

1 JNDI lookup 及其關聯的參考

JNDI 查找名稱	關聯的參考
java:comp/env	應用程式環境項目
java:comp/env/jdbc	JDBC DataSource 資源管理程式連線 Factory
java:comp/env/ejb	EJB 參考
java:comp/UserTransaction	UserTransaction 參考
java:comp/env/mail	JavaMail 階段作業連線 Factory
java:comp/env/url	URL 連線 Factory
java:comp/env/jms	JMS 連線 Factory 與目標
java:comp/ORB	應用程式元件共用的 ORB 實例

2 JNDI 介紹

JNDI (The Java Naming and Directory Interface) 是一個應用程式介面，它向使用 Java 語言編寫的應用提供了命名與目錄功能。

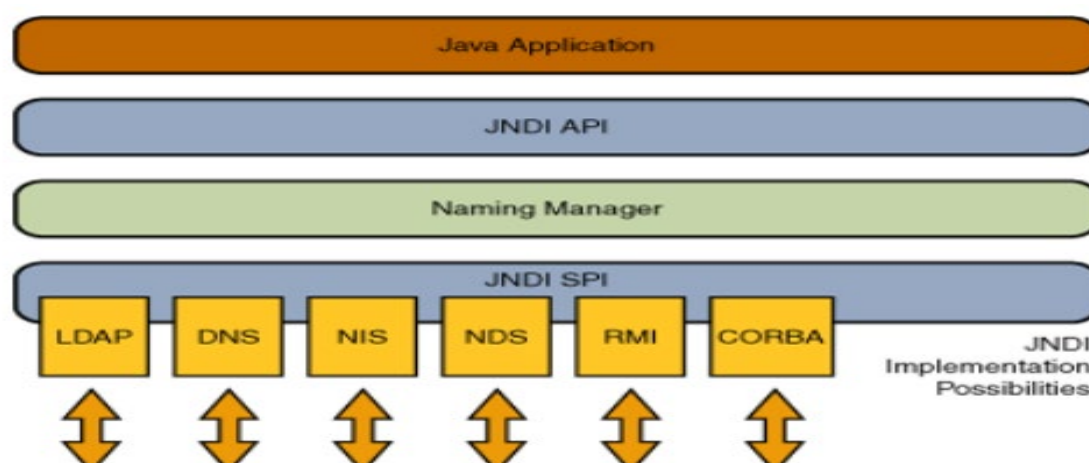
JNDI 被定義為獨立於任何特定目錄實現，所以可以使用通用的方法走訪不同的實現。

架構

JNDI 架構由 PI 與 SPI (service provider interface) 構成。

java 應用使用 JNDI 應用程式介面走訪不同的命名與目錄服務。

SPI 允許通過透明方式載入不同的命名與目錄服務，因此允許 Java 應用通過 JNDI API 走訪它們的服務。



套件

JNDI 屬於 Java SE 平臺。如果要使用 JNDI，必需具有 JNDI 類與一個或多個服務提供者。

JDK 套件含的服務提供者：

- Lightweight Directory Access Protocol (LDAP)
- Common Object Request Broker Architecture (CORBA) Common Object Services (COS) name service
- Java Remote Method Invocation (RMI) Registry
- Domain Name Service (DNS)

JNDI 被分為五個套件

- javax.naming
- javax.naming.directory
- javax.naming.ldap
- javax.naming.event
- javax.naming.spi

2.1 命名套件

javax.naming 套件中套件含了用於走訪命名服務的類與介面。

Context

javax.naming 套件定義了 **Context** 介面，用於尋找、繫結/接觸繫結、重新命名物件與建立銷毀上下文。

- Lookup : lookup()用於查詢給定名稱對應的物件。
- Bindings : listBindings()回傳名稱與物件繫結的列舉。繫結是一個名稱、類別、物件的元組。

- List：方法回傳物件名稱與物件類名的列舉。由於此操作不會回傳物件，所以比

`listBindings()`減少記憶體。

- Name：Name 是表示一般名稱的介面，命名系統使用此介面定義複合命名規範的名稱。
- References：一個 reference 是一個物件的緊湊表示，JNDI 定義 `Reference` 類表示引用。

JNDI 將在目錄中查詢到的引用轉化為 **Java** 物件，這樣可以使客戶端產生儲存在目錄中的是 **java** 物件的錯覺。

初始 Context

JNDI 之中，所有的命名與目錄操作的執行都涉及 **context**。

由於沒有絕對根目錄，JNDI 定義了 **InitialContext**，用於提供命名與目錄操作的起始點。

一旦有了一個初始 **context**，那麼就可以利用它來查詢其他 **context** 與物件。

Exceptions

JNDI 定義了一個類異常體系，用於表示執行命名與目錄操作的期間發生的異常。

此異常體系的根是 **NamingException**。

處理特定異常的程式可以用於捕獲異常的對應子類。

2.2 目錄與 LDAP 套件

目錄套件

`javax.naming.directory` 套件擴充套件了 `javax.naming` 套件用於提供走訪目錄服務的功能。

這個套件使應用可以檢索儲存在目錄中關聯的物件，或使用特定的屬性搜尋物件。

目錄 context

DirContext 介面表示一個目錄上下文。**DirContext** 擴充套件了 **Context** 介面，所以目錄物件可以提供命名上下文。

`DirContext` 定義了檢測更新關聯目錄條目的屬性的方法。

Attributes :

`getAttributes()`方法檢索關聯目錄條目的屬性。`modifyAttributes()`方法用於新增、替換、移除屬性或屬性值。

Searches :

通過將屬性集合提交給 `search()`方法來檢索物件。過載的 `search()`方法可以使用搜索過濾器進行更複雜的搜尋。

LDAP 套件

`javax.naming.ldap` 套件括使用 LDAP v3 特性的類與介面，這些特性還沒有被 `javax.naming.directory` 所覆蓋。

事實上，大多數使用 LDAP 的 JNDI 應用使用 `javax.naming.directory` 中的操作就足夠了。

`javax.naming.ldap` 主要用於需要擴充套件操作、控制或主動通知的應用。

擴充套件操作

除了指定意義明確的操作如搜尋與修改，LDAP v3 指定了一種在客戶端與服務端傳輸未定義操作的方式。

這些操作稱為"extended"操作。

擴充套件操作需要被標準組織或提供商進行定義。

控制

LDAP v3 允許請求與響應使用未定義的修飾符進行擴充套件，這些未定義的修飾符稱為控制。與請求一起傳送的控制稱之為請求控制，與響應一起傳送的控制稱之為響應控制。控制可以被標準組織或提供商定義。控制控制與響應控制並不需要成對出現。

`LdapContext` 介面表示用於執行擴充套件操作、傳送請求控制、接收響應控制的上下文。

2.3 事件與服務提供者套件

事件套件

`javax.naming.event` 套件含的類與介面在命名與目錄服務中提供事件通知功能。

`Events`，`NamingEvent` 表示一個由命名/目錄服務產生的事件。這個事件套件含一個 `type` 域，用於表示事件的型別。

`Listener`，`NamingListener` 是一個用於監聽 `NamingEvent` 的物件。每一類事件都有一個對應型別的 `NamingListener`。

為了接收事件，監聽器需要註冊在 `EventContext` 或 `EventDirContext` 中。

一旦註冊，當對應型別的改變發生在命名/目錄服務中，這個監聽器將會接收事件通知。

服務提供者套件

`javax.naming.spi` 提供了一組應用呼叫 `JNDI` 的公共介面，用於使用不同命名/目錄服務提供者的開發者開發的實現。

插入架構

`javax.naming.spi` 允許不同實現動態插入。這些實現套件括 `initial context` 與可以通過 `initial context` 到達的 `contexts`。

java 物件支援

`java.naming.spi` 支援 `Context.lookup()`與相關方法的實現回傳 `Java` 物件。

`java.naming.spi` 支援 `Context.bind()`與相關方法的實現接收 `Java` 物件，然後將物件以命名/目錄服務接受的形式儲存。

多命名系統

`JNDI` 操作允許應用提供跨多命名系統的名稱。

完成此操作，一個服務提供者需要與其他服務提供者進行互動。