

Comunicazione tra plugin grazie al Message Bus

Come far comunicare tra loro due plugin differenti?

Ci sono vari modi per trasferire informazioni tra due plugin differenti. Uno di questi è lo scambio di messaggi tramite Message Bus.

Possiamo individuare due ruoli principali in questa comunicazione:

- *Message producer* – Il plugin che crea il messaggio
- *Message consumer* – Il plugin che riceve il messaggio

Vediamo quali sono i passi necessari per implementare questo tipo di comunicazione

Message producer – Creare la Destination

Per prima cosa occorre creare una destinazione per i nostri messaggi. Questa operazione può essere fatta nel file `messaging-spring.xml` ma in questo modo ogni volta che il plugin sarà deployato la destination verrà ricreata, perdendo tutti i listener eventualmente associati. Un modo per ovviare a questo problema è creare la destination tramite codice java:

Il miglior momento in cui invocare questo metodo è all'interno del metodo di startup del plugin. Per i dettagli vedere [QUI](#)

Message producer – Inviare il messaggio

Per inviare il messaggio possiamo utilizzare in qualsiasi punto del nostro codice alcuni metodi statici messi a disposizione da Liferay. Il messaggio creato può

Message consumer – Implementare il listener

Per prima cosa dobbiamo implementare la classe che gestirà la ricezione del messaggio. Per farlo dobbiamo estendere la classe `com.liferay.portal.kernel.messaging.MessageListener`: in particolare la logica di gestione del messaggio deve essere implementata nel metodo `void receive(Message message)`:

Message consumer – Agganciare il listener alla destinazione

Una volta implementato il listener va agganciato alla destinazione creata. Ricordiamoci di gestire anche il redeploy del plugin, quindi di controllare se è già presente un listener del tipo implementato prima di aggiungerlo. Per farlo possiamo usare codice di questo tipo:

Non facciamoci spaventare da un codice all'apparenza più complesso del solito: si tratta di scorrere tutta la lista dei listener agganciati alla destinazione ed eventualmente sganciare il listener della classe `MyListener`. La complicazione nasce dal fatto che in alcuni casi viene utilizzato il proxy `InvokerMessageListener` per wrappare la vera classe del listener. Non ci addentriamo nei dettagli di questa scelta ma ci basta sapere come gestirla: invocando il

metodo `getMessageListener()` sul proxy, che ci restituisce la vera istanza del listener e ci permette di verificarne la classe.

Anche in questo caso conviene fare questa operazione all'interno del codice di startup del plugin in modo da essere sicuri che venga eseguita sempre quando il plugin è presente

Approfondimenti e riferimenti

- <https://www.liferay.com/it/documentation/liferay-portal/6.1/development/-/ai/lp-6-1-dgen09-using-message-bus-0>