

JAVA means DURGA SOFT

Design Patterns

Enterprise Application Level Design Patterns

4. Service Locator



India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Service Locator

Problem:

J2EE clients interact with service components, such as Enterprise JavaBeans (EJB) and Java Message Service (JMS) components, which provide business services and persistence capabilities. To interact with these components, clients must either locate the service component (referred to as a lookup operation) or create a new component. For instance, an EJB client must locate the enterprise bean's home object, which the client then uses either to find an object or to create or remove one or more enterprise beans. Similarly, a JMS client must first locate the JMS Connection Factory to obtain a JMS Connection or a JMS Session.

All Java 2 Platform Enterprise Edition (J2EE) applications clients use the JNDI common facility to look up and create EJB and JMS components. The JNDI API enables clients to obtain an initial context object that holds the component name to object bindings. The client begins by obtaining the initial context for a bean's home object. The initial context remains valid while the client session is valid. The client provides the JNDI registered name for the required object to obtain a reference to an administered object. In the context of an EJB application, a typical administered object is an enterprise bean's home object. For JMS applications, the administered object can be a JMS Connection Factory (for a Topic or a Queue) or a JMS Destination (a Topic or a Queue).

www.durgasoftonlinetraining.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

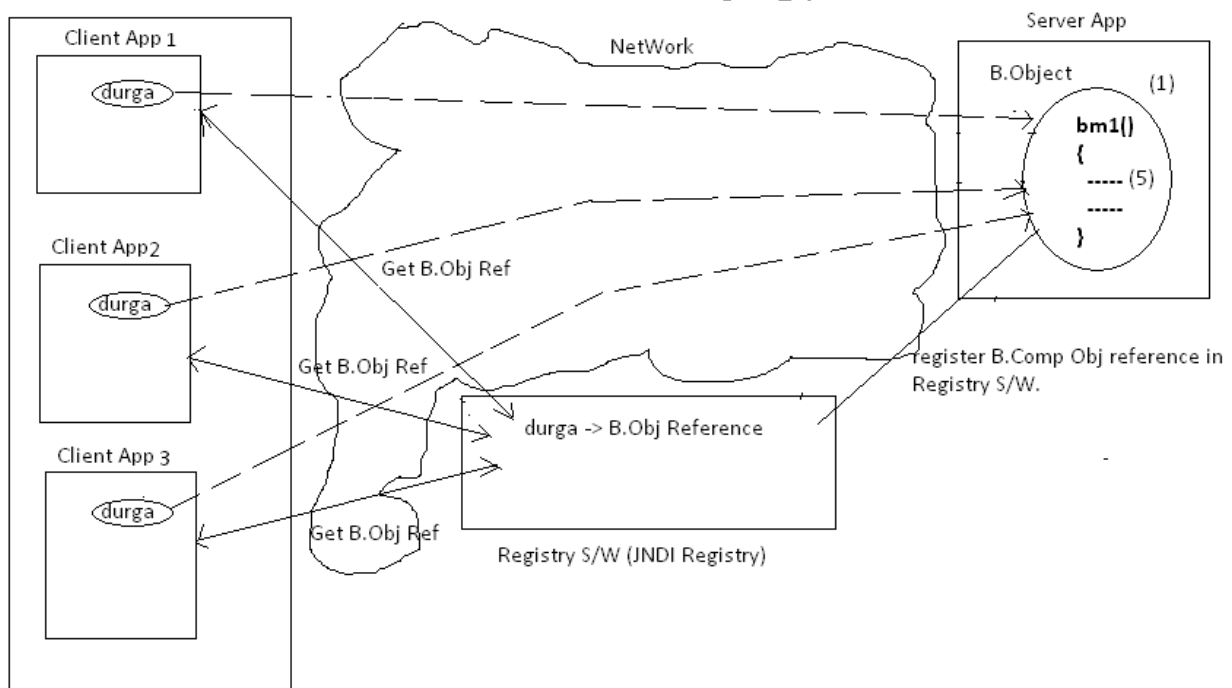
**Ph: +91-8885252627, 7207212427
+91-7207212428**

 **USA Ph : 4433326786**

E-mail : durgasoftonlinetraining@gmail.com

So, locating a JNDI-administered service object is common to all clients that need to access that service object. That being the case, it is easy to see that many types of clients repeatedly use the JNDI service, and the JNDI code appears multiple times across these clients. This results in an unnecessary duplication of code in the clients that need to look up services.

Also, creating a JNDI initial context object and performing a lookup on an EJB home object utilizes significant resources. **If multiple clients repeatedly require the same bean home object, such duplicate effort can negatively impact application performance** i.e. multiple applications of a single project wants to interact with same business component of distributed application then they need to interact with the JNDI registry to get the business object reference. In this context **each client application should interact with JNDI registry separately to get the business object reference.**

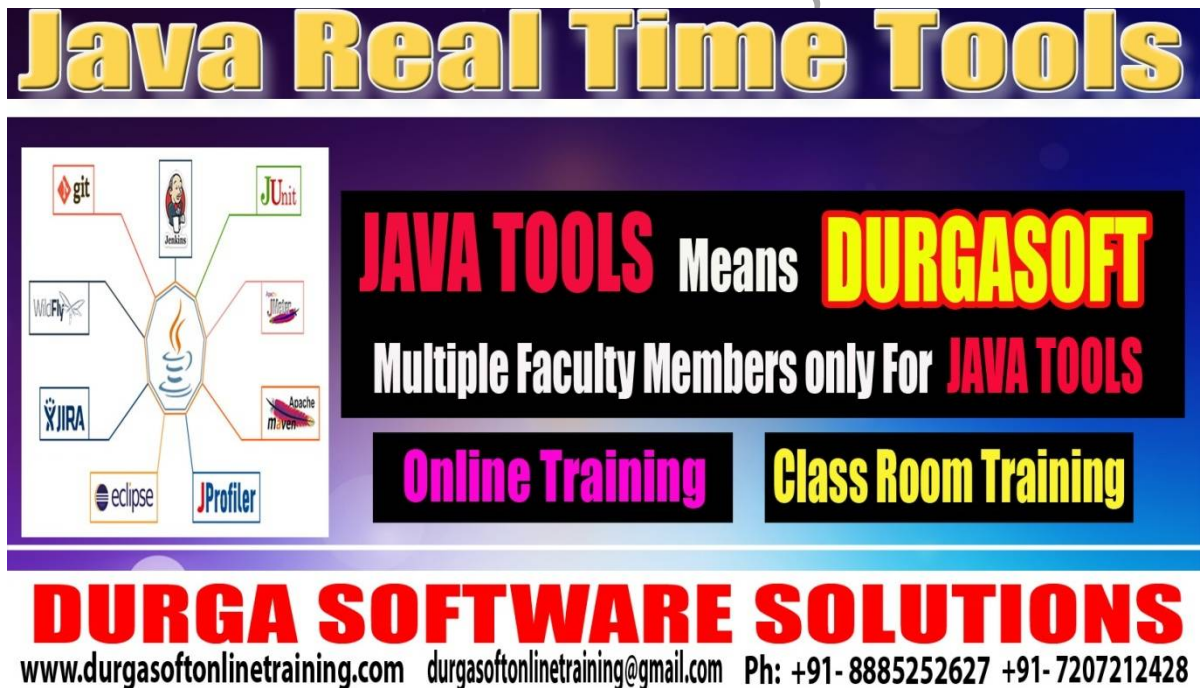


Here all the 3 client applications of single project are interacting with registry software over the network and getting the same business object reference through network.

Solution:

Use a **Service Locator** object to abstract all JNDI usage and to hide the complexities of initial context creation, EJB home object lookup, and EJB object re-creation. **Multiple clients can reuse the Service Locator object to reduce code complexity, provide a single point of control, and improve performance by providing a caching facility.**

This pattern reduces the client complexity that results from the client's dependency on and need to perform lookup and creation processes, which are resource-intensive. To eliminate these problems, this pattern provides a mechanism to abstract all dependencies and network details into the Service Locator.



The advertisement is a colorful banner. At the top, it says "Java Real Time Tools" in large, bold, yellow letters with a blue outline. Below this, on the left, is a diagram showing a central Java logo with lines connecting it to various tools: git, Jenkins, JUnit, WildFly, JIRA, Apache Maven, Eclipse, and JProfiler. To the right of the diagram, the text "JAVA TOOLS Means DURGASOFT" is written in large, bold, red and yellow letters. Below this, it says "Multiple Faculty Members only For JAVA TOOLS" in white text on a black background. Further down, there are two boxes: "Online Training" in pink text on a black background, and "Class Room Training" in yellow text on a black background. At the bottom, "DURGA SOFTWARE SOLUTIONS" is written in large, bold, red letters. Below that, the website "www.durgasoftonlinetraining.com", email "durgasoftonlinetraining@gmail.com", and phone numbers "Ph: +91- 8885252627 +91- 7207212428" are listed.

Java Real Time Tools

JAVA TOOLS Means DURGASOFT

Multiple Faculty Members only For JAVA TOOLS

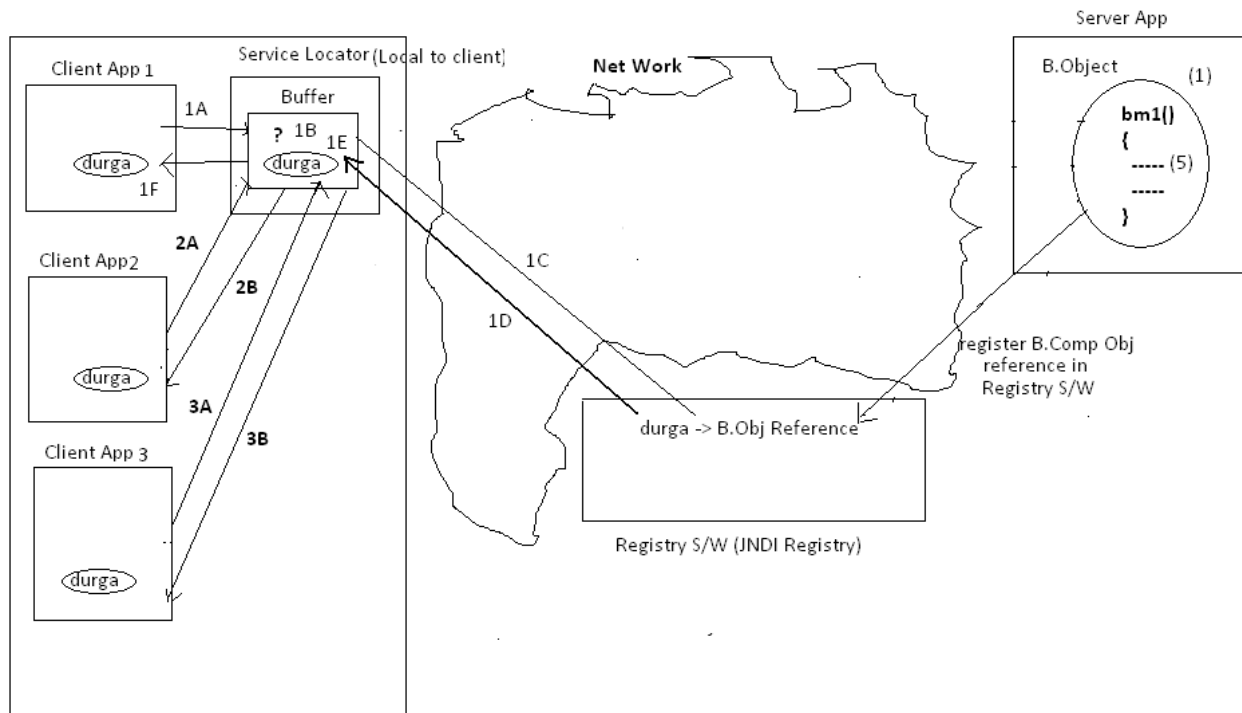
Online Training **Class Room Training**

DURGA SOFTWARE SOLUTIONS

www.durgasoftonlinetraining.com durgasoftonlinetraining@gmail.com Ph: +91- 8885252627 +91- 7207212428

The Service Locator pattern centralizes distributed service object lookups, provides a centralized point of control, and may act as a cache that eliminates redundant lookups. It also encapsulates any vendor-specific features of the lookup process.

JAVA Means DURGA SOFT



In the above diagram, client App1 makes Service Locator to get business object reference from JNDI registry software and keeps that business object reference in the local buffer (cache) of Service Locator where as the remaining client applications will directly gather business object reference from local buffer of service locator.

www.durgajobs.com

Continuous Job Updates for every hour

Fresher Jobs
Govt Jobs
Bank Jobs

Walk-ins
Placement Papers
IT Jobs

Interview Experiences

Complete Job information across India

Here **steps 1A -1F**: indicates client App1 gets business object reference from JNDI registry software, keep in local buffer and gives to client App1; **Steps 2A-2B** indicates client App2 request service locator for business object reference and then service locator gives

business object reference to client App2 by collecting from local buffer without communicating with registry software; **Steps 3A-3B** indicates client App3 request service locator for business object reference and then service locator gives business object reference to client App3 by collecting from local buffer without communicating with registry software.

Note: To make all clients working with single buffer of service locator, the service locator should be taken as **Singleton Java Class**.

Advantages:

- ✓ Lookup object in JNDI, RMI, JMS, etc.
- ✓ Encapsulate any vendor-specific features of lookup process
- ✓ Simplify the lookup process
- ✓ Improve the performance (with the Singleton pattern and caching)

Note:

- ❖ If application and its client resides & executes on the same JVM then that client is called as **Local Client to Application**.
- ❖ If application and its client resides on 2 different JVM's of same computer or on 2 different computers then that client is called as **Remote Client to application**.

LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...

JAVA MEANS DURGA SOFT

INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

AN ISO 9001:2008 CERTIFIED

DURGA

SOFTWARE SOLUTIONS

#202 2nd FLOOR

www.durgasoft.com

040-64512786

+91 9246212143

+91 8096969696

LEARN FROM EXPERTS ...

COMPLETE JAVA

CORE JAVA, ADV. JAVA, ORACLE, STRUTS, HIBERNATE, SPRING, WEB SERVICES,...

COMPLETE .NET

C#.NET, ASP.NET, SQL SERVER, MVC 5 & WCF

TESTING TOOLS

MANUAL + SELENIUM

ORACLE | D2K

MSBI | SHARE POINT

HADOOP | ANDROID

C, C++, DS, UNIX

CRT & APTITUDE TRAINING

AN ISO 9001:2008 CERTIFIED

DURGA
Software Solutions®

202, 2nd Floor, HUDA Maitrivanam,
Ameerpet, Hyd. Ph: 040-64512786,

9246212143, 8096969696

www.durgasoft.com