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  
019import java.beans.IntrospectionException;  
020import java.util.Collection;  
021import java.util.Collections;  
022import java.util.HashSet;  
023import java.util.Set;  
024  
025/\*\*  
026 \* <p>  
027 \* A specialized {@code BeanIntrospector} implementation which suppresses some properties.  
028 \* </p>  
029 \* <p>  
030 \* An instance of this class is passed a set with the names of the properties it should  
031 \* process. During introspection of a bean class it removes all these properties from the  
032 \* {@link IntrospectionContext}. So effectively, properties added by a different  
033 \* {@code BeanIntrospector} are removed again.  
034 \* </p>  
035 \*  
036 \* @version $Id$  
037 \* @since 1.9.2  
038 \*/  
039public class SuppressPropertiesBeanIntrospector implements BeanIntrospector {  
040 /\*\*  
041 \* A specialized instance which is configured to suppress the special {@code class}  
042 \* properties of Java beans. Unintended access to the property {@code class} (which is  
043 \* common to all Java objects) can be a security risk because it also allows access to  
044 \* the class loader. Adding this instance as {@code BeanIntrospector} to an instance  
045 \* of {@code PropertyUtilsBean} suppresses the {@code class} property; it can then no  
046 \* longer be accessed.  
047 \*/  
048 public static final SuppressPropertiesBeanIntrospector SUPPRESS\_CLASS =  
049 new SuppressPropertiesBeanIntrospector(Collections.singleton("class"));  
050  
051 /\*\* A set with the names of the properties to be suppressed. \*/  
052 private final Set<String> propertyNames;  
053  
054 /\*\*  
055 \* Creates a new instance of {@code SuppressPropertiesBeanIntrospector} and sets the  
056 \* names of the properties to be suppressed.  
057 \*  
058 \* @param propertiesToSuppress the names of the properties to be suppressed (must not  
059 \* be <b>null</b>)  
060 \* @throws IllegalArgumentException if the collection with property names is  
061 \* <b>null</b>  
062 \*/  
063 public SuppressPropertiesBeanIntrospector(final Collection<String> propertiesToSuppress) {  
064 if (propertiesToSuppress == null) {  
065 throw new IllegalArgumentException("Property names must not be null!");  
066 }  
067  
068 propertyNames = Collections.unmodifiableSet(new HashSet<String>(  
069 propertiesToSuppress));  
070 }  
071  
072 /\*\*  
073 \* Returns a (unmodifiable) set with the names of the properties which are suppressed  
074 \* by this {@code BeanIntrospector}.  
075 \*  
076 \* @return a set with the names of the suppressed properties  
077 \*/  
078 public Set<String> getSuppressedProperties() {  
079 return propertyNames;  
080 }  
081  
082 /\*\*  
083 \* {@inheritDoc} This implementation removes all properties from the given context it  
084 \* is configured for.  
085 \*/  
086 public void introspect(final IntrospectionContext icontext) throws IntrospectionException {  
087 for (final String property : getSuppressedProperties()) {  
088 icontext.removePropertyDescriptor(property);  
089 }  
090 }  
091}