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 \*/  
017  
018package org.apache.commons.beanutils;  
019  
020  
021  
022  
023  
024/\*\*  
025 \* <p>A specialized extension to <code>DynaClass</code> that allows properties  
026 \* to be added or removed dynamically.</p>  
027 \*  
028 \* <p><strong>WARNING</strong> - No guarantees that this will be in the final  
029 \* APIs ... it's here primarily to preserve some concepts that were in the  
030 \* original proposal for further discussion.</p>  
031 \*  
032 \* @version $Id$  
033 \*/  
034  
035public interface MutableDynaClass extends DynaClass {  
036  
037  
038 /\*\*  
039 \* Add a new dynamic property with no restrictions on data type,  
040 \* readability, or writeability.  
041 \*  
042 \* @param name Name of the new dynamic property  
043 \*  
044 \* @throws IllegalArgumentException if name is null  
045 \* @throws IllegalStateException if this DynaClass is currently  
046 \* restricted, so no new properties can be added  
047 \*/  
048 public void add(String name);  
049  
050  
051 /\*\*  
052 \* Add a new dynamic property with the specified data type, but with  
053 \* no restrictions on readability or writeability.  
054 \*  
055 \* @param name Name of the new dynamic property  
056 \* @param type Data type of the new dynamic property (null for no  
057 \* restrictions)  
058 \*  
059 \* @throws IllegalArgumentException if name is null  
060 \* @throws IllegalStateException if this DynaClass is currently  
061 \* restricted, so no new properties can be added  
062 \*/  
063 public void add(String name, Class<?> type);  
064  
065  
066 /\*\*  
067 \* Add a new dynamic property with the specified data type, readability,  
068 \* and writeability.  
069 \*  
070 \* @param name Name of the new dynamic property  
071 \* @param type Data type of the new dynamic property (null for no  
072 \* restrictions)  
073 \* @param readable Set to <code>true</code> if this property value  
074 \* should be readable  
075 \* @param writeable Set to <code>true</code> if this property value  
076 \* should be writeable  
077 \*  
078 \* @throws IllegalArgumentException if name is null  
079 \* @throws IllegalStateException if this DynaClass is currently  
080 \* restricted, so no new properties can be added  
081 \*/  
082 public void add(String name, Class<?> type, boolean readable,  
083 boolean writeable);  
084  
085  
086 /\*\*  
087 \* Is this DynaClass currently restricted, if so, no changes to the  
088 \* existing registration of property names, data types, readability, or  
089 \* writeability are allowed.  
090 \*  
091 \* @return <code>true</code> if this Mutable {@link DynaClass} is restricted,  
092 \* otherwise <code>false</code>  
093 \*/  
094 public boolean isRestricted();  
095  
096  
097 /\*\*  
098 \* Remove the specified dynamic property, and any associated data type,  
099 \* readability, and writeability, from this dynamic class.  
100 \* <strong>NOTE</strong> - This does <strong>NOT</strong> cause any  
101 \* corresponding property values to be removed from DynaBean instances  
102 \* associated with this DynaClass.  
103 \*  
104 \* @param name Name of the dynamic property to remove  
105 \*  
106 \* @throws IllegalArgumentException if name is null  
107 \* @throws IllegalStateException if this DynaClass is currently  
108 \* restricted, so no properties can be removed  
109 \*/  
110 public void remove(String name);  
111  
112  
113 /\*\*  
114 \* Set the restricted state of this DynaClass to the specified value.  
115 \*  
116 \* @param restricted The new restricted state  
117 \*/  
118 public void setRestricted(boolean restricted);  
119  
120  
121}