For the discussion we will look at different types of Enterprise Java Beans EJBs. Select **one** of the topics below. Then, answer the “what,” “how,” and “why” questions below about your selected topic. Provide a simple code or code snippet example to further illustrate your thoughts.

* Session Beans
* Entity
* Message-Driven Beans

**“What,” “How,” and “Why” Questions**

* **~~What:~~**~~What are you writing about? Give the audience a brief overview of the topic by providing them with foundational information (history, background information, etc.).~~
* **How:** How is the information relevant? Apply personal knowledge (this can be through research or actual practiced knowledge) to build trust with the audience.
* **Why:** Justify your position and/or course of action. The audience needs proof the information you are presenting is creditable and actionable.

For this discussion post, I will discuss session beans and explore what, how, and why behind it all. According to Oracle, a session bean "encapsulates business logic" so clients can use it through "local, remote, and web service client views". The session bean "performs work for its client, shielding it from complexity by executing business tasks inside the server" (Oracle, n.d.). There are stateful, stateless, and singleton bean types. Stateful session beans represent the current state of a "unique client/bean session" (Oracle, n.d.). The state of a session bean is only kept while a session is active; it ends when the bean is removed (Oracle, n.d.). Stateless session beans do not continue a "conversational state with the client," a specific state may be invoked through a variable, so the state is not retained (Oracle, n.d.). A singleton session bean is "instantiated once per application" and remains for the rest of the application's life (Oracle, n.d.). There can only be one singleton session bean per application (Oracle, n.d.). Session beans can be used to show the interaction between a client and a bean, and when information needs to be held (Oracle, n.d.). Knowing the difference between the session beans helps understand how to utilize them best.

This information is relevant for a few reasons. Session beans expand how Java EE can be utilized to its fullest. Session beans provide a way to store and process business logic, manage sessions for all users, and improve security functions. Because of this, they are well worthy of implementation in specific Java EE code.

My information is credible and can be used when developing session beans. All information has been sourced from the Oracle website, so it is reliable information from the direct source. Understanding session beans also means it is easier to implement them in code.

Here is an example of a stateless session bean (Oracle, n.d.):

package com.sun.tutorial.javaee.ejb;

import javax.ejb.Stateless;

import javax.jws.WebMethod;

import javax.jws.WebService;

@Stateless

@WebService

public class HelloServiceBean {

private String message = "Hello, ";

public void HelloServiceBean() {}

@WebMethod

public String sayHello(String name) {

return message + name + ".";

}

}

**Reference**

Oracle. (n.d.). *What Is a Session Bean? - The Java EE 6 Tutorial*. Docs.oracle.com. Retrieved June 9, 2025, from https://docs.oracle.com/javaee/6/tutorial/doc/gipjg.html

**Assignment Requirements and Grading:**

* An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CST**.
* Submit your post by clicking on the assignment link above, then Create Thread. You must create a thread in order to view your peers' posts. Tip: Create your post in a Word document and then copy and paste your work into the thread.
* A minimum of three (3) responses, to the original threads of other students, of 100-200 words each are due by **Sunday, 11:59 p.m., CST**.
* This discussion board is worth **25 Points**.
* To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric.](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf)

Hey, Jessica! You did a fantastic job on your discussion board for this module! You thoroughly defined the what, how, and why of Session Beans. The code you included perfectly elaborates on your thoughts. I also chose a stateless session bean example for my post since I also chose to write about Session Beans in Java. I liked learning more about the background of Session Beans. You mentioned that there are stateless and stateful session beans, which is correct! There is actually a third type called a singleton session bean. When a singleton session bean is used, that means there is only one per application.

Hello, Samir! I really enjoyed reading your post for this module. I think you did a great job of touching on what session beans are, why they should be used, and how they are relevant. I am glad that you mentioned that session beans encapsulate the business logic of an application, since it is the main point of session beans. I also found there to be three types of session beans. Session Beans really do simplify development. Any time a feature or part of a programming language can be reused, it adds to its value since reusing code can be beneficial for many reasons.

Hi there, Jacob! I think your post for this discussion was very well said. It is funny that you mentioned that message-driven beans do not get enough attention. I try to reply to posts on each topic, so I am not just repeatedly reading and replying to the same topic with each reply. But most people, including myself, chose to write about session beans, which seems to prove your point. When an element in a language is more backend-focused, it can be easy to overlook, even when it is beneficial. I liked hearing about your personal experience working on a mock logistics app.