/\*\*

\* Licensed to the Apache Software Foundation (ASF) under one or more

\* contributor license agreements. See the NOTICE file distributed with

\* this work for additional information regarding copyright ownership.

\* The ASF licenses this file to You under the Apache License, Version 2.0

\* (the "License"); you may not use this file except in compliance with

\* the License. You may obtain a copy of the License at

\*

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

package camelinaction;

import java.net.URL;

import java.net.URLConnection;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import org.apache.camel.CamelContext;

import org.apache.camel.Exchange;

import org.apache.camel.Processor;

import org.apache.camel.builder.RouteBuilder;

import org.apache.camel.impl.DefaultCamelContext;

import javax.jms.ConnectionFactory;

import org.apache.activemq.ActiveMQConnectionFactory;

import org.apache.camel.component.file.FileComponent;

import org.apache.camel.component.jms.JmsComponent;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

public class IncidentReportProducer {

public static void main(String args[]) throws Exception {

// create CamelContext

CamelContext context = new DefaultCamelContext();

// connect to ActiveMQ JMS broker listening on localhost on port 61616

ConnectionFactory connectionFactory = new ActiveMQConnectionFactory("tcp://localhost:61616");

context.addComponent("jms", JmsComponent.jmsComponentAutoAcknowledge(connectionFactory));

// add our route to the CamelContext

context.addRoutes(new RouteBuilder() {

public void configure() {

String uchicagoIncidentReportURL = "https://incidentreports.uchicago.edu/rss.php";

from("rss:" + uchicagoIncidentReportURL + "?filter=false&throttleEntries=false")

.marshal().rss()

.setBody(xpath("/rss/channel/item/link/text()"))

.to("file:data/outbox")

.to("jms:queue:RSS\_UCHICAGO\_INCIDENT\_REPORTS");

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

from("jms:queue:RSS\_UCHICAGO\_INCIDENT\_REPORTS")

.process(new Processor() {

@Override

public void process(Exchange exchange) throws Exception {

String url = exchange.getIn().getBody(String.class);

String content = null;

URLConnection connection = null;

try {

connection = new URL(url).openConnection();

Scanner scanner = new Scanner(connection.getInputStream());

scanner.useDelimiter("\\Z");

content = scanner.next();

}catch ( Exception ex ) {

ex.printStackTrace();

}

exchange.getIn().setBody(content);

}

})

.unmarshal()

.tidyMarkup()

.to("file:data/html");

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

from("file:data/html")

.process(new Processor() {

@Override

public void process(Exchange exchange) throws Exception {

String html = exchange.getIn().getBody(String.class);

Document doc = Jsoup.parse(html);

Element table = doc.select("table[class=table ucpd]").first();

Element body = table.select("tbody").first();

Elements rows = body.select("tr");

for(Element row : rows) {

Elements data = row.select("td");

StringBuilder stringBuilder = new StringBuilder();

String n = System.getProperty("line.separator");

stringBuilder.append("INCIDENT:" + data.get(0).text() + n);

stringBuilder.append("LOCATION:" + data.get(1).text() + n);

stringBuilder.append("REPORTED:" + data.get(2).text() + n);

stringBuilder.append("OCCURRED:" + data.get(3).text() + n);

stringBuilder.append("COMMENTS:" + data.get(4).text() + n);

stringBuilder.append("DISPOSITION:" + data.get(5).text() + n);

stringBuilder.append("UCPDI:" + data.get(6).text() + n);

exchange.getOut().setBody(stringBuilder.toString());

}

}

})

.to("file:data/reports");

}

});

// start the route and let it do its work

context.start();

Thread.sleep(10000);

// stop the CamelContext

context.stop();

}

}