001/\*  
002 \* Licensed to the Apache Software Foundation (ASF) under one or more  
003 \* contributor license agreements. See the NOTICE file distributed with  
004 \* this work for additional information regarding copyright ownership.  
005 \* The ASF licenses this file to You under the Apache License, Version 2.0  
006 \* (the "License"); you may not use this file except in compliance with  
007 \* the License. You may obtain a copy of the License at  
008 \*  
009 \* http://www.apache.org/licenses/LICENSE-2.0  
010 \*  
011 \* Unless required by applicable law or agreed to in writing, software  
012 \* distributed under the License is distributed on an "AS IS" BASIS,  
013 \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
014 \* See the License for the specific language governing permissions and  
015 \* limitations under the License.  
016 \*/  
017package org.apache.commons.beanutils.converters;  
018  
019import java.sql.Timestamp;  
020import java.text.DateFormat;  
021import java.util.Locale;  
022import java.util.TimeZone;  
023  
024/\*\*  
025 \* {@link DateTimeConverter} implementation that handles conversion to  
026 \* and from <b>java.sql.Timestamp</b> objects.  
027 \* <p>  
028 \* This implementation can be configured to handle conversion either  
029 \* by using java.sql.Timestamp's default String conversion, or by using a  
030 \* Locale's default format or by specifying a set of format patterns.  
031 \* See the {@link DateTimeConverter} documentation for further details.  
032 \* <p>  
033 \* Can be configured to either return a <i>default value</i> or throw a  
034 \* <code>ConversionException</code> if a conversion error occurs.  
035 \*  
036 \* @version $Id$  
037 \* @since 1.3  
038 \*/  
039public final class SqlTimestampConverter extends DateTimeConverter {  
040  
041 /\*\*  
042 \* Construct a <b>java.sql.Timestamp</b> <i>Converter</i> that throws  
043 \* a <code>ConversionException</code> if an error occurs.  
044 \*/  
045 public SqlTimestampConverter() {  
046 super();  
047 }  
048  
049 /\*\*  
050 \* Construct a <b>java.sql.Timestamp</b> <i>Converter</i> that returns  
051 \* a default value if an error occurs.  
052 \*  
053 \* @param defaultValue The default value to be returned  
054 \* if the value to be converted is missing or an error  
055 \* occurs converting the value.  
056 \*/  
057 public SqlTimestampConverter(final Object defaultValue) {  
058 super(defaultValue);  
059 }  
060  
061 /\*\*  
062 \* Return the default type this <code>Converter</code> handles.  
063 \*  
064 \* @return The default type this <code>Converter</code> handles.  
065 \* @since 1.8.0  
066 \*/  
067 @Override  
068 protected Class<?> getDefaultType() {  
069 return Timestamp.class;  
070 }  
071  
072 /\*\*  
073 \* Return a <code>DateFormat<code> for the Locale.  
074 \* @param locale TODO  
075 \* @param timeZone TODO  
076 \*  
077 \* @return The DateFormat.  
078 \* @since 1.8.0  
079 \*/  
080 @Override  
081 protected DateFormat getFormat(final Locale locale, final TimeZone timeZone) {  
082 DateFormat format = null;  
083 if (locale == null) {  
084 format = DateFormat.getDateTimeInstance(DateFormat.SHORT, DateFormat.SHORT);  
085 } else {  
086 format = DateFormat.getDateTimeInstance(DateFormat.SHORT, DateFormat.SHORT, locale);  
087 }  
088 if (timeZone != null) {  
089 format.setTimeZone(timeZone);  
090 }  
091 return format;  
092 }  
093}