

Spring Boot ve Custom Servlet/Filter/Listener Tanımları



Servlet, Filter veya EventListener Tanımlamak

- Uygulamaya özel Servlet, Filter veya HTTP EventListener tanımlamak da mümkündür
- Yazılan Servlet, Filter veya EventListener sınıfları **@WebServlet**, **@WebFilter**, **@WebListener** anotasyonları ile işaretlenmelidir
- Ayrıca uygulamanın konfigürasyon sınıfında da **@ServletComponentScan** anotasyonu kullanılmalıdır

Servlet, Filter veya EventListener Tanımlamak

```
@WebServlet(urlPatterns={"/pcs"})  
public class PetClinicServlet  
    extends HttpServlet {  
  
    @Override  
    protected void doGet(  
        HttpServletRequest req,  
        HttpServletResponse resp)  
        throws ServletException,  
            IOException {  
        resp.getWriter()  
            .write("pcs...");  
    }  
}
```

```
@WebFilter(urlPatterns=/*")  
public class PetClinicFilter  
    implements Filter {  
  
    ...  
    @Override  
    public void doFilter(ServletRequest  
arg0, ServletResponse arg1, FilterChain  
arg2)  
        throws IOException,  
            ServletException {  
        try {  
            System.out.println("before...");  
            arg2.doFilter(arg0, arg1);  
        } finally {  
            System.out.println("after...");  
        }  
    }  
}
```

Servlet, Filter veya EventListener Tanımlamak

@WebListener

```
public class PetClinicHttpSessionListener implements HttpSessionListener {  
  
    @Override  
    public void sessionCreated(HttpSessionEvent se) {  
        System.out.println("Session created :" + se.getSession().getId());  
    }  
  
    @Override  
    public void sessionDestroyed(HttpSessionEvent se) {  
        System.out.println("Session destroyed :" +  
se.getSession().getId());  
    }  
  
}
```

@SpringBootApplication

@ServletComponentScan

```
public class PetClinicApplication {  
    ...  
}
```

Servlet, Filter veya EventListener Tanımlamak

- 3rd party sınıflarda @WebServlet, @WebFilter veya @EventListener anotasyonları **kullanılmamış** olabilir
- Bu gibi durumlarda konfigürasyon sınıfı içerisinde **ServletRegistrationBean, FilterRegistrationBean, veya ServletListenerRegistrationBean** sınıfları ile aynı şekilde servlet, filter veya listener tanımlanabilir

Servlet, Filter veya EventListener Tanımlamak

@Configuration

```
public class PetClinicConfiguration {  
    @Bean  
    public ServletRegistrationBean pcs() {  
        return new ServletRegistrationBean(  
            new PetClinicServlet(), "/pcs");  
    }  
  
    @Bean  
    public FilterRegistrationBean pcsFilter() {  
        FilterRegistrationBean bean = new FilterRegistrationBean(  
            new PetClinicFilter());  
        bean.setUrlPatterns(Arrays.asList("/*"));  
        bean.setOrder(Ordered.HIGHEST_PRECEDENCE);  
        return bean;  
    }  
  
    @Bean  
    public ServletListenerRegistrationBean pcsListener() {  
        return new ServletListenerRegistrationBean<>(  
            new PetClinicHttpSessionListener());  
    }  
}
```

İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

