

# **JAVA SERVER FACES BEAN ANNOTATIONS LESSONS**

SADRZAJ – 10

JSF Anotacije (karakteristike kao napomene)

▶ **@ManagedBean anotacija**

```
@ManagedBean  
public class TestBean{  
    private String abc;  
    ...  
}
```

- ▶ navođenje: #{testBean.abc}, gdje naziv bean-a odgovara nazivu klase, s tim što je prvo slovo naziva malo slovo
- ▶ request scope je podrazumijevani scope
- ▶ abc je naziv property-ja (koji ima odgovarajuće getere i setere) ili naziv metode
- ▶ ako action controller metoda vraća string "xyz", i ako ne postoje eksplicitno definisana navigaciona pravila, onda je rezultujuća strana xyz.xhtml

▶ **name atribut @ManagedBean-a**

```
@ManagedBean(name="testBean2")  
public class TestBean{  
    private String abc;  
    ...  
}
```

- ▶ navođenje: #{testBean2.abc}, gdje je testBean2 vrijednost name atributa
- ▶ request scope je i dalje podrazumijevani scope

▶ **eager atribut @ManagedBean-a**

```
@ManagedBean(eager=true)
@ApplicationScoped
public class TestBean{
    ...
    ...
}
```

- ▶ ako je “eager=true” i scope je application, onda bean mora biti kreiran u trenutku učitavanja aplikacije, a ne u trenutku prvog referenciranja bean-a, tj. prije opsluživanja bilo kojeg zahtjeva

```
<managed-bean eager="true">
<managed-bean-name>cbbhBean</managed-bean-name>
<managed-bean-class>net.etfbl.s_cube.external.beans.CBBHBean</managed-bean-class>
<managed-bean-scope>application</managed-bean-scope>
</managed-bean>
```

▶ **scope**

▶ **@RequestScoped**

- podrazumijevani
- nova instanca se kreira pri svakom HTTP zahtjevu

▶ **@SessionScoped**

- scope sesije
- korisnik sa istim cookie-jem, ako sesija nije istekla, uvijek pristupa istom sesijskom bean-u
- bean bi trebao biti serijalizabilan

▶ **@ApplicationScoped**

- scope aplikacije
- dijele ga svi korisnici aplikacije
- ovaj bean ili nema stanje ili je potrebno izvršiti sinhronizaciju pristupa

- ▶ **scope**
- ▶ **@ViewScoped**
  - ista instanca bean-a se koristi dok god je korisnik na istoj stranici, npr. u slučaju AJAX zahtjeva
  - trebao bi biti serijalizabilan
- ▶ **@CustomScoped(value="#{someMap}")**
  - bean je smješten u Map objekat, i programer upravlja njegovim životnim vijekom
- ▶ **@NoneScoped**
  - bean nije smješten u scope
  - korisno u slučajevima kada bean treba biti referenciran od strane drugih bean-ova koji su u nekom od scope-ova

- ▶ **scope**
- ▶ **obično se smješta iza @ManagedBean**

```
@ManagedBean
@SessionScoped
public class TestBean {
    ...
}
```

- ▶ **@ManagedProperty**
- ▶ **JSF podržava dependency injection**
  - moguće je dodijeliti vrijednosti property-ju managed bean-a, bez hard kodovanja u definiciji klase
- ▶ **način I:**  
`@ManagedProperty(value="#{someBean}")`  
`private SomeType someField;`
- ▶ **način II:**
  - `<managed-property>` u `faces-config.xml` datoteci
  - isto kao i u verziji JSF 1.x
- ▶ **setter metoda za property je obavezna**  
`setSomeField`
- ▶ **ako ime setter metode ne odgovara imenu property-ja, moguće je koristiti "name" atribut @ManagedProperty-ja**