# FIlter里面怎么注入Service

原创 老程序员刘飞 于 2019-11-18 20:22:23 发布 阅读量2k 收藏 2 点赞数 

我们知道Filter是无法注入Service的，这是由于顺序是监听器—》过滤器-----》 servlet
我们的service实体化后是交给spring管理的所以无法所以过滤器这个时候初始化，Service还没有初始化

这里我是要把一个redisTemplte注入到shiroFIlter里面去
记录下代码
public class ShiroLoginFilter extends UserFilter{

```
private String sessionId;
private RedisTemplate<String, ?> redisTemplate;

@Override
protected boolean isAccessAllowed(ServletRequest request, ServletResponse response, Object mappedValue) {
        // TODO Auto-generated method stub
        if (redisTemplate == null) {
            redisTemplate = (RedisTemplate) ShiroSpringUtils.getBean("redisTemplate");
        }
        Subject subject = getSubject(request, response);
        boolean authenticated = subject.isAuthenticated();
        if (subject.isAuthenticated()) {
                User user =(User) (subject.getPrincipal());
                String userID = user.getUserID();
                sessionId = redisTemplate.opsForValue().get(userID).toString();
                if("0".equals(sessionId)) {
                        return false;
                }
}
        return super.isAccessAllowed(request, response, mappedValue);
}

@Override
protected boolean onAccessDenied(ServletRequest request,ServletResponse response) throws Exception {
        HttpServletRequest req= (HttpServletRequest) request;
    HttpServletResponse httpServletResponse = (HttpServletResponse) response;
    if("0".equals(sessionId)) {
        Subject subject = SecurityUtils.getSubject();
        subject.logout();
        //httpServletResponse.sendRedirect("/login.html");
    }
    if(isAjax(req)){//如果是ajax请求返回状态码
         httpServletResponse.setCharacterEncoding("UTF-8");
         httpServletResponse.setContentType("application/json");
         httpServletResponse.getWriter().write(JSONObject.toJSON(ApiResponse.ofMsg(403,"您还未登录，重新登录")).toStrin
         httpServletResponse.setStatus(HttpServletResponse.SC_FORBIDDEN);
    }else{
        /**
         * @Mark 非ajax请求重定向为登录页面
         */
         httpServletResponse.sendRedirect("/login.html");
    }
    return false;
}

public boolean isAjax(HttpServletRequest request){
    String header = request.getHeader("X-Requested-With");
    boolean isAjax = "XMLHttpRequest".equalsIgnoreCase(header) ? true:false;
    return isAjax;
}
```

}
这里我们来看下这个类

```
public class ShiroLoginFilter extends UserFilter{

    private String sessionId;
    private RedisTemplate<String, ?> redisTemplate;

    @Override
    protected boolean isAccessAllowed(ServletRequest request, ServletResponse response, Obje
        // TODO Auto-generated method stub
        if (redisTemplate == null) {
            redisTemplate = (RedisTemplate) ShiroSpringUtils.getBean("redisTemplate");
        }
        Subject subject = getSubject(request, response);
        boolean authenticated = subject.isAuthenticated();
        if (subject.isAuthenticated()) {
```



package com.crsri.config;

import org.springframework.beans.BeansException;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ApplicationContextAware;
import org.springframework.stereotype.Component;
/**
*

- 做一个fiter可以注入service的工具类
  */
  @Component
  public class ShiroSpringUtils implements ApplicationContextAware

  {private static ApplicationContext applicationContext;

  @Override
  public void setApplicationContext(ApplicationContext applicationContext)
  throws BeansException {
  if (ShiroSpringUtils.applicationContext == null) {
  ShiroSpringUtils.applicationContext = applicationContext;
  }
  }

  public static ApplicationContext getApplicationContext() {
  return applicationContext;
  }

  //根据name
  public static Object getBean(String name) {
  return getApplicationContext().getBean(name);
  }

  //根据类型
  public static T getBean(Class clazz) {
  return getApplicationContext().getBean(clazz);
  }

  public static T getBean(String name, Class clazz) {
  return getApplicationContext().getBean(name, clazz);
  }
  }

```java
@Component
public class ShiroSpringUtils implements ApplicationContextAware {

    private static ApplicationContext applicationContext;

    @Override
    public void setApplicationContext(ApplicationContext applicationContext)
            throws BeansException {
        if (ShiroSpringUtils.applicationContext == null) {
            ShiroSpringUtils.applicationContext = applicationContext;
        }
    }

    public static ApplicationContext getApplicationContext() {
        return applicationContext;
    }

    //根据name
    public static Object getBean(String name) {
        return getApplicationContext().getBean(name);
    }
```

实现这我们就可以继承它里面的set方法

然后用这个applicationContext来得到你的实体类就可以了