

## Java:

технология Enterprise Java Beans 3.0

Лекция 2. Приложения на основе EJB

# Темы лекции



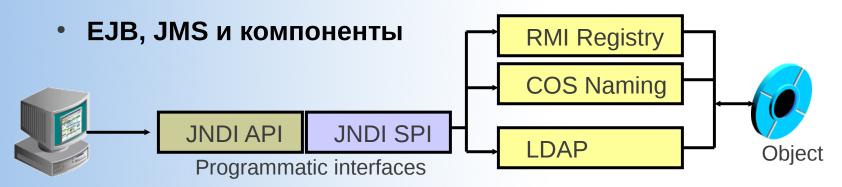
- Технология JNDI
- Регистрация ресурсов и доступ к ним
- Лабораторная работа: создание консольного приложения иллюстрирующего работу с JNDI
- Получение локальной ссылки на экземпляр компонента
- Структура Java ЕЕ приложения
- Структура deployment descriptor`a Java EE
- Развертывание приложения на сервере приложений
- Лабораторная работа: создание Java EE приложения и развертывание его в сервере приложений

## Цели лекции

- По окончинию занятия Вы должны научиться :
  - Развертывать приложение на сервере
  - Получать доступ к ресурсам и EJBs с помощью:
    - JNDI API
    - Dependency injection
  - Вызывать бины

## Доступ к русурсам через JNDI

- The Java Naming and Directory Interface (JNDI):
  - Это тандартный набор интерфейсов, которые обеспечивают
    - Сервис имен (Naming services)
    - Сервис каталогов
  - Сервис предоставляется Java EE контейнерами для поиска Java EE ресурсов или объектов, таких как:
    - DataSource
    - Resource Environment References



JNDI client

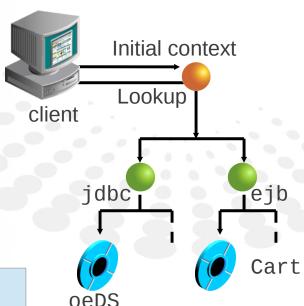
## Как обратиться к ресерсу через JNDI?

### Импортируем JNDI свойства:

- java.naming.factory.initial
- java.naming.security.principal
- java.naming.security.credentials
- java.naming.provider.url
- Устанавливаем initial context:

#### • Производим поиск:

```
import javax.sql.DataSource; ...
DataSource ds =
   (DataSource)ctx.lookup("jdbc/SRDemoDS");
```



## **JNDI** Properties

jndi.properties файл, который находится в CLASSPATH

```
java.naming.factory.initial=oracle.j2ee.rmi.RMIInitialContextFactory
java.naming.provider.url=ormi://localhost:23791/default
java.naming.security.principal=oc4jadmin
java.naming.security.credentials=welcome1
```

• Системные свойства, переданные параметром - D при запуске виртуальной машины

```
java –Djava.naming.provider.url=ormi://localhost:23791/default ...
ou.j2ee.CartClient
```

• Записи, добавленные в java.util.Hashtable, в коде приложения

Страница ¬ 6 www.specialist.ru

## **Dependency Injection**

- Технология Dependency injection основана на объявлении ресурсов с помощию аннотаций
- Исключает явные JNDI API вызовы
- Используется классами управляемых контейнером J2EE
- Не может быть использован классами внешних контейнеров Java EE
- Реализуется с помощью следующих ключевых аннотации:
  - @ Resource (javax.annotation.Resource)
  - @ EJB (javax.ejb.EJB)

## Аннотация @Resource

```
public @interface Resource {
  public enum AuthenticationType {CONTAINER, APPLICATION}
  String name() default "";
  Class type() default Object.class;
  AuthenticationType authenticationType()
    default AuthenticationType.CONTAINER;
  boolean shareable() default true;
  String mappedName default "";
  String description() default "";
}
```

- Параметры аннотации @Resource могут быть:
  - Производными (Derived)
  - Явно указаны (Explicitly specified)

Страница ¬ 8 www.specialist.ru

## Использование @Resource

- Использование ресурсов Java EE с применением dependency injection:
  - DataSource

```
@Resource(name="jdbc/oeDS")
private javax.jdbc.DataSource myDS;
Connection conn = myDS.getConnection();
```

#### - JMS

```
@Resource(name="jms/demoQueue")
private Queue myQueue;
```

### Environment entries

```
@Resource
int maxItemsPerCart = 20;
```

### Работа с DI

- Внедрение зависимостей:
  - Исключает сложности использования JNDI
  - Меньше строк кода
  - Дает более лаконичный код. Тем не менее, аннотации могут:
    - Хардкодить ресурсы
    - переопределяться с помощью XML-элементы в дескрипторах
- Не может быть использоваться с вспомогательными классами (static), там можно использовать JNDI

## **EJB** и Dependency Injection

@EJB

```
@Target({TYPE, METHOD, FIELD})
@Retention(RUNTIME)
public @interface EJB {
   String name() default "";
   String beanName() default "";
   String mappedName() default "";
   String description() default "";
   Class beanInterface() default Object.class;
}
```

• Элемент <ejb-ref> дескриптора

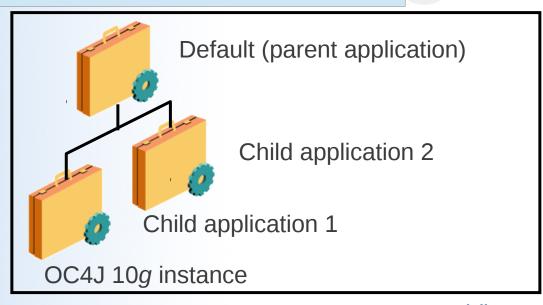
Страница ¬ 11 www.specialist.ru

## **Global JNDI lookup**

• глобального поиска JNDI, устанасливается в aттрибуте global-jndi-lookup-enabled в server.xml

```
<application-server ...
  global-jndi-lookup-enabled="true" ...>
    ...
</application-server>
```

Поиск будет осуществляться во всех развернутых приложениях



Страница ¬ 12 www.specialist.ru

## Кратко

- Что есть Java Naming и Directory Interface (JNDI)?
- Как искать ресурсы и EJBs с помощью:
  - JNDI API,
  - Внедрение зависимостей

# Ejb-jar.xml

<ejb-jar xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/j2ee

http://java.sun.com/xml/ns/javaee/ejb-jar\_3\_0.xsd"

version="3.0">

www.specialist.ru

# Ejb-jar.xml

```
<enterprise-beans>
  <session>
     <ejb-name>MyService</ejb-name>
           <env-entry>
           <env-entry-name>myProp</env-entry-name>
           <env-entry-type>java.lang.String/env-entry-type>
           <env-entry-value>Bye, </env-entry-value>
           </env-entry>
   </session>
</enterprise-beans>
```

Страница ¬ 15 www.specialist.ru



Страница ¬ 16 www.specialist.ru