1 EJB

1.1 Sumario

2 Desenvolvemento baseado en compoñentes

Coa tecnoloxía J2EE Enterprise JavaBeans é posible desenvolver compoñentes (enterprise beans) que logo se poden reutilizar e ensamblar en distintas aplicacións empresariais Por exemplo, pódese desenvolver un bean Cliente que represente un cliente nunha base de datos. Pódese usar despois ese bean Cliente nun programa de contabilidade ou nunha aplicación de comercio electrónico ou virtualmente en calquera programa no que se necesite representar un cliente. De feito, ata sería posible que o desenvolvedor do bean e o ensamblador da aplicación non fosen a mesma persoa, ou nin sequera traballasen na mesma empresa.

O desenvolvemento baseado en compoñentes é un paso máis na programación orientada a obxectos. Coa programación orientada a obxectos pódense reutilizar clases, pero con compoñentes é posible reutilizar un maior nivel de funcionalidades e ata é posible modificar estas funcionalidades e adaptalas a cada contorno de traballo particular sen tocar o código do compoñente desenvolvido.

Un compoñente é como un obxecto tradicional cun conxunto de servizos adicionais soportados en tempo de execución polo **contedor de compoñentes**. O contedor de compoñentes denomínase contedor EJB e, salvando as distancias, pódese ver como o sistema operativo no que estes residen.

Asemade, con RMI é posible enviar peticións a obxectos que están executándose noutra JVM. Podemos ver un compoñente EJB como un obxecto remoto RMI que reside nun contedor EJB que lle proporciona un conxunto de servizos adicionais.

O **despregamento** (*deployment*) do compoñente e tan importante como o seu desenvolvemento. Entendemos por despregamento a incorporación do compoñente ao contedor EJB e ao contorno de traballo (bases de datos, arquitectura da aplicación, etc.). **O despregamento defínese de forma declarativa**, mediante un ficheiro XML (**descriptor do despregamento**, *deployment descriptor*) no que se definen todas as características do bean.

A API para os EJB está definida no paquete javax.ejb de J2EE.

3 Servizos do contedor EJB

Os compoñentes "viven" nun contedor EJB que os envolve proporcionando unha capa de servizos engadidos. Os máis importantes son os seguintes:

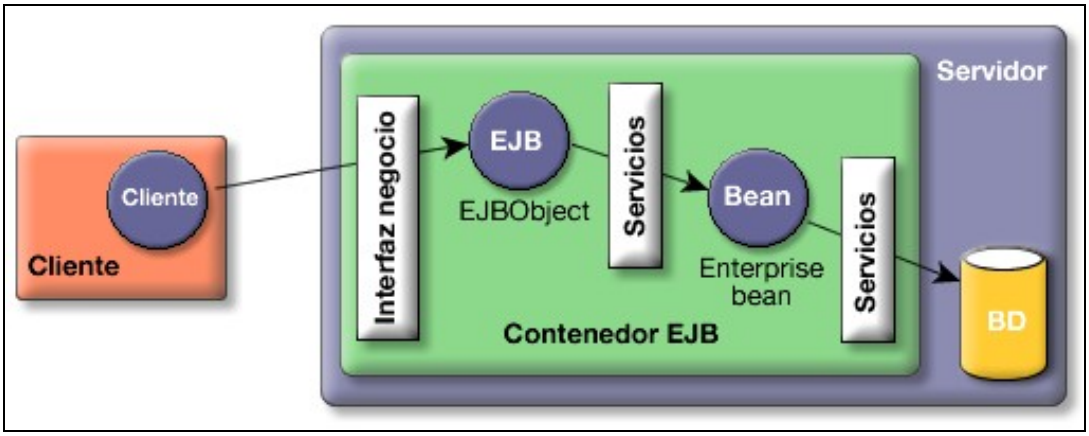
- **Manexo de transaccións**: apertura e peche de transaccións asociadas ás chamadas aos métodos do bean.
- **Seguridade**: comprobación de permisos de acceso aos métodos do bean.
- **Concorrencia**: chamada simultánea a un mesmo bean desde múltiples clientes.
- **Servizos de rede**: comunicación entre o cliente e o bean en máquinas distintas.
- **Xestión de recursos**: xestión automática de múltiples recursos, como colas de mensaxes, bases de datos ou fontes de datos en aplicacións herdadas que non foron traducidas a novas linguaxes/contornos e seguen usándose na empresa.
- **Persistencia**: sincronización entre os datos do bean e táboas dunha base de datos.
- **Xestión de mensaxes**: manexo de *Java Message Service* (JMS).
- **Escalabilidad**: posibilidade de constituír clusters de servidores de aplicacións con múltiples hosts para poder dar resposta a aumentos repentinos de carga da aplicación con só engadir hosts adicionais.
- **Adaptación en tempo de despregamento**: posibilidade de modificación de todas estas características no momento do despregamento do bean.

4 Funcionamento dos compoñentes EJB

O funcionamento dos compoñentes EJB baséase fundamentalmente no traballo do contedor EJB. O contedor EJB é un programa Java que corre no

servidor e que contén todas as clases e obxectos necesarios para o correcto funcionamento dos enterprise beans.

Na figura seguinte pódese ver unha representación de moi alto nivel do funcionamento básico dos enterprise beans:



En primeiro lugar, o cliente que realiza peticións ao bean e o servidor que contén o bean están executándose en máquinas virtuais Java distintas. Ata poden estar en distintos hosts. Outra cousa a resaltar é que o cliente nunca se comunica directamente co enterprise bean, senón que o contedor EJB proporciona un **EJBObject** que fai de interfaz. Calquera petición do cliente (unha chamada a un método de negocio do enterprise bean) débese facer a través do obxecto EJB, o cal solicita ao colector EJB unha serie de servizos e comunícase co enterprise bean. Para rematar, o bean realiza as peticións correspondentes á base de datos.

O contedor EJB preocúpase de cuestións como:

- Ten o cliente permiso para chamar ao método?
- Hai que abrir a transacción ao comezo da chamada e pechala ao rematar.
- É necesario refrescar o bean cos datos da base de datos?

Imos ver un exemplo para entender o fluxo de chamadas. Supoñamos que temos unha aplicación de bolsa e o bean proporciona unha implementación dun Broker. A interfaz de negocio do Broker está composta por varios métodos, entre eles, por exemplo, os métodos compra ou vende. Supoñamos que desde o obxecto cliente queremos chamar ao método compra. Isto vai provocar a seguinte secuencia de chamadas:

1. **Cliente**: "Necesito realizar unha petición de compra ao bean Broker."
2. **EJBObject**: "Espera un momento, necesito comprobar os teus permisos."
3. **Contedor EJB**: "Si, o cliente ten permisos suficientes para chamar ao método compra."
4. **Contedor EJB**: "Necesito un bean Broker para realizar unha operación de compra. E non esquezades comezar a transacción no momento de instanciarvos."
5. **Pool de beans**: "A ver... a quen de nós lle toca esta vez?".
6. **Contedor EJB**: "Xa teño un bean Broker. Pásalle a petición do cliente."

5 Tipos de beans

A tecnoloxía EJB define tres tipos de beans: beans de sesión, beans de entidade e beans dirixidos por mensaxes.

Os **beans de entidade** representan un obxecto concreto que ten existencia nalgunha base de datos da empresa. Unha instancia dun bean de entidade representa unha fila nunha táboa da base de datos. Por exemplo, poderiamos considerar o bean Cliente, cunha instancia do bean sendo Eva Martínez (ID# 342) e outra instancia Francisco Gómez (ID# 120).

Os **beans dirixidos por mensaxes** poden escoitar mensaxes dun servizo de mensaxes JMS. Os clientes destes beans nunca os chaman directamente, senón que é necesario enviar unha mensaxe JMS para comunicarse con eles. Os beans dirixidos por mensaxes non necesitan obxectos EJBObject porque os clientes non se comunican nunca con eles directamente. Un exemplo de bean dirixido por mensaxes podería ser un bean ListenerNuevoCliente que se activase cada vez que se envía unha mensaxe comunicando que se deu de alta a un novo cliente.

Un **bean de sesión** representa un proceso ou unha acción de negocio. Normalmente, calquera chamada a un servizo do servidor debería comezar cunha chamada a un bean de sesión. Mentres que un bean de entidade representa unha cousa que se pode representar cun nome, ao pensar nun bean de sesión deberías pensar nun verbo. Exemplos de beans de sesión poderían ser un carriño da compra dunha aplicación de negocio electrónico ou un sistema verificador de cartóns de crédito.

6 Pasos para o desenvolvemento de beans

O desenvolvemento e programación dos beans adoita ser un proceso bastante similar sexa cal for o tipo de bean. Consta dos seguintes 5 pasos:

1. Escribir e compilar a clase bean que contén a todos os métodos de negocio.
2. Escribir e compilar as dúas interfaces do bean: home e compoñente.
3. Crear un descritor XML do despregamento no que se describa que é o bean e como debe manexarse. Este ficheiro debe chamarse **ejb-jar.xml**.
4. Poñer a clase bean, os interfaces e o descritor XML do despregamento nun ficheiro EJB JAR . Podería haber máis dun bean no mesmo ficheiro EJB JAR, pero nunca haberá máis dun descritor de despregamento.
5. Despregar o bean no servidor usando as ferramentas proporcionadas polo servidor de aplicacións.

7 Roles EJB

A arquitectura EJB define seis papeis principais:

- **Desenvolvedor de beans**: desenvolve os compoñentes enterprise beans.
- **Ensamblador de aplicacións**: compón os enterprise beans e as aplicacións cliente para conformar unha aplicación completa
- **Despregador**: desprega a aplicación nun contorno operacional particular (servidor de aplicacións)
- **Administrador do sistema**: configura e administra a infraestrutura de computación e de rede do negocio
- **Proporcionador do Contedor EJB e Proporcionador do Servidor EJB**: un fabricante (ou fabricantes) especializado en manexo de transaccións e de aplicacións e outros servizos de baixo nivel. Desenvolven o servidor de aplicacións.

8 Fontes de información e recursos

- Introdución a EJB (principal fonte destes apuntes)
- Exemplo de creación, compilación e despregamento dun EJB.
- Exame tipo test sobre a API da especificación 3.0
- Exame tipo test sobre EJB