# Spring Boot ve JSF

harezmi bilişim çözümleri

# Spring Boot ve JSF

- Spring Boot kullanan uygulamlarda **UI teknolojisi olarak JSF**'in de kullanılması mümkündür
- Gömülü sunucularda çalışırken, JSF **ManagedBean'ların Spring ApplicationContext tarafından yönetilmesi** en kolay yoldur
- Bu durumda **custom ViewScope** konfigürasyonuna ihtiyaç duyulacaktır
- Internet'de değişik kaynaklardan custom ViewScope gerçekleştirimi bulunabilir

# JSF Konfigürasyon Adımları

- JSF konfigürasyonundaki **temel iş adımları** şunlardır
  - JSF bağımlılıklarının eklenmesi
  - FacesServlet'in tanımlanması
  - ConfigureListener'ın tanımlanması
  - Konfigürasyonun executable war içerisinden yüklenmesinin sağlanması
  - Custom ViewScope konfigürasyonu
  - faces-config.xml konfigürasyonu
  - Proje artefact'ının war yapılması

# JSF Konfigürasyonu

- **JSF bağımlılıkları** classpath'e eklenmelidir

```
<dependency>
    <groupId>com.sun.faces</groupId>
    <artifactId>jsf-api</artifactId>
    <version>2.2.14</version>
</dependency>
<dependency>
    <groupId>com.sun.faces</groupId>
    <artifactId>jsf-impl</artifactId>
    <version>2.2.14</version>
</dependency>
```

- Ayrıca JSF Container JSP derleyicisine ihtiyaç duyduğu için **classpath'de JSP derleyicisi**ninde olması gerekir

```
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
</dependency>
```

# JSF Konfigürasyonu

```
@Configuration
public class PetClinicConfig implements ServletContextAware {
    @Bean
    public ServletRegistrationBean facesServletRegistration() {
        ServletRegistrationBean registration =
            new ServletRegistrationBean(new FacesServlet(), "*.xhtml");
        registration.setLoadOnStartup(1);
        return registration;
    }

    @Bean
    public ServletListenerRegistrationBean<ConfigureListener>
                                                    jsfConfigureListener() {
        return new ServletListenerRegistrationBean<ConfigureListener>(
                    new ConfigureListener());
    }

    ...
}
```

# JSF Konfigürasyonu

JSF konfigürasyonunun web container içerisinde yüklenebilmesi için gerekli bir context paramdır

```
@Configuration
public class PetClinicConfig implements ServletContextAware {
    @Override
    public void setServletContext(ServletContext servletContext) {
        servletContext.setInitParameter(
            "com.sun.faces.forceLoadConfiguration",
            Boolean.TRUE.toString());
    }

    @Bean
    public static CustomScopeConfigurer viewScope() {
        CustomScopeConfigurer configurer = new CustomScopeConfigurer();
        configurer.setScopes(
                Collections.singletonMap("view", new ViewScope()));
        return configurer;
    }

    ...
}
```

# JSF Konfigürasyonu

classpath:META-INF/resources/ dizini altında olmalıdır

```
<faces-config ...>
    <application>
        <el-resolver>
            org.springframework.web.jsf.el.SpringBeanFacesELResolver
        </el-resolver>

        <system-event-listener>
            <system-event-listener-class>
                com.javaegitimleri.petclinic.javaee.ViewScopeCallbackRegistrar
            </system-event-listener-class>
            <system-event-class>
                javax.faces.event.PostConstructViewMapEvent
            </system-event-class>
            <source-class>javax.faces.component.UIViewRoot</source-class>
        </system-event-listener>

        <system-event-listener>
            <system-event-listener-class>
                com.javaegitimleri.petclinic.javaee.ViewScopeCallbackRegistrar
            </system-event-listener-class>
            <system-event-class>
                javax.faces.event.PreDestroyViewMapEvent
            </system-event-class>
            <source-class>javax.faces.component.UIViewRoot</source-class>
        </system-event-listener>
    </application>
</faces-config>
```

# JSF Konfigürasyonu

- JSF sayfaları **sadece executable war** dosyaları içerisinden erişilebilmektedir
- Dolayısı ile pom.xml'deki packaging tipi **war** yapılmalıdır

```
<packaging>war</packaging>
```

- JSF sayfaları da **src/main/webapp** dizini altında bir lokasyona konulmalıdır

# İletişim

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