

ExtractSupertypeRefactoringContribution.java – ExtractInterfaceRefactoringContribution.java

```

1/*****
2 * Copyright (c) 2005, 2008 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation – initial API and implementation
10 *****/
11 package org.eclipse.jdt.internal.corext.refactoring.scripting;
12
13 import java.util.Map;
14
15 import org.eclipse.core.runtime.CoreException;
16
17 import org.eclipse.ltk.core.refactoring.Refactoring;
18 import org.eclipse.ltk.core.refactoring.RefactoringDescriptor;
19 import org.eclipse.ltk.core.refactoring.RefactoringStatus;
20 import org.eclipse.ltk.core.refactoring.participants.ProcessorBasedRefactoring;
21
22 import org.eclipse.jdt.core.refactoring.descriptors.JavaRefactoringDescriptor;
23
24 import org.eclipse.jdt.internal.core.refactoring.descriptors.RefactoringSignatureDescriptor
Factory;
25 import org.eclipse.jdt.internal.corext.refactoring.JavaRefactoringArguments;
26 import org.eclipse.jdt.internal.corext.refactoring.structure.ExtractSupertypeProcessor;
27
28 /**
29 * Refactoring contribution for the extract supertype refactoring.
30 *
31 * @since 3.2
32 */
33 public final class ExtractSupertypeRefactoringContribution extends
JavaUIRefactoringContribution {
34
35     /**
36      * {@inheritDoc}
37      */
38     public final Refactoring createRefactoring(JavaRefactoringDescriptor
descriptor, RefactoringStatus status) throws CoreException {
39         JavaRefactoringArguments arguments= new
JavaRefactoringArguments(descriptor.getProject(), retrieveArgumentMap(descriptor));
40         ExtractSupertypeProcessor processor= new
ExtractSupertypeProcessor(arguments, status);
41         return new ProcessorBasedRefactoring(processor);
42     }
43
44     public RefactoringDescriptor createDescriptor() {
45         return
RefactoringSignatureDescriptorFactory.createExtractSuperclassDescriptor();
46     }
47
48     public RefactoringDescriptor createDescriptor(String id, String project, String

```

```

1/*****
2 * Copyright (c) 2005, 2007 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation – initial API and implementation
10 *****/
11 package org.eclipse.jdt.internal.corext.refactoring.scripting;
12
13 import java.util.Map;
14
15 import org.eclipse.core.runtime.CoreException;
16
17 import org.eclipse.ltk.core.refactoring.Refactoring;
18 import org.eclipse.ltk.core.refactoring.RefactoringDescriptor;
19 import org.eclipse.ltk.core.refactoring.RefactoringStatus;
20 import org.eclipse.ltk.core.refactoring.participants.ProcessorBasedRefactoring;
21
22 import org.eclipse.jdt.core.refactoring.descriptors.JavaRefactoringDescriptor;
23
24 import org.eclipse.jdt.internal.core.refactoring.descriptors.RefactoringSignatureDescriptor
Factory;
25 import org.eclipse.jdt.internal.corext.refactoring.JavaRefactoringArguments;
26 import org.eclipse.jdt.internal.corext.refactoring.structure.ExtractInterfaceProcessor;
27
28 /**
29 * Refactoring contribution for the extract interface refactoring.
30 *
31 * @since 3.2
32 */
33 public final class ExtractInterfaceRefactoringContribution extends
JavaUIRefactoringContribution {
34
35     /**
36      * {@inheritDoc}
37      */
38     public final Refactoring createRefactoring(JavaRefactoringDescriptor
descriptor, RefactoringStatus status) throws CoreException {
39         JavaRefactoringArguments arguments= new
JavaRefactoringArguments(descriptor.getProject(), retrieveArgumentMap(descriptor));
40         ExtractInterfaceProcessor processor= new
ExtractInterfaceProcessor(arguments, status);
41         return new ProcessorBasedRefactoring(processor);
42     }
43
44     public RefactoringDescriptor createDescriptor() {
45         return
RefactoringSignatureDescriptorFactory.createExtractInterfaceDescriptor();
46     }
47
48     public RefactoringDescriptor createDescriptor(String id, String project, String

```

ExtractSupertypeRefactoringContribution.java - ExtractInterfaceRefactoringContribution.java

```
48description, String comment, Map arguments, int flags) {  
49    return  
    RefactoringSignatureDescriptorFactory.createExtractSuperclassDescriptor(project,  
    description, comment, arguments, flags);  
50}  
51}  
52
```

```
48description, String comment, Map arguments, int flags) {  
49    return  
    RefactoringSignatureDescriptorFactory.createExtractInterfaceDescriptor(project,  
    description, comment, arguments, flags);  
50}  
51}  
52
```