



Universidad de
Oviedo



MANAGED BEANS

Enol García González
Universidad de Oviedo
10 de noviembre de 2025

CONTENIDOS

1 Beans

¿QUÉ ES UN BEAN?

Definición

Componente software reutilizable para proveer de atributos y comportamiento.

¿Y un Bean Gestionado?

Es un Bean del que no nos vamos a preocupar de gestionar. JSF lo gestionará y nosotros solo tendremos que utilizarlo para proveer a las vistas xhtml de atributos y métodos

MANAGED BEAN – EJEMPLO

```
@ManagedBean // En versiones antiguas
@Named("healthPlanHandler")
public class HealthPlanHandler {
    private String employeeName="JoeUser";
    private String employeeId="a1234"
    private String healthPlanName="CareFirst";

    public String getEmployeeName() { /** ... */ }
    public void setEmployeeName(String employeeName) { /** ... */ }
    public String getEmployeeId() { /** ... */ }
    public void setEmployeeId(String employeeId) { /** ... */ }
    public String getHealthPlanName() { /** ... */ }
    public void setHealthPlanName(String healthPlanName) { /** ... */ }
    public String doRegistration() { /** ... */ }
```

MANAGED BEAN – EJEMPLO

```
[...]
<div align="center">
<h1 class="title">Health Plan Signup</h1>
<br/>
<h:form>
    Employee Name:
    <h:inputText value="#{healthPlanHandler.employeeName}" /><br/>
    Employee ID:
    <h:inputText value="#{healthPlanHandler.employeeId}" /><br/>
    Health Plan Name:
    <h:inputText value="#{healthPlanHandler.healthPlanName}" /><br/>
    <h:commandButton value="Register" action="#{healthPlanHandler.doRegistration
        }"/>
</h:form>
[...]
```

MANAGED BEAN – CONSIDERACIONES

- Todas las propiedades a las que se accede tienen que tener getters y setters.
- Los getters y setters se DEBEN llamar get... y set... con la primera letra mayúscula.
- Para las propiedades booleanas se puede usar tanto el get como el is.
- PROBLEMA: Los métodos getter se llaman múltiples veces durante el ciclo de vida de JSF.
- No incluir código pesado en los getters y setters. Por ejemplo, acceso a base de datos.

MANAGED BEAN – REGISTRO

Con la anotación @ManagedBean sobre la clase (2.0 <= JSF < 2.2), con la anotación @Named (JSF >= 2.2) o con el siguiente código XML en el fichero faces-config.xml

```
<!-- faces-config.xml -->

<managed-bean>
    <managed-bean-name>healthPlanHandler</managed-bean-name>
    <managed-bean-class>com.tew.beans.HealthPlanHandler</managed-bean-class>
</managed-bean>
```

MANAGED BEAN – ÁMBITOS

- Request. El ámbito por defecto
- Session
- Application
- View
- Conversation

En las versiones 1.x sólo existían Request, Session y Application.

Típicamente utilizaremos Session y Application con las anotaciones @SessionScoped y @ApplicationScoped

MANAGED BEAN – VIEWSCOPE

- El Bean se mantiene con vida mientras se está mostrando la misma vista.
- En el momento en el que el usuario cambia de vista se elimina el Bean.
- Se suele utilizar con Ajax

MANAGED BEAN – CONVERSATIONSCOPE

- Este es un Scope característico de JSF que no se suele ver en la mayoría de aplicaciones web.
- El ManagedBean se crea y mantiene con vida mientras dure una “conversación”.
- Una conversación hace referencia a un conjunto de peticiones agrupadas.
- Es similar a la sesión, pero en este caso nosotros controlamos la conversación.
- Existe un objeto de tipo Conversation con los métodos begin y end con el que se gestiona la duración de la conversación.

MANAGED BEAN – PSEUDOÁMBITOS

- Los ámbitos que vimos hasta ahora son ámbitos normales. De hecho todos implementan la anotación @NormalScoped, que es la que se utiliza para definir este tipo de ámbitos.
 - En JSF hay otro tipo de ámbito que en realidad es un pseudo-scope. Es el tipo @Dependant.
 - El tipo @Dependant sirve para marcar objetos que serán dependientes de otro MB, y queremos que JSF se encargue de mantener el ciclo de vida de ambos elementos de forma coordinada.
 - En la asignatura no lo vamos a utilizar.
 - Esta información sólo está por si os aparece en alguna búsqueda de documentación cuando estéis resolviendo dudas en internet.
-