

Curso de Especialização Tecnológica

Tecnologia e Programação de Sistemas de Informação – 3ª Edição

Componente de formação: Integração de SI – Ferramentas

RSS - Really Simple SyndicationStax

Um documento RSS é um ficheiro XML que pode ser publicado como post de blog ou notícias.

O formato do ficheiro xml é definido via RSS consoante as suas especificações.

Geralmente um ficheiro RSS é fornecido por um web server, que cada cliente RSS vai ler. Este Cliente RSS analisa o ficheiro e publica-o.

Para perceber melhor o funcionamento do RSS, crie um projeto chamado de.vogella.rss

De seguida crie as seguintes classes:

```
package de.vogella.rss.model;

/*
 * Represents one RSS message
 */
public class FeedMessage {

    String title;
    String description;
    String link;
    String author;
    String guid;

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
```

```
    this.title = title;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public String getLink() {
    return link;
}

public void setLink(String link) {
    this.link = link;
}

public String getAuthor() {
    return author;
}

public void setAuthor(String author) {
    this.author = author;
}

public String getGuid() {
    return guid;
}

public void setGuid(String guid) {
    this.guid = guid;
}
```

@Override

Curso de Especialização Tecnológica

Tecnologia e Programação de Sistemas de Informação – 3ª Edição

```
public String toString() {  
    return "FeedMessage [title=" + title + ", description=" + description  
        + ", link=" + link + ", author=" + author + ", guid=" + guid  
        + "];"  
}  
}
```

```
package de.vogella.rss.model;  
  
import java.util.ArrayList;  
import java.util.List;  
  
/*  
 * Stores an RSS feed  
 */  
public class Feed {  
  
    final String title;  
    final String link;  
    final String description;  
    final String language;  
    final String copyright;  
    final String pubDate;  
  
    final List<FeedMessage> entries = new ArrayList<FeedMessage>();  
  
    public Feed(String title, String link, String description, String  
language,  
        String copyright, String pubDate) {  
        this.title = title;  

```

```
    this.link = link;
    this.description = description;
    this.language = language;
    this.copyright = copyright;
    this.pubDate = pubDate;
}

public List<FeedMessage> getMessages() {
    return entries;
}

public String getTitle() {
    return title;
}

public String getLink() {
    return link;
}

public String getDescription() {
    return description;
}

public String getLanguage() {
    return language;
}

public String getCopyright() {
    return copyright;
}

public String getPubDate() {
    return pubDate;
}

@Override
```

Curso de Especialização Tecnológica

Tecnologia e Programação de Sistemas de Informação – 3ª Edição

```
public String toString() {  
    return "Feed [copyright=" + copyright + ", description=" + description  
        + ", language=" + language + ", link=" + link + ", pubDate=" +  
        pubDate + ", title=" + title + "];"  
}  
}
```

Read RSS Feeds with Stax – Criar o XML Reader class

Criar a seguinte class para ler o ficheiro XML.

```
package de.vogella.rss.read;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.MalformedURLException;  
import java.net.URL;  
  
import javax.xml.stream.XMLEventReader;  
import javax.xml.stream.XMLInputFactory;  
import javax.xml.stream.XMLStreamException;  
import javax.xml.stream.events.Characters;  
import javax.xml.stream.events.XMLEvent;  
  
import de.vogella.rss.model.Feed;  
import de.vogella.rss.model.FeedMessage;  
  
public class RSSFeedParser {  
    static final String TITLE = "title";  
    static final String DESCRIPTION = "description";  
    static final String CHANNEL = "channel";  
    static final String LANGUAGE = "language";  
}
```

```

static final String COPYRIGHT = "copyright";
static final String LINK = "link";
static final String AUTHOR = "author";
static final String ITEM = "item";
static final String PUB_DATE = "pubDate";
static final String GUID = "guid";

final URL url;

public RSSFeedParser(String feedUrl) {
    try {
        this.url = new URL(feedUrl);
    } catch (MalformedURLException e) {
        throw new RuntimeException(e);
    }
}

public Feed readFeed() {
    Feed feed = null;
    try {
        boolean isFeedHeader = true;
        // Set header values initial to the empty string
        String description = "";
        String title = "";
        String link = "";
        String language = "";
        String copyright = "";
        String author = "";
        String pubdate = "";
        String guid = "";

        // First create a new XMLInputFactory
        XMLInputFactory inputFactory = XMLInputFactory.newInstance();
        // Setup a new eventReader
        InputStream in = read();
        XMLEventReader eventReader = inputFactory.createXMLEventReader(in);
    }
}

```

Curso de Especialização Tecnológica

Tecnologia e Programação de Sistemas de Informação – 3ª Edição

```
// read the XML document
while (eventReader.hasNext()) {
    XMLEvent event = eventReader.nextEvent();
    if (event.isStartElement()) {
        String localPart = event.asStartElement().getName()
            .getLocalPart();
        switch (localPart) {
            case ITEM:
                if (isFeedHeader) {
                    isFeedHeader = false;
                    feed = new Feed(title, link, description, language,
                        copyright, pubdate);
                }
                event = eventReader.nextEvent();
                break;
            case TITLE:
                title = getCharacterData(event, eventReader);
                break;
            case DESCRIPTION:
                description = getCharacterData(event, eventReader);
                break;
            case LINK:
                link = getCharacterData(event, eventReader);
                break;
            case GUID:
                guid = getCharacterData(event, eventReader);
                break;
            case LANGUAGE:
                language = getCharacterData(event, eventReader);
                break;
            case AUTHOR:
                author = getCharacterData(event, eventReader);
```

```

        break;
    case PUB_DATE:
        pubdate = getCharacterData(event, eventReader);
        break;
    case COPYRIGHT:
        copyright = getCharacterData(event, eventReader);
        break;
    }
} else if (event.isEndElement()) {
    if (event.asEndElement().getName().getLocalPart() == (ITEM)) {
        FeedMessage message = new FeedMessage();
        message.setAuthor(author);
        message.setDescription(description);
        message.setGuid(guid);
        message.setLink(link);
        message.setTitle(title);
        feed.getMessages().add(message);
        event = eventReader.nextEvent();
        continue;
    }
}
}
} catch (XMLStreamException e) {
    throw new RuntimeException(e);
}
return feed;
}

```

```

private String getCharacterData(XMLEvent event, XMLEventReader
eventReader)

```

```

    throws XMLStreamException {
    String result = "";
    event = eventReader.nextEvent();
    if (event instanceof Characters) {
        result = event.asCharacters().getData();
    }
}

```


Curso de Especialização Tecnológica

Tecnologia e Programação de Sistemas de Informação – 3ª Edição

```
    return result;
}

private InputStream read() {
    try {
        return url.openStream();
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
}
```

Utilizando o seguinte método main faça o teste ao código implementado.

```
package de.vogella.rss.tests;

import de.vogella.rss.model.Feed;
import de.vogella.rss.model.FeedMessage;
import de.vogella.rss.read.RSSFeedParser;

public class ReadTest {
    public static void main(String[] args) {
        RSSFeedParser parser = new
RSSFeedParser("http://www.vogella.com/article.rss");
        Feed feed = parser.readFeed();
        System.out.println(feed);
        for (FeedMessage message : feed.getMessages()) {
            System.out.println(message);
        }
    }
}
```

Write RSS Feeds with Stax - *Create writer*

Crie a seguinte *class* para escrever o ficheiro XML.

```
package de.vogella.rss.write;

import java.io.FileOutputStream;

import javax.xml.stream.XMLEventFactory;
import javax.xml.stream.XMLEventWriter;
import javax.xml.stream.XMLOutputFactory;
import javax.xml.stream.XMLStreamException;
import javax.xml.stream.events.Characters;
import javax.xml.stream.events.EndElement;
import javax.xml.stream.events.StartDocument;
import javax.xml.stream.events.StartElement;
import javax.xml.stream.events.XMLEvent;

import de.vogella.rss.model.Feed;
import de.vogella.rss.model.FeedMessage;

public class RSSFeedWriter {

    private String outputFile;
    private Feed rssfeed;

    public RSSFeedWriter(Feed rssfeed, String outputFile) {
        this.rssfeed = rssfeed;
        this.outputFile = outputFile;
    }

    public void write() throws Exception {

        // create a XMLOutputFactory
```

Curso de Especialização Tecnológica

Tecnologia e Programação de Sistemas de Informação – 3ª Edição

```
XMLOutputFactory outputFactory = XMLOutputFactory.newInstance();

// create XMLEventWriter
XMLEventWriter eventWriter = outputFactory
    .createXMLEventWriter(new FileOutputStream(outputFile));

// create a EventFactory

XMLEventFactory eventFactory = XMLEventFactory.newInstance();
XMLEvent end = eventFactory.createDTD("\n");

// create and write Start Tag

StartDocument startDocument = eventFactory.createStartDocument();

eventWriter.add(startDocument);

// create open tag
eventWriter.add(end);

StartElement rssStart = eventFactory.createStartElement("", "", "rss");
eventWriter.add(rssStart);
eventWriter.add(eventFactory.createAttribute("version", "2.0"));
eventWriter.add(end);

eventWriter.add(eventFactory.createStartElement("", "", "channel"));
eventWriter.add(end);

// write the different nodes

createNode(eventWriter, "title", rssfeed.getTitle());
```

```

createNode(eventWriter, "link", rssfeed.getLink());

createNode(eventWriter, "description", rssfeed.getDescription());

createNode(eventWriter, "language", rssfeed.getLanguage());

createNode(eventWriter, "copyright", rssfeed.getCopyright());

createNode(eventWriter, "pubdate", rssfeed.getPubDate());

for (FeedMessage entry : rssfeed.getMessages()) {
    eventWriter.add(eventFactory.createStartElement("", "", "item"));
    eventWriter.add(end);
    createNode(eventWriter, "title", entry.getTitle());
    createNode(eventWriter, "description", entry.getDescription());
    createNode(eventWriter, "link", entry.getLink());
    createNode(eventWriter, "author", entry.getAuthor());
    createNode(eventWriter, "guid", entry.getGuid());
    eventWriter.add(end);
    eventWriter.add(eventFactory.createEndElement("", "", "item"));
    eventWriter.add(end);
}

eventWriter.add(end);
eventWriter.add(eventFactory.createEndElement("", "", "channel"));
eventWriter.add(end);
eventWriter.add(eventFactory.createEndElement("", "", "rss"));

eventWriter.add(end);

eventWriter.add(eventFactory.createEndDocument());

eventWriter.close();
}

```

Curso de Especialização Tecnológica

Tecnologia e Programação de Sistemas de Informação – 3ª Edição

```
private void createNode(XMLEventWriter eventWriter, String name,
String value) throws XMLStreamException {
    XMLEventFactory eventFactory = XMLEventFactory.newInstance();
    XMLEvent end = eventFactory.createDTD("\n");
    XMLEvent tab = eventFactory.createDTD("\t");
    // create Start node
    StartElement sElement = eventFactory.createStartElement("", "", name);
    eventWriter.add(tab);
    eventWriter.add(sElement);
    // create Content
    Characters characters = eventFactory.createCharacters(value);
    eventWriter.add(characters);
    // create End node
    EndElement eElement = eventFactory.createEndElement("", "", name);
    eventWriter.add(eElement);
    eventWriter.add(end);
}
}
```

Para testar o código anteriormente implementado, corra o programa para gerar o ficheiro articles.rss

```
package de.vogella.rss.tests;

import java.text.SimpleDateFormat;
```

```

import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.Locale;

import de.vogella.rss.model.Feed;
import de.vogella.rss.model.FeedMessage;
import de.vogella.rss.write.RSSFeedWriter;

public class WriteTest {

    public static void main(String[] args) {
        // create the rss feed
        String copyright = "Copyright hold by Lars Vogel";
        String title = "Eclipse and Java Information";
        String description = "Eclipse and Java Information";
        String language = "en";
        String link = "http://www.vogella.com";
        Calendar cal = new GregorianCalendar();
        Date creationDate = cal.getTime();
        SimpleDateFormat date_format = new SimpleDateFormat("EEE', 'dd' 'MMM'
'yyyy' 'HH:mm:ss' 'Z", Locale.US);
        String pubdate = date_format.format(creationDate);
        Feed rssFeeder = new Feed(title, link, description, language,
            copyright, pubdate);

        // now add one example entry
        FeedMessage feed = new FeedMessage();
        feed.setTitle("RSSFeed");
        feed.setDescription("This is a description");
        feed.setAuthor("nonsense@somewhere.de (Lars Vogel)");
        feed.setGuid("http://www.vogella.com/tutorials/RSSFeed/article.html");
        feed.setLink("http://www.vogella.com/tutorials/RSSFeed/article.html");
        rssFeeder.getMessages().add(feed);

        // now write the file
    }
}

```

Curso de Especialização Tecnológica

Tecnologia e Programação de Sistemas de Informação – 3ª Edição

```
RSSFeedWriter writer = new RSSFeedWriter(rssFeeder, "articles.rss");
try {
    writer.write();
} catch (Exception e) {
    e.printStackTrace();
}
}
```

Bom trabalho!