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;  
018  
019/\*\*  
020 \* <p>Decorates a {@link DynaBean} to provide <code>Map</code> behavior.</p>  
021 \*  
022 \* <p>The motivation for this implementation is to provide access to {@link DynaBean}  
023 \* properties in technologies that are unaware of BeanUtils and {@link DynaBean}s -  
024 \* such as the expression languages of JSTL and JSF.</p>  
025 \*  
026 \* <p>This can be achieved either by wrapping the {@link DynaBean} prior to  
027 \* providing it to the technology to process or by providing a <code>Map</code>  
028 \* accessor method on the DynaBean implementation:  
029 \* <pre><code>  
030 \* public Map<String, Object> getMap() {  
031 \* return new DynaBeanPropertyMapDecorator(this);  
032 \* }</code></pre>  
033 \* </ul>  
034 \* </p>  
035 \*  
036 \* <p>This, for example, could be used in JSTL in the following way to access  
037 \* a DynaBean's <code>fooProperty</code>:  
038 \* <ul><li><code>${myDynaBean.<b>map</b>.fooProperty}</code></li></ul>  
039 \* </p>  
040 \*  
041 \* <h3>Usage</h3>  
042 \*  
043 \* <p>To decorate a {@link DynaBean} simply instantiate this class with the  
044 \* target {@link DynaBean}:</p>  
045 \*  
046 \* <ul><li><code>Map<String, Object> fooMap = new DynaBeanPropertyMapDecorator(fooDynaBean);</code></li></ul>  
047 \*  
048 \* <p>The above example creates a <b><i>read only</i></b> <code>Map</code>.  
049 \* To create a <code>Map</code> which can be modified, construct a  
050 \* <code>DynaBeanPropertyMapDecorator</code> with the <b><i>read only</i></b>  
051 \* attribute set to <code>false</code>:</p>  
052 \*  
053 \* <ul><li><code>Map<String, Object> fooMap =  
054 \* new DynaBeanPropertyMapDecorator(fooDynaBean, false);</code></li></ul>  
055 \*  
056 \* <h3>Limitations</h3>  
057 \* <p>In this implementation the <code>entrySet()</code>, <code>keySet()</code>  
058 \* and <code>values()</code> methods create an <b><i>unmodifiable</i></b>  
059 \* <code>Set</code> and it does not support the Map's <code>clear()</code>  
060 \* and <code>remove()</code> operations.</p>  
061 \*  
062 \* @since BeanUtils 1.9.0  
063 \* @version $Id$  
064 \*/  
065public class DynaBeanPropertyMapDecorator extends BaseDynaBeanMapDecorator<String> {  
066 /\*\*  
067 \* Construct a Map for the specified {@link DynaBean}.  
068 \*  
069 \* @param dynaBean The dyna bean being decorated  
070 \* @param readOnly <code>true</code> if the Map is read only  
071 \* otherwise <code>false</code>  
072 \* @throws IllegalArgumentException if the {@link DynaBean} is null.  
073 \*/  
074 public DynaBeanPropertyMapDecorator(final DynaBean dynaBean, final boolean readOnly) {  
075 super(dynaBean, readOnly);  
076 }  
077  
078 /\*\*  
079 \* Constructs a read only Map for the specified  
080 \* {@link DynaBean}.  
081 \*  
082 \* @param dynaBean The dyna bean being decorated  
083 \* @throws IllegalArgumentException if the {@link DynaBean} is null.  
084 \*/  
085 public DynaBeanPropertyMapDecorator(final DynaBean dynaBean) {  
086 super(dynaBean);  
087 }  
088  
089 @Override  
090 protected String convertKey(final String propertyName) {  
091 return propertyName;  
092 }  
093}