Spring Framework
Module 4 – JNDI

Evgeniy Krivosheev
Vyacheslav Yakovenko
Last update: Feb, 2012

Contents

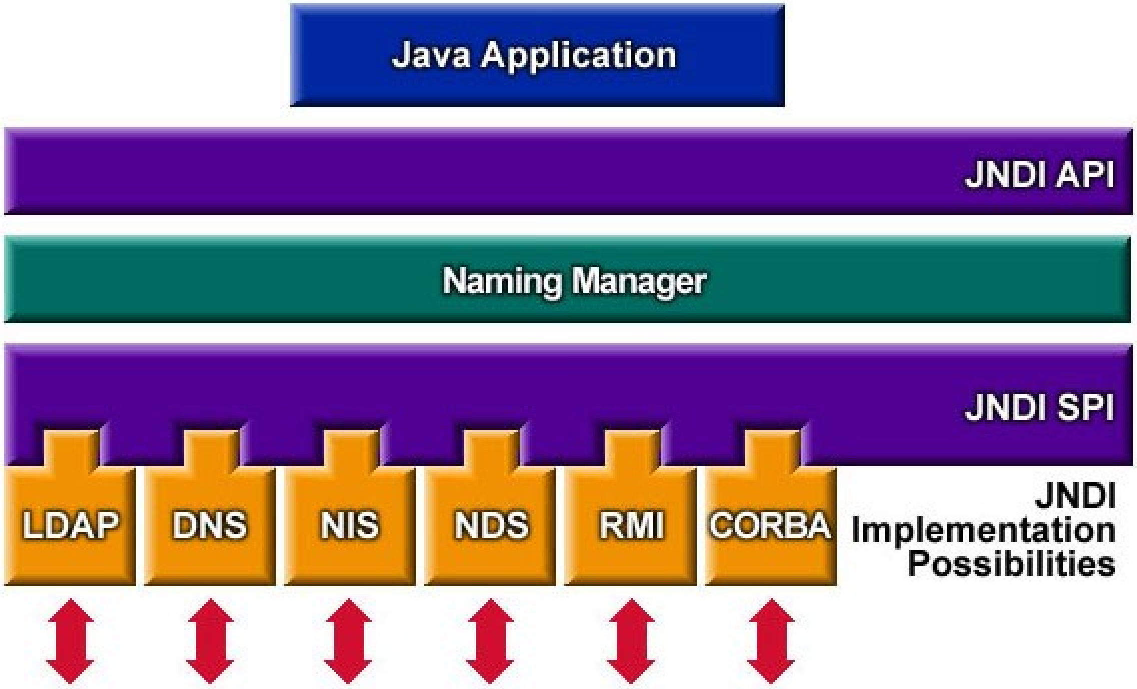
- Spring integration with JNDI

Spring :: Integration with JNDI

- Basic function of any naming service is to depict understandable object names (such as addresses, identifiers) and link them to objects that are generally used in software. For example, DNS translates computer names (www.sun.com) into IP addresses (192.9.48.5).

Spring :: Integration with JNDI

- JNDI: Java Naming and Directory Interface



Spring :: Integration with JNDI

- Generally, referencing to JNDI is needed when it is necessary to access DataSource and JtaTransactionManager. Spring provides rather simple and intuitive way of accessing them:

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:jee="http://www.springframework.org/schema/jee"
  xsi:schemaLocation=" http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
  http://www.springframework.org/schema/jee
  http://www.springframework.org/schema/jee/spring-jee-3.0.xsd">

  <jee:jndi-lookup id="dataSource" jndi-name="jdbc/jpetstore"/>

</beans>
```

Spring :: Integration with JNDI

- However, JNDI support in Spring is not limited to this;
- Main classes used for work with JNDI in Spring are:
  - org.springframework.jndi.JndiTemplate
  - Interface org.springframework.jndi.JndiCallback
  - org.springframework.jndi.JndiObjectFactoryBean

Spring :: Integration with JNDI

org.springframework.jndi.JndiTemplate:

- Responsible for object lookup, binding, rebinding, and unbind;
- Implements callback-based interface:
  - `<T> T execute(JndiCallback<T> contextCallback)`

Spring :: Integration with JNDI

org.springframework.jndi.JndiTemplate:

```
JndiTemplate template = new JndiTemplate();
…
template.bind("SomeKey", "SomeValue");
…
String value = (String)template.lookup("SomeKey");
template.rebind("SomeKey", "SomeValue");
```

Spring :: Integration with JNDI

org.springframework.jndi.JndiCallback :

- For example, when you want to log JNDI operations you have to:
  - Create new class that will implement JndiCallback;
  - Move it to execute method from JndiTemplate;

Spring :: Integration with JNDI

org.springframework.jndi.JndiCallback :

```
JndiTemplate template = new JndiTemplate();
CustomJndiAccessForLogging callback =
   new CustomJndiAccessForLogging("key");
Object result = template.execute(callback);
---
public class CustomJndiAccessForLogging implements JndiCallback {
   private String key;
   public CustomJndiAccessForLogging(String key) { this.key = key;}

   public Object doInContext(Context context) throws
   NamingException {
        System.out.println("Start lookup operation");
        Object value = context.lookup(key);
        System.out.println("End lookup operation");
        return value;
   }
}
```

Spring :: Integration with JNDI

org.springframework.jndi.JndiObjectFactoryBean :

- Responsible for data retrieval from JNDI repositiry;
- Main strategy is to look up objects and cache them;
- Can be modified either declaratively or programmatically;

Spring :: Integration with JNDI

org.springframework.jndi.JndiObjectFactoryBean :

```
<bean id=“infoFromJndi” class=“InfoFromJndi”>
  <property name=“welcomeMessage”>
          <bean class=“org.springframework.jndi.JndiObjectFactoryBean”>
          <property name=“jndiName”>
       <value>java:comp/env/welcomeMessage</value>
              </property>
      </bean>
  </property>
</bean>
```

- In InfoFromJndi bean the value for welcomeMessage field is taken from JNDI
- Attribute *value* for welcomeMessage is substituted for JndiObjectFactoryBean that gets its value from JNDI

Any questions!?