

# Creating JSDoc Comments

---

To make your code easy to use by other developers it is considered good practice to provide an HTML documentation of its Application Programming Interface (API). Such documentation can be generated automatically by the [JSDoc](#) tool. All you need, is supply your code with *documentation comments* in accordance with the *JSDoc standard*. The tool retrieves information from your comments and renders it in HTML using a built-in template.

You can find a detailed description of the JSDoc syntax with examples and explanation of their use in the article [An Introduction to JSDoc](#).

IntelliJ IDEA creates stubs of JSDoc comments on typing the opening tag `/**` and pressing Enter. If this feature is [applied to a method or a function](#), `@param` tags are created. In any other places IntelliJ IDEA adds an empty documentation stub.

[TODO patterns](#) and [Closure Compiler](#) annotations inside documentation comments are also recognized and are involved in code completion, intention actions, and other types of coding assistance.

Documentation comments in your source code are available for the [Quick Documentation Lookup](#) and open for review on pressing `Ctrl+Q` or `Alt+Button2` Click.

IntelliJ IDEA checks syntax in the comments and treats it according to the [Code Inspections](#) settings.

On this page:

- [Example of JavaScript comment](#)
- [Enabling creation of documentation comments](#)
- [Creating a JSDoc comment block](#)

## Example of JavaScript comment

Consider the following function:

```
function loadDocs(myParam1, myParam2){}
```

Type the opening documentation comment and press Enter to generate the documentation comment stub:

```
/**
 * @param myParam1
 * @param myParam2
 */
```

## To enable or disable automatic creation of documentation comments

1. [Open the Settings dialog box](#) and navigate to the [Editor. Smart Keys](#) page.
2. In the **Enter** section, select or clear **Insert documentation comment stub** check box.

## To create a JSDoc comment block for a method or a function

1. Place the caret before the method or function declaration.
2. Type the opening block comment `/**` and press Enter.
3. Describe the listed parameters and return values.

Documentation comment can be created with the dedicated action **Fix Doc Comment**. It can be invoked by means of [Find Action](#) command.

Press `Ctrl+Shift+A`, with the caret somewhere within a class, method, function, or field, which should be documented, and enter the action name **Fix Doc String**. The missing documentation stub with the corresponding tags is added. For example:

```

        /**
         *
         * @param userInput
         * @return
         */
        static boolean processRepeatConversion (@NotNull String userInput) {
            boolean repeatConversion = false;
            if (((userInput.equals("y")) || (userInput.equals("Y")))) {
                repeatConversion = true;
            }
            return repeatConversion;
        }
    }
}

```

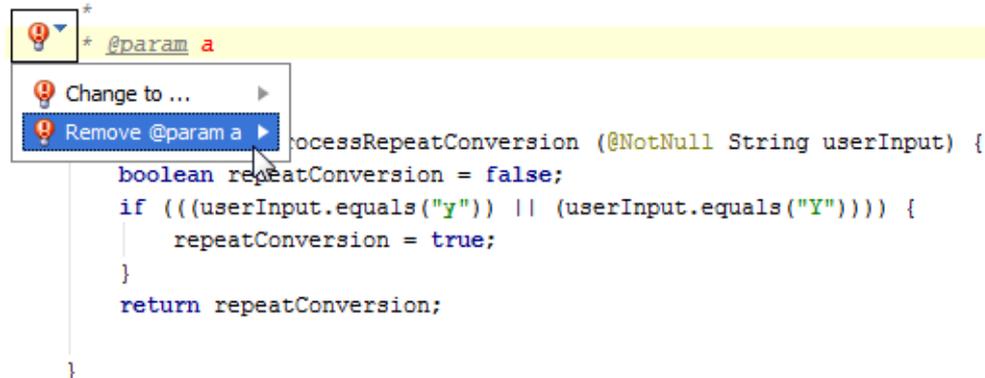
The next case lays with fixing problems in the existing documentation comments.

For example, if a method signature has been changed, IntelliJ IDEA highlights a tag that doesn't match the method signature, and suggests a quick fix.

```

public class MetersToInchesConverter {
    /**
     *
     * @param a
     */
    processRepeatConversion (@NotNull String userInput) {
        boolean repeatConversion = false;
        if (((userInput.equals("y")) || (userInput.equals("Y")))) {
            repeatConversion = true;
        }
        return repeatConversion;
    }
}

```



Note that the action has no keyboard shortcut bound with it. You can do it as described in the section [Configuring Keyboard Shortcuts](#).

## See Also

### Procedures:

- [Creating Documentation Comments](#)
- [Viewing Inline Documentation](#)

### Reference:

- [Inspections](#)

### Web Resources:

- [Developer Community](#) 