

## Service Locator

- ▶ **Área:** [Patrones de Diseño](#)
- ▶ **Grupo:** Capa de negocio
- ▶ **Carácter del recurso:** [Recomendado](#)

**Código:** RECU-0147

**Tipo de recurso:** Patrón

### Descripción

Este patrón se emplea para localizar de forma transparente y uniforme los componentes de negocio.

### Contexto

La búsqueda y creación de servicios implican interfaces complejos y operaciones de red.

### Problema

Los clientes J2EE interactúan con componentes de servicio, como componentes JavaBeans Enterprise (EJB) y Java Message Service (JMS), que proporcionan servicios de negocio y capacidades de persistencia. Para interactuar con estos componentes, los clientes deben o localizar el componente de servicio (referido como una operación de búsqueda) o crear un nuevo componente.

Todos los clientes de aplicaciones J2EE utilizan la facilidad JNDI para buscar y crear componentes EJB o JMS. El API JNDI permite a los clientes obtener un objeto *ContextoInicial* que contiene el nombre del componente a uniones de objetos. El cliente empieza obteniendo el *ContextoInicial* para un objeto home de un bean. El *ContextoInicial* permanece válido mientras sea válida la sesión del cliente. El cliente proporciona el nombre registrado en JNDI del objeto requerido para obtener una referencia a un objeto administrado. En el contexto de una aplicación EJB, un objeto administrado típico es un objeto home de un Bean Enterprise. Para aplicaciones JMS, el objeto administrado puede ser una Factoría de Conexiones JMS (para un Topic o una Queue) o un Destino JMS (un Topic o una Queue).

Por eso, localizar un objeto servicio administrado JNDI es un tarea común para todos los clientes que necesiten acceder al objeto de servicio. También, crear un objeto de contexto JNDI inicial y realizar una búsqueda para objeto home EJB utiliza muchos recursos. Si varios clientes requieren de forma repetitiva el mismo objeto home de un bean, dicha duplicación de esfuerzos puede impactar de forma negativa en el rendimiento de la aplicación.

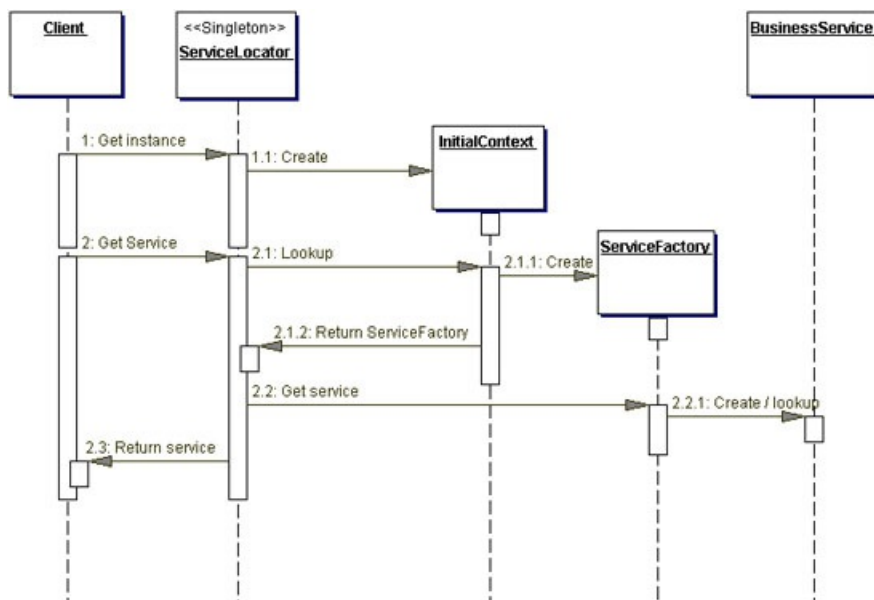
### Solución

Utilizar un objeto *Service Locator* para abstraer toda la utilización JNDI y para ocultar las complejidades de la creación del contexto inicial, de búsqueda de objetos home EJB y de re-creación de objetos EJB. Varios clientes pueden reutilizar el objeto *Service Locator* para reducir la complejidad del código, proporcionando un punto de control y mejorando el rendimiento proporcionando facilidades de caché.

Este patrón reduce la complejidad del cliente que resulta de las dependencias del cliente y de la necesidad de realizar los procesos de búsqueda y creación, que consumen muchos recursos. Para eliminar estos problemas, este patrón proporciona un mecanismo para abstraer todas las dependencias y detalles de red dentro del *Service Locator*.

### Implementación y Participantes

A continuación el modelo gráfico del diagrama de secuencia del patrón:



- **Client**: es un objeto que busca acceder a objetos de negocio.
- **ServiceLocator**: abstrae la tarea de búsqueda e instanciación de los servicios, ofreciendo un interfaz simplificado a los clientes.
- **ContextoInicial**: Es el punto de comienzo en el proceso de búsqueda y creación. Los proveedores de los servicios proveen el objeto contexto que varía en función del tipo de servicio.
- **FactoríaServicios**: Representa la implementación del registro que almacena las referencias a los servicios.
- **BusinessService**: Representa el servicio.

## Clasificación

Otros

## Enlaces externos

► [Fuente original](#)

## Pautas

Área: [Desarrollo](#) » [Patrones de Diseño](#) » [Capa de negocio](#)

Código	Título	Tipo	Carácter
<a href="#">LIBP-0347</a>	<a href="#">Uso de Patrones J2EE de la Capa de Negocio</a>	Directriz	Recomendada

## Recursos

Área: [Desarrollo](#) » [Patrones de Diseño](#)

Código	Título	Tipo	Carácter
<a href="#">RECU-0013</a>	<a href="#">Patrones de diseño</a>	Ficha	Recomendado

Source URL: <http://madeja.i-administracion.junta-andalucia.es/servicios/madeja/contenido/recurso/147>