Wildfly中使用JMS

首先正常的wildfly服务器是不会启动消息服务的。

那么修改改配置文件喽。

首先需要wildfly启动消息服务插件。

<extension module="org.jboss.as.messaging"/>

但是仅仅这样是不够的。需要加上这个配置

<subsystem xmlns="urn:jboss:domain:messaging:3.0">

</subsystem>

这个东西在不同版本的wildfly中配置是不一样的。

我介绍下wildfly9的配置,9用的是hornetQ的消息服务器。

<hornetq-server>

</hornetq-server>

在hornetQ-server中有五个配置

1. 启动安全验证
2. 连接器
3. 接收器
4. 连接器工厂
5. 目的地

安全验证这样配置

<security-enabled>false</security-enabled>

连接器这样配置

<connectors>

<in-vm-connector name="in-vm" server-id="0"/>

</connectors>

接收器这样配置

<acceptors>

<in-vm-acceptor name="in-vm-acceptor" server-id="0"/>

</acceptors>

连接器工厂这样配置

<jms-connection-factories>

<connection-factory name="InVmConnectionFactory">

<connectors>

<connector-ref connector-name="in-vm"/>

</connectors>

<entries>

<entry name="java:/ConnectionFactory"/>

</entries>

</connection-factory>

<pooled-connection-factory name="hornetq-ra">

<transaction mode="xa"/>

<connectors>

<connector-ref connector-name="in-vm"/>

</connectors>

<entries>

<entry name="java:/JmsXA"/>

<entry name="java:jboss/DefaultJMSConnectionFactory"/>

</entries>

</pooled-connection-factory>

</jms-connection-factories>

JMS目的地这样配置

<jms-destinations>

<jms-queue name="testQueue">

<entry name="jms/queue/test"/>

<entry name="java:jboss/exported/jms/queue/test"/>

</jms-queue>

<jms-topic name="testTopic">

<entry name="jms/topic/test"/>

<entry name="java:jboss/exported/jms/topic/test"/>

</jms-topic>

</jms-destinations>

但是仅仅这样就够了吗？

还需要在ejb子系统下配置消息驱动bean

<mdb>

<resource-adapter-ref resource-adapter-name="hornetq-ra"/>

<bean-instance-pool-ref pool-name="mdb-strict-max-pool"/>

</mdb>

好啦，服务器环境就搭建好了。

在发送消息的类中这样写

首先注入2个字段嘛

@Resource(name = "java:jboss/exported/jms/queue/test")

**private** Queue queue;

@Resource(name = "java:jboss/DefaultJMSConnectionFactory")

**private** XAQueueConnectionFactory connectionFactory;

代码这样

**try** (Connection connection = connectionFactory.createConnection()) {

Session session = connection.createSession();

TextMessage textMessage = session.createTextMessage("愛しています！");

MessageProducer producer = session.createProducer(queue);

producer.send(textMessage);

} **catch** (JMSException e) {

**throw** **new** EJBException(e);

}

对应的消息驱动BEAN这样写

@MessageDriven(activationConfig = {

@ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "java:jboss/exported/jms/queue/test"),

@ActivationConfigProperty(propertyName = "connectionFactoryLookup", propertyValue = "java:jboss/DefaultJMSConnectionFactory"),

@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue")

})

**public** **class** SimpleMDB **implements** MessageListener {

@Override

**public** **void** onMessage(Message m) {

**if** (m **instanceof** TextMessage) {

**try** {

String text = ((TextMessage) m).getText();

System.***err***.println(text);

} **catch** (JMSException e) {

}

}

}

}

那么JMS有什么难的呢？