP, previous=STARTING]
tor-0] com.netflix.discovery.DiscoveryClient     : DiscoveryClient_DEMO-EUREKA-SERVER-AUTH/192.168.1.4:demo-eureka-server-auth:5005: registering service...
ead-9] o.s.c.n.e.server.EurekaServerBootstrap    : isAws returned false
ead-9] o.s.c.n.e.server.EurekaServerBootstrap    : Initialized server context
  main] o.s.b.w.embedded.tomcat.TomcatWebServer   : Tomcat started on port(s): 5005 (http) with context path ''
  main] .s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 5005
  main] r.r.d.DemoEurekaServerAuthApplication     : Started DemoEurekaServerAuthApplication in 1.939 seconds (process running for 2.266)
xec-1] o.a.c.c.C.[Tomcat].[localhost].[/]        : Initializing Spring DispatcherServlet 'dispatcherServlet'
xec-1] o.s.web.servlet.DispatcherServlet         : Initializing Servlet 'dispatcherServlet'
xec-1] o.s.web.servlet.DispatcherServlet         : Completed initialization in 1 ms
xec-1] c.n.e.registry.AbstractInstanceRegistry   : Registered instance DEMO-EUREKA-SERVER-AUTH/192.168.1.4:demo-eureka-server-auth:5005 with status UP (replication=false)
tor-0] com.netflix.discovery.DiscoveryClient     : DiscoveryClient_DEMO-EUREKA-SERVER-AUTH/192.168.1.4:demo-eureka-server-auth:5005 - registration status: 204
```

从面板上看也是正常的，能够成功给自己注册



System Status

Environment	dev
Data center	default

Current time	2022-12-30T11:56:22 +0800
Uptime	00:02
Lease expiration enabled	false
Renews threshold	1
Renews (last min)	0

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DEMO-EUREKA-SERVER-AUTH	n/a (1)	(1)	UP (1) - 192.168.1.4:demo-eureka-server-auth:5005

General Info

Name	Value
total-avail-memory	94mb
num-of-cpus	8
current-memory-usage	69mb (73%)
server-uptime	00:02
registered-replicas	
unavailable-replicas	

至此大功告成

1.5 参考以下内容

【security】spring security放行不生效,security放行后还是被拦截，路径变成了/error

Spring Security without the WebSecurityConfigurerAdapter

原文链接：Eureka 添加安全验证的两种方式其一：使用 Spring Security(2022 最新版本)