



链滴

关于 SpringBoot 下 Shiro 的拦截器配置中 anon 不起作用的问题

作者: [tuchief](#)

原文链接: <https://ld246.com/article/1584410376875>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

我在配置文件application.yml中的配置和shiroFilter中动态拦截器注入的代码如下图所示，启动后访问login/auth 登录请求接口时，依然是被Shiro拦截的；

```
#动态设置拦截器
permission-config:
  perms:
    - key: /
      value: anon
    - key: /login/auth
      value: anon
    - key: /login/logout
      value: logout
    - key: /static/**
      value: anon
    - key: /error
      value: anon
    - key: /swagger-ui.html
      value: anon
    - key: /swagger-resources/**
      value: anon
    - key: /v2/api-docs/**
      value: anon
    - key: /webjars/springfox-swagger-ui/**
      value: anon
    - key: /druid/**
      value: anon
    - key: /app/**
      value: anon
    - key: /api/spider/**
      value: anon
    - key: /**
      value: jwt
```

```
@Bean
public ShiroFilterFactoryBean shiroFilter(DefaultWebSecurityManager securityManager, JedisUtils jedisUtils,
    ShiroFilterFactoryBean shiroFilter = new ShiroFilterFactoryBean();
    shiroFilter.setSecurityManager(securityManager);

    // 添加jwt过滤器
    Map<String, Filter> filterMap = new HashMap<>();
    filterMap.put("jwt", new JwtFilter(jwtProp, syncCacheService, jedisUtils));
    filterMap.put("logout", new SystemLogoutFilter(jedisUtils));
    shiroFilter.setFilters(filterMap);

    //动态配置拦截器注入，切记这里的filterRuleMap应该为LinkedHashMap，而不是HashMap
    Map<String, String> filterRuleMap = new LinkedHashMap<>(initialCapacity: 16);
    List<Map<String, String>> perms = this.getShiroFilterProperties().getPerms();
    perms.forEach(perm -> filterRuleMap.put(perm.get("key"), perm.get("value")));

    shiroFilter.setFilterChainDefinitionMap(filterRuleMap);

    return shiroFilter;
```

其实问题就出在shiroFilter的动态拦截注入的代码中；因为shiro的拦截是从上往下依次匹配的，所以

配置中，需要将

/**放在最后，然后依次设置需要放行或拦截的资源。所以既然配置文件中的资源已经是正确的配置了为啥启动后访问还是被拦截了呢？这是因为HashMap是无序的，在启动时，配置文件被加载了，存到了 filterRuleMap中，但是它已经变成无序的了；解决方法就是让 filterRuleMap变的有序，修改的代码如下：

```
@Bean
public ShiroFilterFactoryBean shiroFilter(DefaultWebSecurityManager securityManager, JedisUtils jedisUtils,
    ShiroFilterFactoryBean shiroFilter = new ShiroFilterFactoryBean();
    shiroFilter.setSecurityManager(securityManager);

    // 添加jwt过滤器
    Map<String, Filter> filterMap = new HashMap<>();
    filterMap.put("jwt", new JwtFilter(jwtProp, syncCacheService, jedisUtils));
    filterMap.put("logout", new SystemLogoutFilter(jedisUtils));
    shiroFilter.setFilters(filterMap);

    //动态配置拦截器注入，切记这里的filterRuleMap应该为LinkedHashMap，而不是HashMap
    Map<String, String> filterRuleMap = new LinkedHashMap<>(initialCapacity: 16);
    List<Map<String, String>> perms = this.getShiroFilterProperties().getPerms();
    perms.forEach(perm -> filterRuleMap.put(perm.get("key"), perm.get("value")));

    shiroFilter.setFilterChainDefinitionMap(filterRuleMap);

    return shiroFilter;
```

好了，重新启动你的服务，然后访问 /login/auth 接口，现在已经按照你的设定完成了操作；