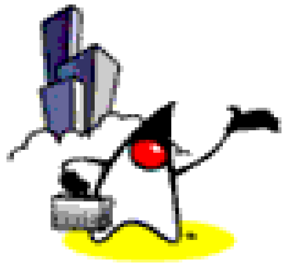
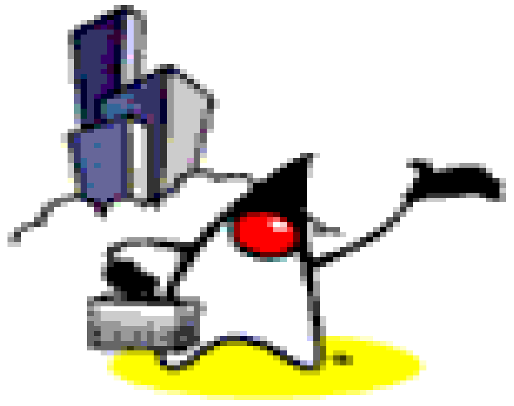
J2EE Support in Spring Framework



Topics

- JMS
- JNDI
- EJB



JMS

JmsTemplate

- Helper class that simplifies JMS access code.
  - This class requires a JMS 1.1+ provider, because it builds on the domain-independent API
  - If you want to use dynamic destination creation, you must specify the type of JMS destination to create, using the "pubSubDomain" property
  - Default settings for JMS Sessions are "not transacted" and "auto-acknowledge"
  - This template uses a DynamicDestinationResolver and a SimpleMessageConverter as default strategies for resolving a destination name or converting a message, respectively.

Configuration

```
<beans>

    <!-- configure connection factory -->
    <bean id="connectionFactory"
         class="org.apache.activemq.ActiveMQConnectionFactory">
        <property name="brokerURL">
            <value>tcp://localhost:61616</value>
        </property>
    </bean>

    <bean id="jmsTemplate"
          class="org.springframework.jms.core.JmsTemplate">
        <property name="connectionFactory">
            <ref local="connectionFactory"/>
        </property>
    </bean>

</beans>
```

Sending a Message

```
JmsTemplate template = (JmsTemplate) ctx.getBean("jmsTemplate");
Destination destination =
        (Destination) ctx.getBean("destination");

template.send(destination, new MessageCreator() {
        public Message createMessage(Session session)
            throws JMSException {
            return session.createTextMessage("Hello World");
        }
});
```

Receiving a Message

```
JmsTemplate template =
            (JmsTemplate) ctx.getBean("jmsTemplate");
Destination destination =
            (Destination) ctx.getBean("destination");

Message msg = template.receive(destination);
System.out.println("Received Message: " + msg);
```

J2EE Support in Spring Framework