**过滤器Filter**

1. **注解方式：**

**@WebFilter**

**filterName** String 过滤器名称.等价于web.xml中<filter-name>

**servletNames**  String[] 指定过滤器将应用与那些Servlet，取值是@WebServlet中的name属性的取值，或web.xml中的<servlet-name>的值

**urlPatterns** String[] 过滤器的URL匹配模式.等价于等价于web.xml中<url-pattern>

**value**  String[]等价于urlPatterns，两者不可同时使用

**dispatcherTypes** DispatcherType 指定过滤器转发模式，值：ASYNC,ERROR,FORWARD,

INCLUDE,REQUEST

**initParams** WebInitParam[]指定一组过滤器初始化参数，等价于web.xml的<init-param>

**asyncSupported** Boolean声明过滤器是否支持异步操作模式，等价于web.xml的<async-supported>

**description**  String 描述信息等价于web.xml的<description>标签

**displayName** String 该过滤器显示的名字，配合工具使用，

等价于web.xml的<display-name>标签

import javax.servlet.\*;  
import javax.servlet.annotation.**WebFilter**;  
import java.io.IOException;**@WebFilter**(filterName = **"TestFilter"**,urlPatterns = **"/api/\*"**)  
public class TestFilter implements Filter {  
 **@Override** public void init(FilterConfig filterConfig) throws ServletException {  
 System.*err*.println(**"初始化...."**);  
 }  
 **@Override** public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {  
 System.*err*.println(**"TestFilter is running..."**);  
 chain.doFilter(request,response);  
 }  
 **@Override** public void destroy() {  
 System.*err*.println(**"TestFilter destroy..."**);  
 }  
}

**启动类要添加@ServletComponentScan**

**@SpringBootApplication**(scanBasePackages={**"com.first.study.\*"**})  
**@MapperScan**(**"com.first.study.\*\*.dao"**)  
**@ServletComponentScan**//Listerner 监听器，Filter过滤器，自定义servlet 启动扫描并注入  
public class Application {  
 //默认启动器  
 public static void main(String[] args) throws Exception {  
 SpringApplication.*run*(Application.class, args);  
 }}

1. 使用@Bean注入

TestFilter不需要@WebFilter

Application不需要@ServletComponentScan

import org.springframework.boot.web.servlet.FilterRegistrationBean;  
import org.springframework.context.annotation.**Bean**;  
import org.springframework.stereotype.**Component**;  
import java.util.ArrayList;  
import java.util.List;  
**@Component**public class TestFilterConfig {  
 **@Bean** public FilterRegistrationBean testFilterRegistration(){  
 FilterRegistrationBean filterRegistrationBean=new FilterRegistrationBean();  
 TestFilter testFilter=new TestFilter();  
 filterRegistrationBean.setFilter(testFilter);  
 List<String> urlPatterns=new ArrayList<>();  
 urlPatterns.add(**"/api/\*"**);//过滤带特定字符的url  
 filterRegistrationBean.setUrlPatterns(urlPatterns);  
 return filterRegistrationBean;  
 }  
}

**3注解方式，推荐使用此方式简单**

**Application不需要@SevletComponentScan**

import org.springframework.stereotype.**Component**;  
import javax.servlet.\*;  
import javax.servlet.annotation.**WebFilter**;  
import java.io.IOException;  
**@WebFilter**(filterName = **"TestFilter"**,urlPatterns = **"/api/\*"**)  
**@Component**//添加进IOC容器  
public class TestFilter implements Filter {  
 **@Override** public void init(FilterConfig filterConfig) throws ServletException {  
 System.*err*.println(**"初始化...."**);  
 }  
 **@Override** public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {  
 System.*err*.println(**"TestFilter is running..."**);  
 chain.doFilter(request,response);  
 }  
 **@Override** public void destroy() {  
 System.*err*.println(**"TestFilter destroy..."**);  
 }  
}