

Extract Interface

With the **Extract Interface** refactoring you have three options:

- Create an interface based on the methods of a class without applying the new interface immediately.
- Create an interface and apply it to the source code.
- Rename the original class, and it implements the newly created interface. In such case, IntelliJ IDEA changes all usages of the original class to use the interface where possible.

In addition, static final fields, declared in the initial class, can be moved to an interface. As a result, an interface will be created containing the specific methods and fields. Thereby, the specified class methods become implementations of the corresponding interface methods.

- [Examples](#)
- [Extracting an interface](#)

Examples

Here we have a class, and perform **Extract Interface** refactoring to create an interface based on the methods of the class.

Before	After
<pre>// File AClass.java class AClass { public static final double CONSTANT=3.14; public void publicMethod() { //some code here } public void secretMethod() { //some code here } }</pre>	<pre>// File AClass.java class AClass implements AnInterface { public void publicMethod() { //some code here } public void secretMethod() { //some code here } } // File AnInterface.java public interface AnInterface { double CONSTANT=3.14; void publicMethod(); }</pre>

Another example of the **Extract Interface** refactoring, when the **Rename original class and use interface where possible** option is selected.

Before	After
<pre>public class FormerAClass implements AClass { public void publicMethod() { //some code here } public void secretMethod() { //some code here } }</pre>	<pre>public interface AClass { double CONSTANT=3.14; void publicMethod(); }</pre>

You can extract an interface from the class that already implements another interface. Let's extract interface from the class that implements `AnInterface`. Depending on whether we want `AnotherInterface` (extracted interface) to extend the `AnInterface` (existing one) or we want source `AClass` to implement them both, we will get the following code:

Extracted Interface extends the existing one Source class:

<pre>class AClass implements AnotherInterface { public void publicMethod() { //some code here } public void secretMethod() { //some code here } }</pre>

Extracted Interface:

<pre>public interface AnotherInterface extends AnInterface { }</pre>
--

Source class implements both interfaces.

Source class:

```
class AClass implements AnInterface, AnotherInterface {
    public void publicMethod() {
        //some code here
    }
    public void secretMethod() {
        //some code here
    }
}
```

Extracted Interface:

```
public interface AnotherInterface {
}
```

Extracting an interface

1. Select a class in the Project view, Structure view, Commander, or place the caret anywhere within a class in the text editor.
2. On the main menu or on the context menu of the selection, choose **Refactor | Extract | Interface**. The **Extract Interface** dialog box appears.
3. To extract a new interface, select the **Extract Interface** option and specify the name for the new interface.
To rename the original class and make it an implementation of the newly created interface, select the **Rename original class and use interface where possible** option and specify the new name for the original class. IntelliJ IDEA will alter all original class usages to the usages of the implementing only where it is still necessary.
4. Specify the package, where the new interface will be located.
5. Select the class members you want to be listed in the interface in the **Members to Form Interface** area. The list shows all the methods of the class, as well as final static fields (constants).
6. In the **JavaDoc** area, select the action to be applied on the **JavaDoc**.
 - To leave it where it is, select the **As is** option.
 - To copy it to the extracted interface, select the **Copy** option.
 - To move it to the extracted interface, select the **Move** option.
7. Click **Refactor** to proceed.