* [**http://wiki.netbeans.org/CreatingEJB3UsingNetbeansAndGlassfish**](http://wiki.netbeans.org/CreatingEJB3UsingNetbeansAndGlassfish)
* [**https://www.slideshare.net/emprovise/21-ejb3-introduction**](https://www.slideshare.net/emprovise/21-ejb3-introduction)
* **Stateless session bean**

A **stateless session bean** does not maintain a conversational state with the client. When a client invokes the methods of a stateless bean, the bean’s instance variables may contain a state specific to that client but only for the duration of the invocation. When the method is finished, the client-specific state should not be retained. Clients may, however, change the state of instance variables in pooled stateless beans, and this state is held over to the next invocation of the pooled stateless bean. Except during method invocation, all instances of a stateless bean are equivalent, allowing the EJB container to assign an instance to any client. That is, the state of a stateless session bean should apply across all clients. Because they can support multiple clients, stateless session beans can offer better scalability for applications that require large numbers of clients. Typically, an application requires fewer stateless session beans than statefull session beans to support the same number of clients. A stateless session bean can implement a web service, but a statefull session bean cannot.

* **Statefull session bean**

The state of an object consists of the values of its instance variables. In a **statefull session bean**, the instance variables represent the state of a unique client/bean session. Because the client interacts (“talks”) with its bean, this state is often called the **conversational state**.

As its name suggests, a session bean is similar to an interactive session. A session bean is not shared; it can have only one client, in the same way that an interactive session can have only one user. When the client terminates, its session bean appears to terminate and is no longer associated with the client.

The state is retained for the duration of the client/bean session. If the client removes the bean, the session ends and the state disappears. This transient nature of the state is not a problem, however, because when the conversation between the client and the bean ends, there is no need to retain the state.

* **Singleton bean**
* A **singleton session bean** is instantiated once per application and exists for the lifecycle of the application. Singleton session beans are designed for circumstances in which a single enterprise bean instance is shared across and concurrently accessed by clients.
* Singleton session beans offer similar functionality to stateless session beans but differ from them in that there is only one singleton session bean per application, as opposed to a pool of stateless session beans, any of which may respond to a client request. Like stateless session beans, singleton session beans can implement web service endpoints.
* Singleton session beans maintain their state between client invocations but are not required to maintain their state across server crashes or shutdowns.
* Applications that use a singleton session bean may specify that the singleton should be instantiated upon application startup, which allows the singleton to perform initialization tasks for the application. The singleton may perform cleanup tasks on application shutdown as well, because the singleton will operate throughout the lifecycle of the application.
* **Message driven bean**

A **message-driven bean** is an enterprise bean that allows Java EE applications to process messages asynchronously. This type of bean normally acts as a JMS message listener, which is similar to an event listener but receives JMS messages instead of events. The messages can be sent by any Java EE component (an application client, another enterprise bean, or a web component) or by a JMS application or system that does not use Java EE technology. Message-driven beans can process JMS messages or other kinds of messages.

Research and read on "Entity lifecycle" for:  
  
\* Stateless Session bean

\* Stateful session bean