

# Generating Getters and Setters

---

You can generate accessor and mutator methods (getters and setters) for the fields that exist in your classes. IntelliJ IDEA generates getters and setters with only one argument, as required by the JavaBeans API.

The getter and setter method names are generated by IntelliJ IDEA according your [code generation naming preferences](#).

In this section:

- [Generating accessor and mutator methods](#)
- [Example 1](#)
- [Example 2](#)

## To create accessor and mutator methods

1. On the main menu, choose **Code | Generate**.  
Alternatively, right-click the editor and choose **Generate** on the context menu, or use **Alt+Insert** keyboard shortcut.

2. In the pop-up list that is displayed in the editor, select one of the following options:

### Getter

Accessor method for getting the current value of the selected fields.

### Setter

Mutator method for setting specified values to the selected fields.

### Getter and Setter

Both methods for the selected fields.

3. In the **Choose Fields to Generate Getters and Setters** dialog box, select the desired fields.

If getters and setters for a field already exist, this field is not included in the list.

4. Click **OK**.

## Example 1

Consider the following code:

```
public class MyClass {
    int aInteger;
}
```

In the **Naming** section of the [Code Generation](#) page, parameter prefix is set to `my`, and parameter suffix to `Param`.

After generating the getter and setter the following code will be produced:

```
public class MyClass {
    int aInteger;
    public int getAInteger() {
        return aInteger;
    }
    public void setAInteger (int myAIntegerParam) {
        aInteger = myAIntegerParam;
    }
}
```

## Example 2

However, if a is specified as a field prefix in the [Code Generation](#) page, then it will not take part in the generation of the method and parameter names:

```
public class MyClass {  
    int aInteger;  
    public int getInteger() {  
        return Integer;  
    }  
    public void setInteger (int myIntegerParam) {  
        aInteger = myIntegerParam;  
    }  
}
```