1 -Declarando managed bean

2. **package** bean;
4. **import** java.io.Serializable;
5. **import** javax.faces.bean.ManagedBean;
6. **import** javax.faces.bean.SessionScoped;
8. @ManagedBean
9. @SessionScoped
10. **public** **class** User **implements** Serializable{
12. **private** String username;
13. **private** String password;
15. **public** String getUsername() {
16. **return** username;
17. }
19. **public** **void** setUsername(String username) {
20. **this**.username = username;
21. }
23. **public** String getPassword() {
24. **return** password;
25. }
27. **public** **void** setPassword(String password) {
28. **this**.password = password;
29. }

32. }

* **Username e password são propriedades do bean;**
* **O escopo é do tipo sessionscoped. Escopos que podem ser associados ao bean:**

1. Sessionscoped : o bean será mantido enquanto a sessão http esta ativa
2. Requestscoped: o bean será mantido enquanto houver requisições http
3. Viewscoped: o bean será mantido enquanto ele estiver na mesma view(pagina)
4. Applicationscoped: o bean será mantido enquanto a aplicação esta online(o bean será acessível a todos)
5. Nonescoped: o bean não será mantido nem inicializado no escopo porem ele pode ser acessado por outro bean herdando assim o escopo do bean que o requisitou.

2 - Inicializando managed bean

2. **package** bean;
4. **import** java.io.Serializable;
5. **import** javax.annotation.PostConstruct;
6. **import** javax.annotation.PreDestroy;
7. **import** javax.faces.bean.ManagedBean;
8. **import** javax.faces.bean.ManagedProperty;
9. **import** javax.faces.bean.SessionScoped;
11. @ManagedBean
12. @SessionScoped
13. **public** **class** User **implements** Serializable{
15. @ManagedProperty(value = "teste")
16. **private** String username;
17. **private** String password;
19. @PostConstruct
20. **private** **void** init(){
21. System.out.println("iniciando o bean..usuario tem valor = "+username);
22. }
24. @PreDestroy
25. **private** **void** clean(){
26. System.out.println("destruindo o bean.");
27. }
29. **public** String getUsername() {
30. **return** username;
31. }
33. **public** **void** setUsername(String username) {
34. **this**.username = username;
35. }
37. **public** String getPassword() {
38. **return** password;
39. }
41. **public** **void** setPassword(String password) {
42. **this**.password = password;
43. }

46. }

* **@ManagedProperty inicializa uma propriedade do bean.**
* **@PostConstruct chama um método quando o bean é inicializado ou instanciado..**
* **@PreDestroy chama um método antes do bean ser destruído.**

3 - Gerenciando dependências de bean

JSF suporta inversão de controle(IoC) ou seja o managed bean pode ser acoplado em tempo de execução sem precisar manipular esse acoplamento no código da aplicação;

2. **package** bean;
4. **import** java.io.Serializable;
5. **import** javax.faces.bean.ManagedBean;
6. **import** javax.faces.bean.ManagedProperty;
7. **import** javax.faces.bean.NoneScoped;
8. @ManagedBean(name = "profession")
9. @NoneScoped
10. **public** **class** Profession **implements** Serializable{
12. @ManagedProperty(value = "Software Engineer")
13. **private** String title;
15. @ManagedProperty(value = "TI")
16. **private** String industry;
18. **public** String getTitle() {
19. **return** title;
20. }
22. **public** **void** setTitle(String title) {
23. **this**.title = title;
24. }
26. **public** String getIndustry() {
27. **return** industry;
28. }
30. **public** **void** setIndustry(String industry) {
31. **this**.industry = industry;
32. }

35. }
36. /\*################################################################################################\*/
37. **package** bean;
39. **import** java.io.Serializable;
40. **import** javax.faces.bean.ManagedBean;
41. **import** javax.faces.bean.ManagedProperty;
42. **import** javax.faces.bean.SessionScoped;
44. @ManagedBean
45. @SessionScoped
46. **public** **class** User **implements** Serializable{
48. @ManagedProperty("#{profession}")
49. **private** Profession profession;
51. **public** Profession getProfession() {
52. **return** profession;
53. }
55. **public** **void** setProfession(Profession profession) {
56. **this**.profession = profession;
57. }


61. }

Acessando managed beans através do código

2. **package** bean;
4. **import** java.io.Serializable;
5. **import** javax.el.ELContext;
6. **import** javax.el.ExpressionFactory;
7. **import** javax.el.ValueExpression;
8. **import** javax.faces.application.Application;
9. **import** javax.faces.bean.ManagedBean;
10. **import** javax.faces.bean.SessionScoped;
11. **import** javax.faces.context.FacesContext;
13. @ManagedBean
14. @SessionScoped
15. **public** **class** OutroBean **implements** Serializable{
17. **public** User getUser(){
19. FacesContext context = FacesContext.getCurrentInstance();
20. Application application = context.getApplication();
21. ELContext el = context.getELContext();
22. ExpressionFactory ef = application.getExpressionFactory();
23. ValueExpression value = ef.createValueExpression(el,"#{user}",User.**class**);
24. User user = (User) value.getValue(el);
26. **return** user;
27. }
28. }

Usando @Named e @Inject para pegar dados de outro bean

1. **package** bean;
3. **import** java.io.Serializable;
4. **import** javax.faces.bean.SessionScoped;
5. **import** javax.inject.Inject;
6. **import** javax.inject.Named;
8. @Named
9. @SessionScoped
10. **public** **class** User **implements** Serializable{
12. @Inject
13. **private** Profession profession;
15. **public** Profession getProfession() {
16. **return** profession;
17. }
19. **public** **void** setProfession(Profession profession) {
20. **this**.profession = profession;
21. }
23. }
24. **package** bean;
26. **import** java.io.Serializable;
27. **import** javax.annotation.PostConstruct;
28. **import** javax.inject.Named;
30. /\*\*
31. \*
32. \* @author Deivid
33. \*/
34. @Named
35. **public** **class** Profession **implements** Serializable {
37. **private** String title;
39. **private** String industry;
41. @PostConstruct
42. **public** **void** ini() {
43. **this**.title = "Software Engineer";
44. **this**.industry = "TI";
45. }
47. **public** String getTitle() {
48. **return** title;
49. }
51. **public** **void** setTitle(String title) {
52. **this**.title = title;
53. }
55. **public** String getIndustry() {
56. **return** industry;
57. }
59. **public** **void** setIndustry(String industry) {
60. **this**.industry = industry;
61. }
63. }