J2EE e Datasources

Helder da Rocha (helder@acm.org)
www.argonavis.com.br

Recursos em servidores J2EE

- Servlets rodando em servidores compatíveis J2EE podem acessar recursos através de JNDI (dominio java:comp/env)
  - Variáveis (environment entries)
  - Referências para componentes EJB
  - Referências para fábricas de recursos (conexões de banco de dados, URLs, serviço de e-mail, JMS, conectores EIS via JCA)
  - Serviços
- Para usar esses recursos
  - Servlet deve estar empacotado em um WAR
  - Nome das variáveis e referências devem ser declarados no web.xml
  - Servlet deve usar como contexto inicial o domínio java:comp/env
- Elementos (filhos de <web-app>) usados no web.xml
  - <env-entry>
  - <ejb-ref>
  - <resource-ref>

Environment Entries

- *Alternativa global (para o WAR) aos <init-param>*
  - *São acessíveis dentro de qualquer servlet ou JSP da aplicação WAR*
  - *Não são visíveis por outras aplicações do servidor (não é um nome JNDI global - está abaixo de java:comp/env - é local à aplicação)*
  - *Acessíveis via ferramentas de deployment (podem ser redefinidas)*
- *Exemplo de uso dentro do <web-app>*

```
<env-entry>
  <env-entry-name>cores/fundo</env-entry-name>
  <env-entry-value>rgb(255, 255, 200)</env-entry-value>
  <env-entry-type>java.lang.String</env-entry-type>
</env-entry>
```

  - *Tipos de dados legais são String e wrappers (Double, Integer, etc.)*
- *Uso dentro do servlet*

```
Context initCtx = new InitialContext();
String fgColor = (String)
                 initCtx.lookup("java:comp/env/cores/fundo");
```

Componentes EJB

- *Servlets e JSPs podem se comunicar com EJBs da aplicação declarando uma referência associada ao bean chamado*
  - *A referência deve informar o tipo do bean (Session, Entity ou MessageDriven e suas interfaces remota e home.*

```
<ejb-ref>
   <description>Cruise ship cabin</description>
   <ejb-ref-name>ejb/CabinHome</ejb-ref-name>
   <ejb-ref-type>Entity</ejb-ref-type>
   <home>com.titan.cabin.CabinHome</home>
   <remote>com.titan.cabin.Cabin</remote>
</ejb-ref>
```

- *Componentes EJB são retornados como objetos CORBA que precisam ser reduzidos através da função narrow.*

```
InitialContext initCtx = new InitialContext();
Object ref = initCtx.lookup("java:comp/env/ejb/CabinHome");
CabinHome home = (CabinHome)
          PortableRemoteObject.narrow(ref, CabinHome.class);
```

Conexões de banco de dados

- *Fábricas de objetos são acessíveis via <resource-ref>. A mais comum é a fábrica de conexões de banco de dados*

```
<resource-ref>
  <description>Cloudscape database</description>
  <res-ref-name>jdbc/BankDB</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>SERVLET</res-auth>
</resource-ref>
```

  - *<res-auth> informa quem é responsável pela autenticação*
- *Através da DataSource, obtém-se uma conexão.*

```
InitialContext initCtx = new InitialContext();
DataSource ds = (DataSource)
                initCtx.lookup("java:comp/env/jdbc/BankDB");
Connection con1 = ds.getConnection();// res-auth: CONTAINER
Connection con2 =
   source.getConnection("user","pass"); // res-auth: SERVLET
```

Exercício

- *1. Altere o exercício onde usamos banco de dados (DAO) e faça com que ele use um DataSource em vez da conexão JDBC usual*
  - *Se possível, use o mesmo banco (se não, crie uma base de dados e a tabela no novo banco)*
  - *Não precisa mexer no SQL.*
  - *Mude apenas a chamada para obter a conexão: obtenha a conexão com Connection con = ds.getConnection() em cada método e libere-a no final com con.close();*
- *2. Grave algumas variáveis como no ambiente do servidor usando <env-entry> e leia-as de um servlet usando JNDI*

helder@acm.org

argonavis.com.br