

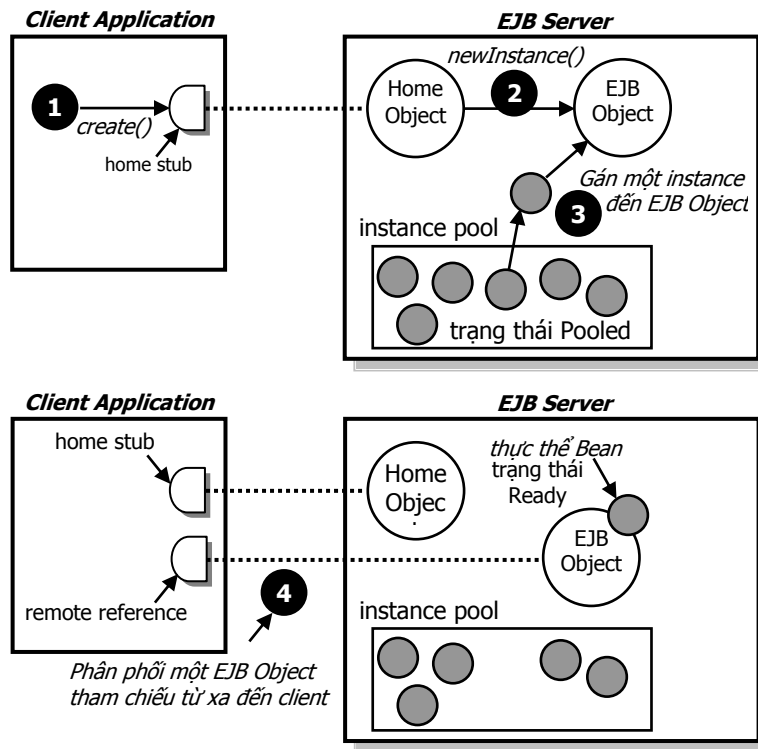
Enterprise Java Beans

1. Local và Remote interface

2. Quản lý tài nguyên

a) Instance Pool

- Các thực thể bean sẽ được tạo khi container hoạt động. Instance pooling là cơ chế để thu giảm số thực thể bean cần có mặt cho việc phục vụ các yêu cầu của client, giảm chi phí tài nguyên khi tạo và hủy bean bằng cách dùng lại các thực thể có trong pool. Như vậy thu giảm việc xử dụng tài nguyên.



- Các instance pool đều cố gắng quản lý các tập hợp bean sao cho có thể truy xuất nhanh các bean trong thời gian chạy. Để thiết lập instance pool, container tạo một số thực thể của lớp bean và giữ chúng cho đến khi cần đến. Khi client yêu cầu một business method, thực thể bean từ pool được gán đến EJB Object liên kết với client. Khi một EJB Object không cần đến thực thể, bean được trả về cho instance pool.

- Một EJB server duy trì nhiều instance pool cho mỗi kiểu pool được triển khai. Mọi thực thể trong instance pool là tương đương với nhau. Thực thể bean sẽ được chọn tùy ý để gán đến EJB Object cần đến nó. Số thực thể trong bean dao động do hoạt động vào/ra của bean.

- Sau khi thực thể bean được đặt trong pool, gọi là trạng thái Pooled, nó lấy tham chiếu đến một **javax.ejb.EJBContext**. **EJBContext** cung cấp một giao diện để bean có thể dùng liên lạc với môi trường EJB.

- Khi bean liên kết với một EJB Object, nó ở trạng thái Ready. Lúc này **EJBContext** có một ý nghĩa mới, nó cung cấp thông tin về client dùng bean, cần khi bean truyền tham chiếu đến chính bean hoặc các bean khác.

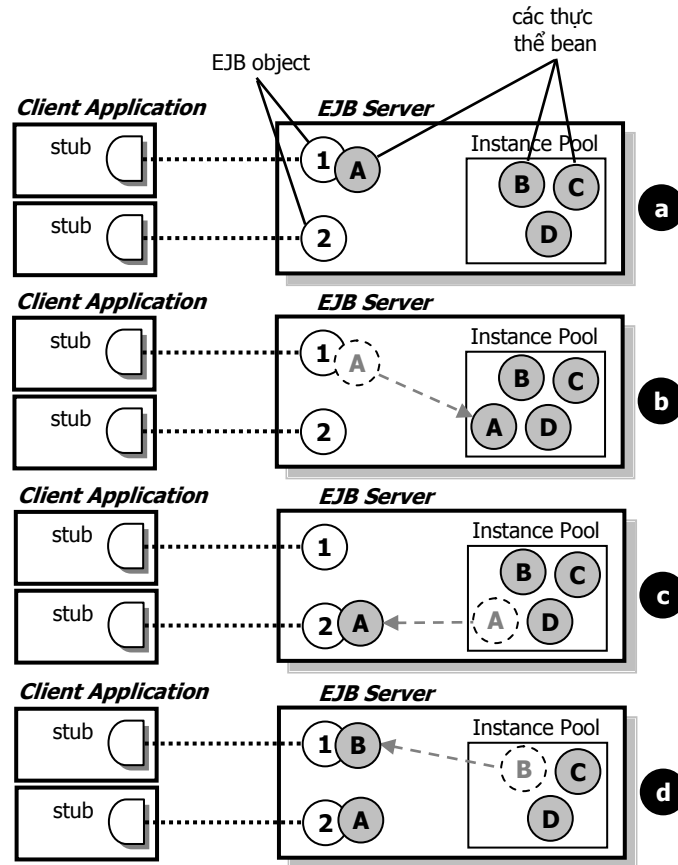
b) Instance swapping (hoán chuyển thực thể bean)

- Sau khi phục vụ yêu cầu của client, bean tách rời EJB Object và quay trở về instance pool. Nếu client yêu cầu đến EJB Object mà không có thực thể bean liên kết với nó, container sẽ tìm bean trong pool và lại gán cho EJB Object. Thao tác này gọi là instance swapping.

- Stateless Session Bean không cần lưu trữ trạng thái conversation trong nó nên mọi triệu gọi phương thức từ nó tiến hành độc lập. Điều này có nghĩa bất kỳ Stateless Session Bean nào cũng có thể phục vụ yêu cầu cho bất kỳ EJB Object nào có kiểu hợp lệ ủy nhiệm đến nó. Container có thể hoán chuyển các thực thể bean vào hoặc ra giữa các triệu gọi phương thức.

- Hình dưới mô tả cơ chế hoán chuyển thực thể khi triệu gọi phương thức của Stateless Session Bean: Bean A phục vụ một triệu gọi business method được ủy nhiệm từ EJB Object 1 (a). Khi bean A thực hiện xong yêu cầu nó chuyển ngược trở về instance pool (b). Khi client triệu gọi phương thức trên EJB Object 2, bean A lại liên kết với EJB Object 2 để phục vụ yêu cầu này (c). Khi bean A đang phục vụ, nếu có một triệu gọi trên EJB Object 1, bean B sẽ phục vụ yêu cầu này.

- Dùng chiến lược hoán chuyển này, vài bean có thể phục vụ cho hàng trăm client. Khi một thực thể bean chấm dứt phục vụ, nó lập tức trở về trạng thái sẵn sàng trong instance pool cho một EJB Object bất kỳ cần ủy nhiệm đến nó.



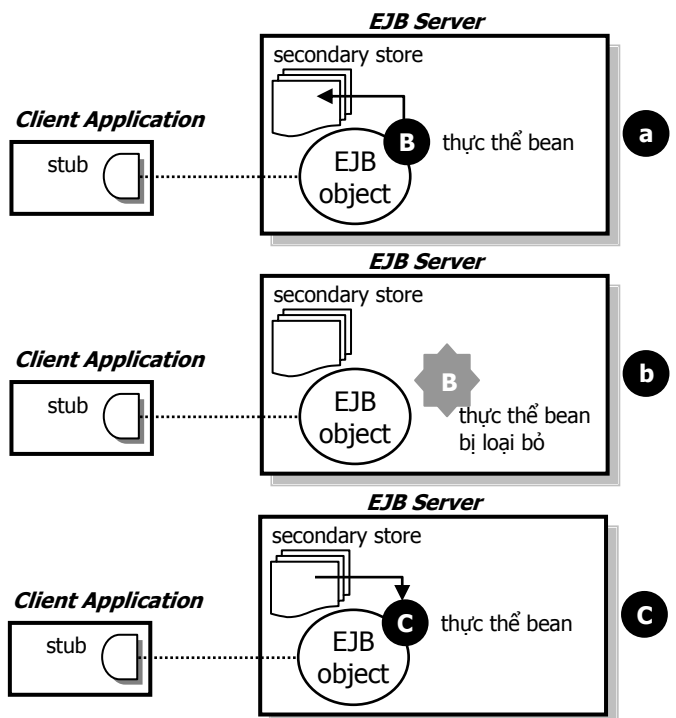
c) Cơ chế activation và passivation

- Không giống như Stateless Session Bean, Entity Bean và Message-Driven Bean, Stateful Session Bean không tham gia vào instance pool. Thay vào đó nó dùng cơ chế passivation và activation để tiết kiệm tài nguyên. Khi container cần tiết kiệm tài nguyên, nó có thể loại bean ra khỏi bộ nhớ, lúc này trạng thái của bean sẽ được lưu trữ (serialize) xuống thiết bị lưu trữ. Khi client triệu gọi phương thức của bean, một thực thể bean mới lại được tạo ra, chứa trạng thái conversation do bean trước lưu trữ.

- Passivation (thụ động hóa) tách bean ra khỏi EJB Object liên kết với nó và lưu trạng thái chứa trong bean (trạng thái kết liên hệ với EJB Object và trạng thái conversation), sau đó loại bean ra khỏi bộ nhớ. Client không biết điều này và có thể vẫn liên lạc với EJB Object trong khi bean đã thụ động hóa.

- Activation (kích hoạt) hồi phục lại thực thể bean liên kết với EJB Object. Lúc này container sẽ tự động tạo một thực thể mới và thiết lập các field theo dữ liệu được lưu trữ khi passivation. Sau đó EJB Object có thể ủy nhiệm các triệu gọi phương thức từ client đến bean như bình thường.

- Hình bên trình bày quá trình passivation và activation một Stateful Session Bean. (a) bean B được thụ động hóa, trạng thái của bean B được lưu trữ; (b) bean B tách ra khỏi EJB Object, loại khỏi bộ nhớ; (c) bean được kích hoạt, một thực thể mới, bean C được tạo ra với thông tin lưu trữ từ bean B và liên kết với EJB Object từ các thông tin nhận được.



3. Session Bean

- Một Session Bean thực hiện một conversation giữa client và server. Session Bean thực hiện một tác vụ bussiness theo yêu cầu của client trong một session đơn. Chúng thực hiện các bussiness logic (logic nghiệp vụ) như: workflow (chuỗi công việc), algorithm (thuật toán cho một công việc), bussiness rule (kiểm tra các luật bussiness).

- Một Session Bean giống như một phần mở rộng của client trên server, như bussiness của client được tách ra và chuyển đến server. Thực tế Session Bean cũng có thể nằm phía client (client tier) như một application client J2EE.

- Ví dụ một Session Bean có thể gửi email, giúp quản lý workflow, thực hiện các thuật toán như nén, mã hóa, ...

- Session Bean có các tính chất sau:

- Thể hiện một conversation giữa client và server.
- Thực hiện đại diện cho một client đơn. Không thể chia sẻ cho nhiều client *cùng lúc*. Như vậy không thể gọi bởi cùng client bằng cách dùng multithread.
- Có thể thực hiện transaction và security.
- Không thể hiện dữ liệu trực tiếp trong cơ sở dữ liệu. Tuy nhiên chúng có thể truy xuất và cập nhập dữ liệu giống như client thường làm.
- Có đời sống ngắn. Nó bị loại bỏ khi client loại bỏ nó lúc chấm dứt session, hoặc khi container ngưng vì một lý do nào đó.

- Có hai kiểu session bean:

- Stateless Session Bean: (bean không lưu vết) không lưu giữ trạng thái của conversation liên kết với một client, gọi là client-specific state (trạng thái đặc trưng cho client. Client phải chịu trách nhiệm duy trì trạng thái conversation nếu cần. Container sẽ dùng lại cùng một thực thể bean để phục vụ cho nhiều client.

Ví dụ: một Stateless Session Bean thực hiện việc đăng ký để kiểm tra người dùng và password, cho phép người dùng tạo một account trong hệ thống. Có remote interface như sau:

```
public interface SignOn extends EJBObject {  
    public void addUser( String username, String password ) throws RemoteException;  
    public boolean validateUser( String login, String password )  
        throws InvalidLoginException, RemoteException;  
}
```

- Stateful Session Bean: (bean lưu vết) lưu trạng thái conversation của server với client cụ thể trong các field của nó. Trạng thái conversation này sẽ được lưu trữ xuyên qua nhiều yêu cầu của client trong cùng một session.

Ví dụ: một Stateless Session Bean dùng mô phỏng phiếu đăng ký các môn học được chọn bởi một sinh viên cụ thể trong một session chọn khóa học. Có remote interface như sau:

```
public interface EnrollmentCart extends EJBObject {  
    public void addCourses( String[] courseIds ) throws RemoteException;  
    public Collection getCourses() throws RemoteException;  
    public void empty() throws RemoteException;  
}
```

dữ liệu đăng ký (trạng thái conversation) sẽ được lưu trong một field của lớp bean.

```
public class EnrollmentCartEJB implements SessionBean {  
    ...  
    private HashSet cart;  
    ...  
    public void addCourses( String[] courseIds ) {  
        if ( courseIds == null ) return;  
        for ( int i = 0; i < courseIds.length ; ++i ) cart.add( courseIds[i] );  
    }  
    public void empty() { cart.clear(); }  
}
```