

Universidade Presbiteriana Mackenzie
Faculdade de Computação e Informática

Serviços de Nomes e Diretórios - JNDI

NOTAS DE AULA

Linguagem de Programação III
2º semestre de 2016
Prof. Fabio K. Takase
Versão 1.0

Objetivos

- Entender o que são serviços de nomes e de diretório.
- Apresentar a api JNDI para acesso a estes serviços.
- Compreender uma implementação em Java que faz uso da API JNDI.

Referência

A referência para esta aula é:

<https://docs.oracle.com/javase/tutorial/jndi/overview/>

Observação: As notas de aula são material de apoio para estudo e não têm o objetivo de apresentar o assunto de maneira exaustiva. Não deixe de ler o material de referência da disciplina.

Serviço de Nomes

- Associa um nome (simples) a um recurso computacional.
 - endereços de memória, de rede, de serviços
 - objetos e referências
- Função básica:
 - associar/mapear um nome a um recurso computacional
 - localizar um recurso computacional a partir do nome

Usos típicos de Serviço de Nomes

- DNS: traduz um nome (ex: `www.mackenzie.br`) para um endereço IP (ex: `143.107.99.244`)
- Um nome em um dado servidor está associado a um componente de software.
- Um nome de arquivo está associado a um bloco de memória.

Ligações (Binding)

- Associação de um nome a um recurso computacional (ou a uma referência ao recurso)
 - Uma referência indica como podemos acessar o recurso computacional.

Convenções (Nomenclatura)

- Todos os nomes seguem regras bem definidas para sua formação.
- Ex: Nome de domínio: lidos da direita para esquerda, separados por .
 - `www.mackenzie.br`

Convenções (Nomenclatura)

- Nomes compostos: formados por nomes atômicos.
 - Ex: nomes atômicos: br, mackenzie, www compõem o nome composto: www.mackenzie.br
- Nome federado: nomes com espaço de nomes distintos em uma composição.
 - Ex: <http://www.mackenzie.br>: http:// define o protocolo - esquema de URL. www.mackenzie.br nome composto de um DNS

Contexto

- Contexto: Um objeto cujo estado é um conjunto de ligações com diferentes nomes atômicos.
- Subcontexto: um contexto limitado por outro contexto do mesmo tipo (segue as mesmas convenções de nomenclatura)

Resolução de nomes

- escopo definido por um contexto.
- cada contexto possui uma operação de resolução de nome associada.
 - Exemplo: www é subcontexto de mackenzie que por sua vez é subcontexto do contexto br.

Serviços de diretório

- Podemos estender o serviço de nomes, acrescentando atributos aos recursos mapeados.
- Devemos ter operações para buscar objetos de diretório por seu nome ou seus atributos.
- Devemos prover serviços para manter objetos de diretório.

JNDI - Java Naming and Directory Interface

JNDI

- Interface única para acesso a serviços de nomes e de diretórios.
 - API única para acesso a vários tipos de serviços de diretório
 - Isola a aplicação dos detalhes específicos do protocolo
 - Pode ser usada para ler objetos Java (serializados) que estejam armazenados em um diretório
 - Pode combinar diferentes tipos de diretório (federação) e tratá-los como um diretório único

JNDI - Componentes

- API - Application Programming Interface.
- SPI - Service Provider Interface.
- faz parte do SDK (Standard Edition) desde a versão 1.3
- pacote base: `javax.naming`
- SP para LDAP, CORBA e Java RMI Registry

JNDI - Arquitetura

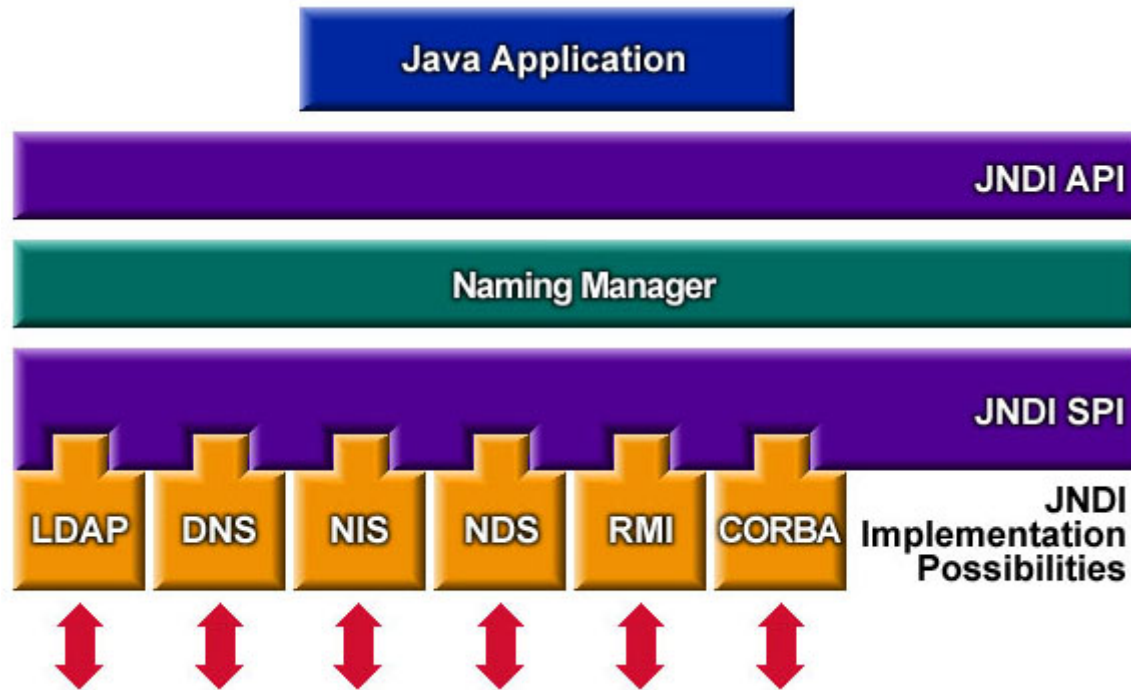


imagem do tutorial JNDI

(<http://docs.oracle.com/javase/jndi/tutorial/getStarted/overview/index.html>)

JNDI - Classes principais

- Context: interface onde se pode recuperar, ligar, desligar e renomear objetos, e criar e destruir contextos.
- InitialContext: ponto de partida para todas as operações.
- Name: abstração de um nome. Contém geralmente um String de texto que corresponde ao nome do objeto ou contexto.

InitialContext

Deve ser criado para que possamos utilizar um serviço de nomes/diretórios.

Definimos o provedor de serviço:

```
Properties env = new Properties();  
env.put(Context.INITIAL_CONTEXT_FACTORY, "classe.do.ProvedorDeServicos");
```

Configuramos o acesso ao serviço:

```
env.put(Context.PROVIDER_URL, "<url_do_prvedor>");  
env.put(Context.OUTER_PROPERTY, "valor");  
(...)
```

Criamos uma instância de InitialContext e referenciamos com uma variável da classe Context:

```
Context ctx = new InitialContext(env);
```

Lookup

- Para obter a referência para um objeto de um contexto usa-se o método `lookup()`
- Para usar o objeto retornado é preciso conhecer o seu tipo e fazer o cast (narrow)

JNDI

Exemplo (Sistema de arquivos):

```
Hashtable env = new Hashtable();  
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.fscontext.RefFSContextFactory");  
Context ctx = new InitialContext(env);  
File file = (File)ctx.lookup( "teste.txt" );
```

Obrigado!

