



# Spring Framework

## Module 4 – JNDI

Evgeniy Krivosheev  
Vyacheslav Yakovenko  
Last update: Feb, 2012

# Содержание

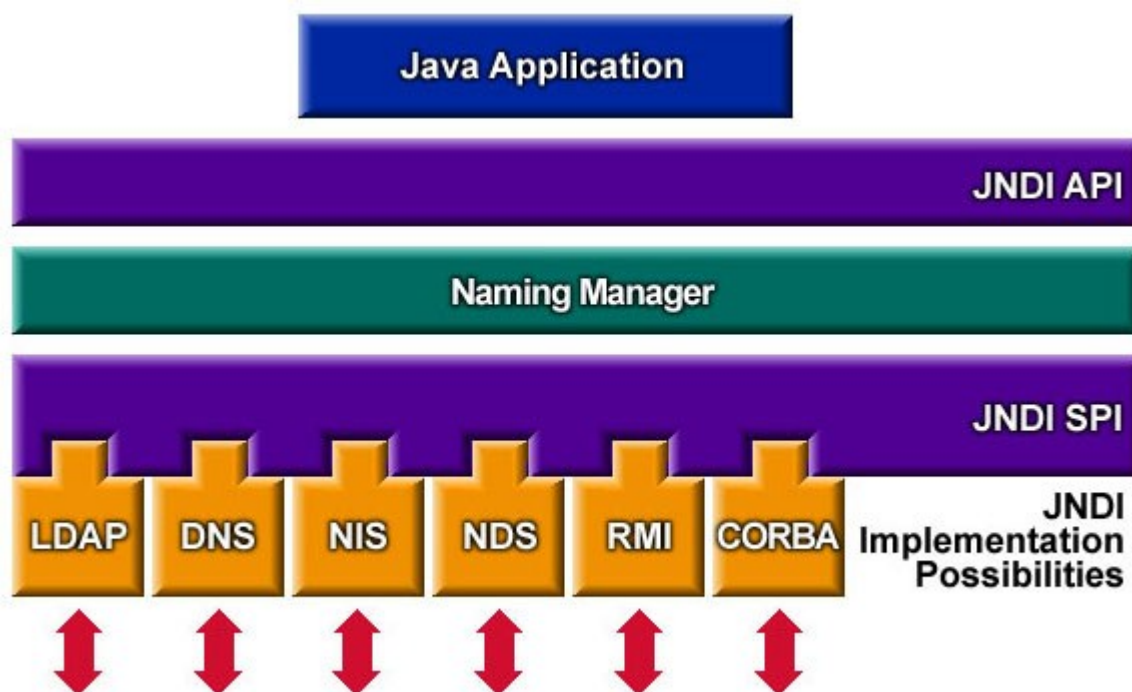
- Интеграция Spring с JNDI

# Spring :: Интеграция с JNDI

- Основной функцией любой службы имен является отображение понятных для людей имен объектов ( таких как адреса, идентификаторы ) и их связывания с объектами, которые обычно используются в ПО. Например, DNS переводит имена компьютеров (www.sun.com) в IP-адреса (192.9.48.5).

# Spring :: Интеграция с JNDI

- JNDI – Java Naming and Directory Interface



# Spring :: Интеграция с JNDI

- В большинстве случаев, в производстве обращение к JNDI требуется для получения доступа к DataSource и JtaTransactionManager. Spring предоставляет достаточно простой и интуитивно понятный вариант обращения к ним:

```
<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 :: Интеграция с JNDI

- Тем не менее поддержка JNDI в Spring, на этом не заканчивается;
- Основные классы для работы с JNDI в Spring:
  - `org.springframework.jndi.JndiTemplate`
  - Interface `org.springframework.jndi.JndiCallback`
  - `org.springframework.jndi.JndiObjectFactoryBean`

## Spring :: Интеграция с JNDI

org.springframework.jndi.JndiTemplate:

- Отвечает за операции поиска и связывания объектов (lookup, binding, rebinding, unbind);
- А также предоставляет интерфейс основанный на callback'ах:
  - `<T> T execute(JndiCallback<T> contextCallback)`

# Spring :: Интеграция с JNDI

org.springframework.jndi.JndiTemplate:

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



# Spring :: Интеграция с JNDI

`org.springframework.jndi.JndiCallback :`

- Например, если мы хотим логировать JNDI операции необходимо:
  - Создать новый класс, реализующий JndiCallback;
  - Передать его в метод `execute` из JndiTemplate;

# Spring :: Интеграция с 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 :: Интеграция с JNDI

**org.springframework.jndi.JndiObjectFactoryBean :**

- Отвечает за получение информации из JNDI репозитория;
- Основная стратегия – поиск и кэширование объектов;
- Она может быть изменена декларативно или программно;

# Spring :: Интеграция с 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>
```

- В бине InfoFromJndi значение поля welcomeMessage должно браться из JNDI
- Вместо атрибута value для свойства welcomeMessage определен JndiObjectFactoryBean – он получает значение из JNDI

# Вопросы!?

