Desenvolvimento de Aplicações Java

Plataforma Corporativa

Tutorial

Aplicação de Monitoramento

Outubro 2016

Sumário

1.Introdução

Implementar uma aplicação java corporativa que monitora os eventos publicados em um destino JMS (fila ou tópico), armazena em um banco de dados estes eventos e gera relatórios de 2 em 2 minutos em imagens SVG.

Para este fim, são utilizados:

- Bean de Mensagem
- Bean de Sessão com execução escalonada
- XML
- XSLT
- SVG
- JPA/JDBC
-

2. Método

- Criar um bean orientado a mensagens (MDB) que consome mensagens da fila de mensagens eventQueue no broker de mensagens do servidor de aplicações e que persiste esta informação em uma tabela no banco de dados. Defina a entidade que deverá ser persistida, crie a tabela e monte um DAO utilizando JPA.
- Criar um bean de sessão com um método que será executado segundo um escalonamento de tempo pré-determinado. Este bean irá criar com uma periodicidade de 5 em 5 minutos um arquivo svg para visualização gráfica do estado de funcionamento do sistema. Vai ser necessário entender como funciona a anotação @Schedule do método em questão.
- Implementar uma aplicação web que disponibilizará o svg gerado para visualização através do browser.

2.1. Registro de eventos no banco de dados

Vamos implementar o MDB que irá consumir os eventos da fila de mensagens e registra-los na base de dados.

- Crie um novo projeto de uma aplicação corporativa chamada **AppMonitor**. Deixe o Netbeans criar o módulo EJB (**AppMonitor-ejb**) e o módulo WEB (**AppMonitor-war**).
- Criar a entidade **Log.java** no módulo EJB.
- Criar a tabela no banco de dados e a sequência para geração automática de identificadores.
- No módulo EJB, crie um Bean de Mensagem chamado **ProcessadorMensagens.java**.
- Criar e configurar a unidade de persistência no módulo EJB.
- Implante e execute as aplicações. Verifique o seu comportamento.

2.2. Sumarização de dados

Vamos implementar o bean de sessão stateless que irá periodicamente consultar os eventos registrados na base de dados e criar um arquivo xml sumarizando estes registros.

- No módulo EJB da aplicação de monitoramento, crie um Bean de Sessão sem estado chamado **Sumarizador.java**.
- Note que este sumarizador irá gerar um arquivo chamado monitoramento.xml na pasta C:\Temp\dados. Você deve criar este diretório.
- Implante e execute as aplicações. Verifique o seu comportamento.

2.3. Apresentação dos dados

Para apresentação dos dados, vamos implementar na camada de apresentação um Servlet que irá ler o arquivo XML e alterar seu formato utilizando um arquivo XSLT. Desta forma, transformaremos um arquivo xml de dados de monitoramento (monitoramento.xml) em um arquivo svg que será visualizado no navegador.

- No módulo WEB da aplicação de monitoramento, crie um Servlet chamado **SVGViewServlet.java**. Não se esqueça de registrar o servlet criado no descritor de deployment (web.xml)!
- Inclua no projeto web as bibliotecas **xalan.jar**, **xercesImpl.jar** e **serializer.jar**.
- Crie o arquivo **cenario.xsl** na pasta ***C:\Temp\dados\xsl***.
- Implante e execute as aplicações. Verifique o seu comportamento.

Log.java

```
package monitor.entities;


import java.sql.Timestamp;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;


@Entity

@Table(name="tb_log")

public class Log {

     @Id

    @Column(name="log_id")

    @SequenceGenerator(name="logGenerator",
sequenceName="log_id_sequence",allocationSize=1)

    @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="logGenerator")

    private int id;


    @Column(name="timestamp")

    private Timestamp timestamp;


    @Column(name="evento")

    private String evento;


    public Log() {

    }


    public Log(int id, Timestamp timestamp, String evento) {

        this.id = id;

        this.timestamp = timestamp;

        this.evento = evento;
```

```
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public Timestamp getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(Timestamp timestamp) {
        this.timestamp = timestamp;
    }

    public String getEvento() {
        return evento;
    }

    public void setEvento(String evento) {
        this.evento = evento;
    }
}
```

TB_LOG

```
CREATE TABLE TB_LOG(
        log_id INTEGER NOT NULL,
            timestamp TIMESTAMP,
            evento VARCHAR(255),
            PRIMARY KEY (log_id) )
```

LOG_ID_SEQUENCE

```
CREATE SEQUENCE LOG_ID_SEQUENCE AS INTEGER START WITH 1 INCREMENT BY 1
NO MAXVALUE
```

persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">

  <persistence-unit name="DerbyMonitPU" transaction-type="JTA">

    <jta-data-source>java:/DerbyDS</jta-data-source>

    <class>monitor.entities.Log</class>

    <exclude-unlisted-classes>false</exclude-unlisted-classes>

    <properties>

        </properties>

  </persistence-unit>

</persistence>
```

Sumarizador.java

```
package monitor.ejbs;


import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.ejb.Stateless;
import javax.ejb.LocalBean;
import javax.ejb.Schedule;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;
import monitor.entities.Log;


@Stateless
@LocalBean
public class Sumarizador {


    @PersistenceContext(unitName = "DerbyMonitPU")
    private EntityManager em;


    @Schedule(second = "*/10", minute = "*", hour = "*", persistent = false)
    public void sumariza() {
        HashMap<String, Integer> sumario;
        sumario = new HashMap<String, Integer>();
        Query query = em.createQuery("FROM Log l");
        List<Log> list = query.getResultList();


        if (list != null) {
```

```
        for (Log l : list) {
            Integer i = sumario.get(l.getEvento());
            if (i != null) {
                i = i + 1;
                sumario.put(l.getEvento(), i);
            } else {
                sumario.put(l.getEvento(), 0);
            }
        }
    }
    publicaDados(sumario);
}

private void publicaDados(HashMap<String, Integer> hash) {
    PrintWriter writer = null;
    try {
        writer = new PrintWriter("C:/Temp/dados/monitoramento.xml", "UTF-8");

        writer.println("<cenario>");
        writer.println("<retangulos>");
        Iterator it = hash.entrySet().iterator();
        int posx = 20;
        while (it.hasNext()) {
            writer.println("<retangulo>");
            writer.println("<posx>"+posx+"</posx>");
            writer.println("<posy>20</posy>");
            writer.println("<largura>50</largura>");
            Map.Entry pair = (Map.Entry) it.next();
            System.out.println(pair.getKey() + " = " + pair.getValue());
            writer.println("<altura>"+pair.getValue()+"</altura>");
            writer.println("<item>"+pair.getKey()+ "</item>");
            writer.println("</retangulo>");
            posx = posx +70;
        }
        writer.println("</retangulos>");
```

```
            writer.println("</cenario>");
        } catch (FileNotFoundException ex) {
            Logger.getLogger(Sumarizador.class.getName()).log(Level.SEVERE, null, ex);
        } catch (UnsupportedEncodingException ex) {
            Logger.getLogger(Sumarizador.class.getName()).log(Level.SEVERE, null, ex);
        } finally{
            if (writer!=null){
            writer.close();
            }
        }
    }
}
```

SVGViewServlet.java

```
package xml.servlet;


import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.transform.Source;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.stream.StreamResult;
import javax.xml.transform.stream.StreamSource;


public class SVGViewServlet extends HttpServlet {


    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");


        String filePath = getServletContext().getRealPath("");
        System.out.println("filePath = " + filePath);
        try (PrintWriter out = response.getWriter()) {


            Source xmlSource = new StreamSource("C:\\Temp\\dados\\monitoramento.xml");
            File xslFile = new File("C:\\Temp\\dados\\xsl\\cenario.xsl");
            TransformerFactory transFact = TransformerFactory.newInstance();
            Transformer trans = transFact.newTransformer(new StreamSource(xslFile));
```

```
            ByteArrayOutputStream bos = new ByteArrayOutputStream();
            trans.transform(xmlSource, new StreamResult(bos));
            out.println(bos);

        } catch (TransformerConfigurationException ex) {
            Logger.getLogger(SVGViewServlet.class.getName()).log(Level.SEVERE, null, ex);
        } catch ( TransformerException ex){
            Logger.getLogger(SVGViewServlet.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }

}
```

cenario.xsl

```
<xsl:stylesheet
   version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns="http://www.w3.org/2000/svg">

  <xsl:output indent="yes" cdata-section-elements="style" />

  <xsl:template match="/">
    <svg>
       <xsl:for-each select="/cenario/retangulos/retangulo">
            <xsl:variable name="posicaox" select="posx"/>
            <xsl:variable name="posicaoy" select="posy"/>
            <xsl:variable name="largura_" select="largura"/>
            <xsl:variable name="altura_" select="altura"/>
            <xsl:variable name="padding" select="$posicaoy + 100 - $altura_"/>
            <rect x="{$posicaox}"
               y="{$padding}"
               height="{$altura_}"
               width="{$largura_}"
         style="fill:blue;stroke:green;stroke-width:5;fill-opacity:0.1; stroke-opacity:0.9"/>
       </xsl:for-each>
    </svg>
  </xsl:template>

</xsl:stylesheet>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">

    <servlet>

        <servlet-name>SVGViewServlet</servlet-name>

        <servlet-class>xml.servlet.SVGViewServlet</servlet-class>

    </servlet>

    <servlet-mapping>

        <servlet-name>SVGViewServlet</servlet-name>

        <url-pattern>/SVGViewServlet</url-pattern>

    </servlet-mapping>

    <session-config>

        <session-timeout>

            30

        </session-timeout>

    </session-config>

</web-app>
```