package com.twitter.search.ingester.util.jndi;

import java.util.Hashtable;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NameNotFoundException;

import org.apache.naming.config.XmlConfigurator;

public abstract class JndiUtil {

// This is different from the search repo---twitter-naming-devtest.xml is

// checked in as a resource in src/resources/com/twitter/search/ingester.

public static final String DEFAULT\_JNDI\_XML =

System.getProperty("jndiXml", "/com/twitter/search/ingester/twitter-naming-devtest.xml");

protected static String jndiXml = DEFAULT\_JNDI\_XML;

protected static boolean testingMode = false;

static {

System.setProperty("javax.xml.parsers.SAXParserFactory",

"org.apache.xerces.jaxp.SAXParserFactoryImpl");

System.setProperty("javax.xml.parsers.DocumentBuilderFactory",

"com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl");

}

public static void loadJNDI() {

loadJNDI(jndiXml);

}

protected static void loadJNDI(String jndiXmlFile) {

try {

Hashtable<String, String> props = new Hashtable<>();

props.put(Context.INITIAL\_CONTEXT\_FACTORY, "org.apache.naming.java.javaURLContextFactory");

Context jndiContext = new InitialContext(props);

try {

jndiContext.lookup("java:comp");

setTestingModeFromJndiContext(jndiContext);

} catch (NameNotFoundException e) {

// No context.

XmlConfigurator.loadConfiguration(JndiUtil.class.getResourceAsStream(jndiXmlFile));

}

} catch (Exception e) {

throw new RuntimeException(String.format("Failed to load JNDI configuration file=%s %s",

jndiXmlFile, e.getMessage()), e);

}

}

public static void setJndiXml(String jndiXml) {

JndiUtil.jndiXml = jndiXml;

}

public static String getJndiXml() {

return jndiXml;

}

public static void setTestingMode(Boolean testingMode) {

JndiUtil.testingMode = testingMode;

}

public static boolean isTestingMode() {

return testingMode;

}

private static void setTestingModeFromJndiContext(Context jndiContext) {

try {

setTestingMode((Boolean) jndiContext.lookup("java:comp/env/testingMode"));

} catch (Exception e) {

setTestingMode(false);

}

}

}