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;  
020  
021/\*\*  
022 \* <p>  
023 \* Definition of an interface for components that can perform introspection on  
024 \* bean classes.  
025 \* </p>  
026 \* <p>  
027 \* Before {@link PropertyUtils} can be used for interaction with a specific Java  
028 \* class, the class's properties have to be determined. This is called  
029 \* <em>introspection</em> and is initiated automatically on demand.  
030 \* <code>PropertyUtils</code> does not perform introspection on its own, but  
031 \* delegates this task to one or more objects implementing this interface. This  
032 \* makes it possible to customize introspection which may be useful for certain  
033 \* code bases using non-standard conventions for accessing properties.  
034 \* </p>  
035 \*  
036 \* @version $Id$  
037 \* @since 1.9  
038 \*/  
039public interface BeanIntrospector {  
040 /\*\*  
041 \* Performs introspection on a Java class. The current class to be inspected  
042 \* can be queried from the passed in <code>IntrospectionContext</code>  
043 \* object. A typical implementation has to obtain this class, determine its  
044 \* properties according to the rules it implements, and add them to the  
045 \* passed in context object.  
046 \*  
047 \* @param icontext the context object for interaction with the initiator of  
048 \* the introspection request  
049 \* @throws IntrospectionException if an error occurs during introspection  
050 \*/  
051 void introspect(IntrospectionContext icontext)  
052 throws IntrospectionException;  
053}