JavaScript Code Refactorings in PhpStorm

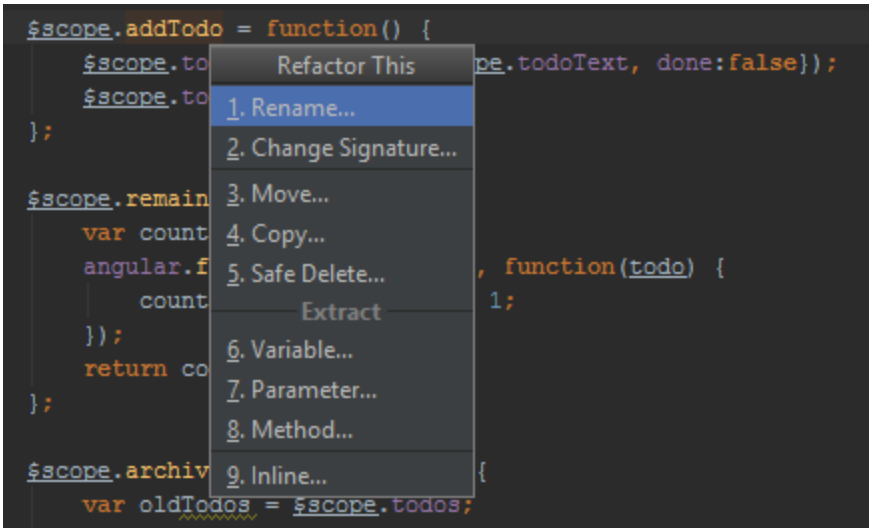
Martin Fowler states "Refactoring is a controlled technique for improving the design of an existing code base." In this tutorial, we'll have a look at the JavaScript code refactorings available in PhpStorm. Many refactorings are available in the IDE, which can carry them out automatically and makes sure existing code is updated and will not break. Let's see.

While this tutorial covers refactorings for JavaScript code, the IDE also supports refactoring other languages like PHP, HTML and so forth. Check our PHP Code Refactorings in PhpStorm tutorial to learn more, or use the the Refactor This i n the various language editors.

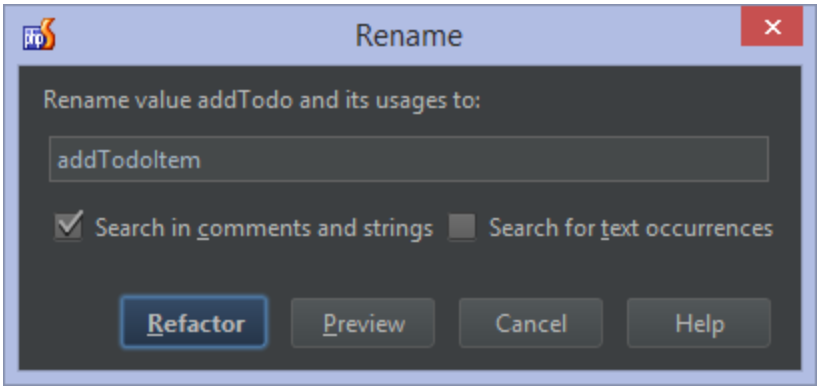
Refactoring in PhpStorm

Let's first see what a typical refactoring looks like. Each refactoring does different things, but there are some things they have in common. Let's try renaming a function in our codebase!

To do this, we can use the Refactor | Rename... context menu on any file or symbol, or place the cursor on it and use the Shift +F6 keyboard shortcut to invoke the rename refactoring immediately. Note that we could also use the Refactor This action. This last one is very convenient to invoke refactorings as we only need to remember one keyboard shortcut, Ctrl+Shift+Alt+T (Ctrl-T on Mac OS X), to show a pop-up with the different refactorings that we can do.



Whichever route we take, PhpStorm will show us the following dialog:

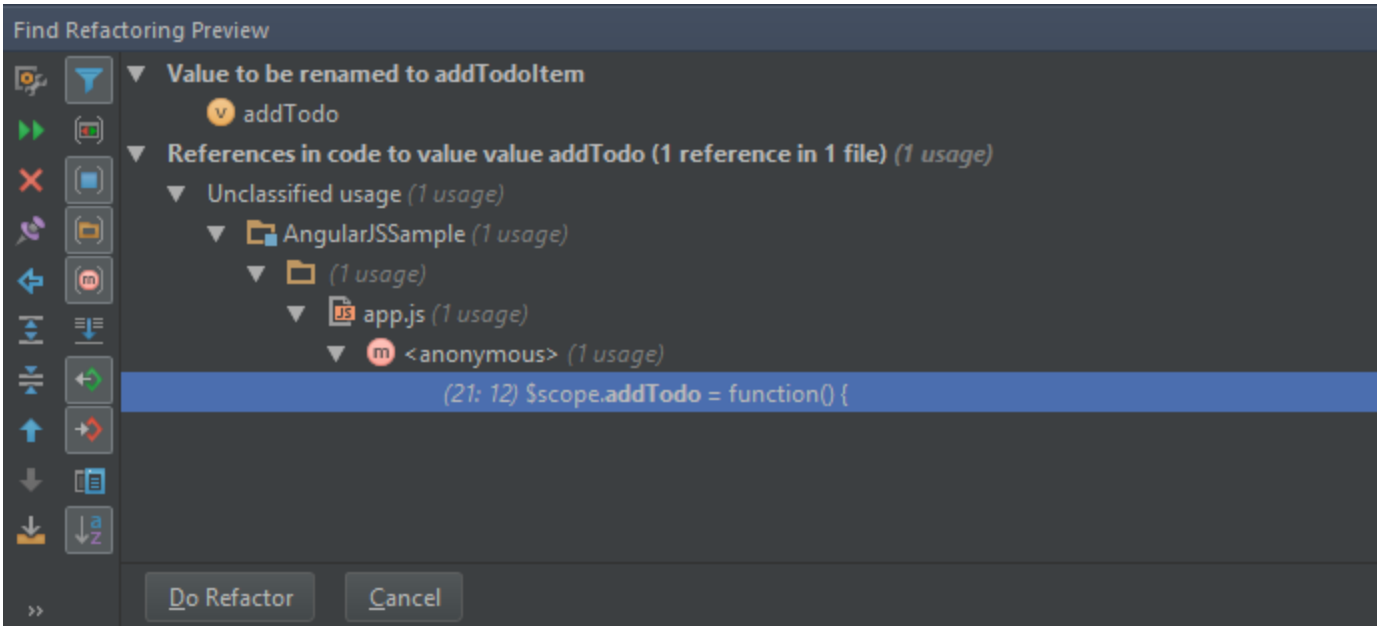


Each refactoring will show a different dialog in which we can provide the options for it. In this case, we have to provide the new name for the function we're about to rename. Some of the options in this dialog will be available for other refactorings as well: we can (optionally) search in comments and strings and do some sort of find-replace in there, we can search for text occurrences and so on.

When invoking a refactoring in PhpStorm, the IDE will:

- Perform the refactoring
- Track down and correct the affected code references automatically
- Warn about occurrences it cannot update automatically

If you are unsure about the outcome, consider clicking the Preview button before carrying out the refactoring. It will open the Find Refactoring Preview Tool Window and show us all of the actions the refactoring will perform.



In this case, it will:

- Rename the function
- Update one of the places where our code uses the addTodo function

We can filter and search in this tool window. Optionally, it also lets us select one of the occurrences and use Delete to have PhpStorm ignore that one whenever we perform the refactoring. This may be handy when we're also searching and replacing in comments and strings. Usually we will not exclude actual code refactorings as this may break our code.

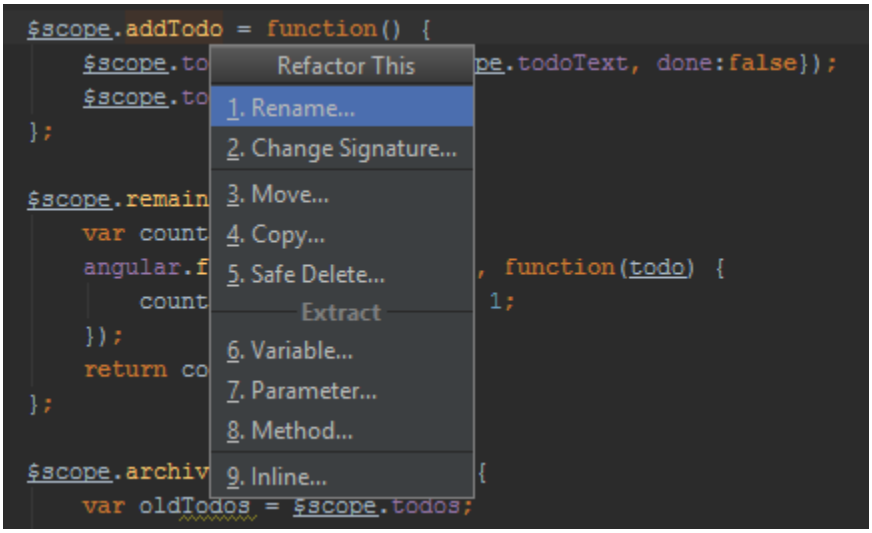
Clicking Do Refactor will perform the refactoring and update our project's codebase.

Why can't we just rename or move a file in the IDE? The reason is that it considers this as a refactoring: it changes the way the project is structured and potentially affects the source code and conventions of the programming language or framework being used. To make sure these conventions are met in the best way possible, the IDE requires us to invoke a rename (or move) refactoring to safely perform this action.

The Refactor This Action

While most refactorings in PhpStorm have their own shortcuts, we may not know all of them by heart. We may also be unfamiliar with the various refactorings available for a given file or symbol. And that's where the Refactor This action comes in handy!

In the Project View, Structure Tool Window, Editor or a UML Class Diagram, we can place the cursor on any file or symbol and use the Refactor | Refactor This context menu or press Ctrl+Shift+Alt+T (Ctrl-T on Mac OS X). This will display a pop-up with the different refactorings that we could carry out.



Using the up/down keys and Enter (or the numeric identifier for each entry in the pop-up), we can invoke the refactoring.

Available Refactorings

There are many refactorings available for JavaScript code. In this section, let's go over them and see what actions they perform. For every refactoring, we'll add a link to the PhpStorm Web Help, which contains a full description of all available options for a given refactoring.

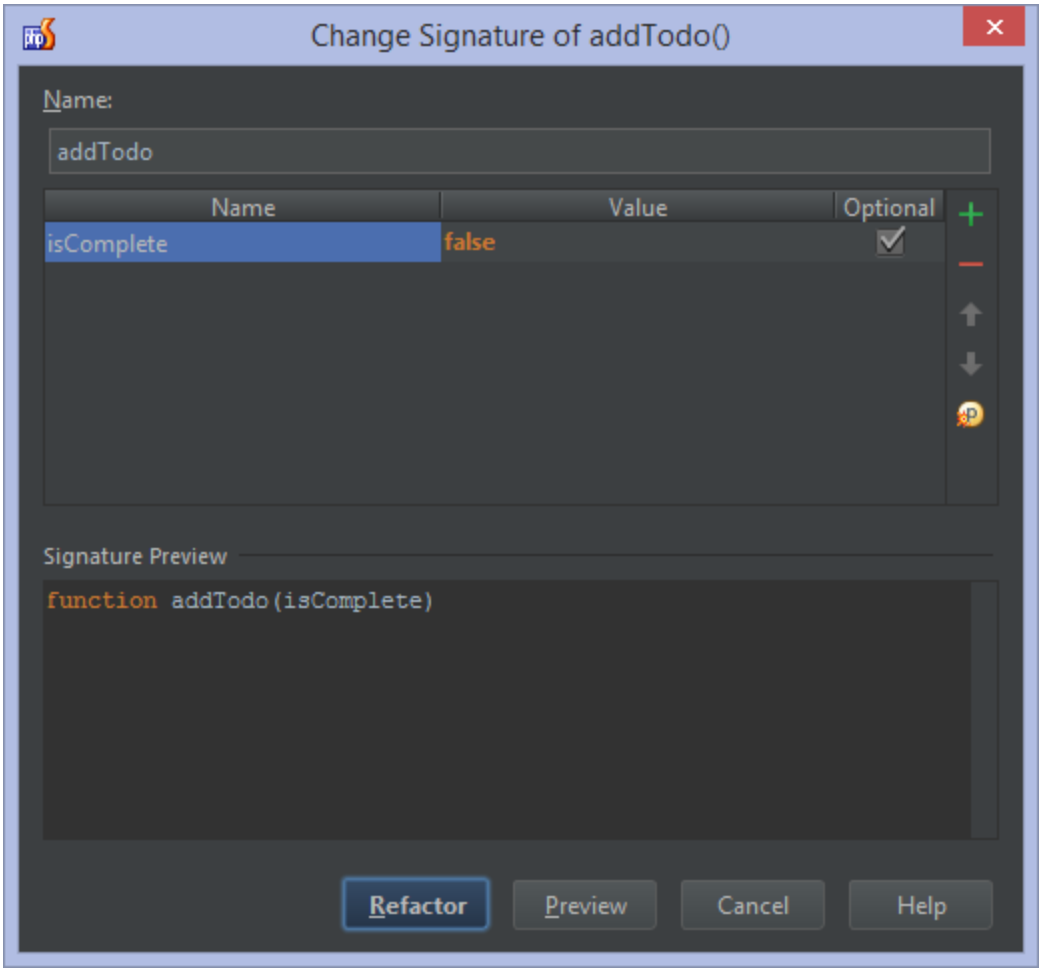
All refactorings are available from the Refactor context menu or via their respective keyboard shortcuts.

> Check out the Hands-on PhpStorm workshop materials to download a sample project in which we can try all of these refactorings.

Change Signature Refactoring

When we want to change the name of a function or add/remove parameters for it, we can use the Change Signature refactorin g by pressing Ctrl+F6 (or CMD-F6 on Mac OS X). It can:

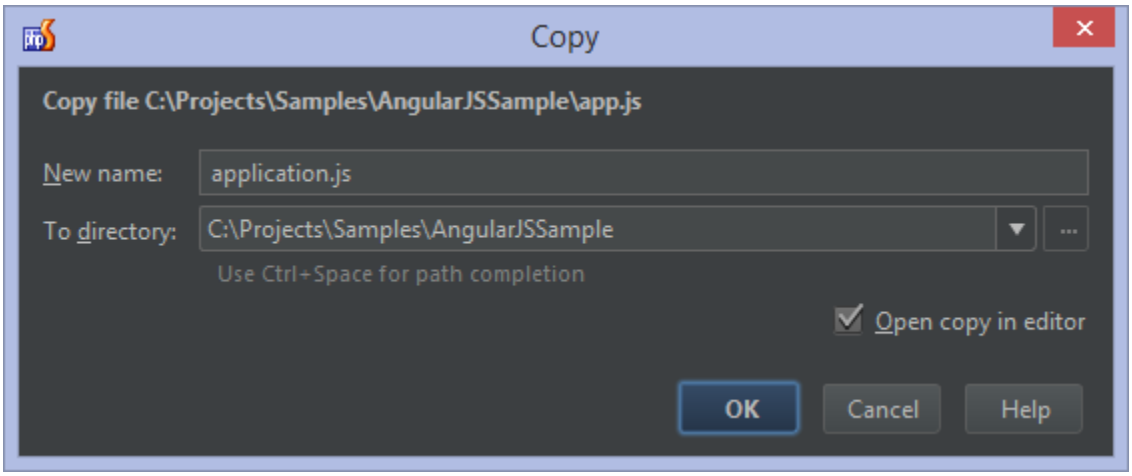
- Change the function name
- Add new parameters and remove the existing ones
- Assign default values to the parameters
- Reorder parameters
- Change parameter names
- Propagate new parameters through the function call hierarchy



Check Web Help for full details.

Copy Refactoring

Using the Copy Refactoring, we can copy a class, file or directory to another directory. We can invoke it by selecting a file and using the keyboard shortcut (F5) or by dragging the file and dropping it in a folder whilst we hold the Ctrl key down.

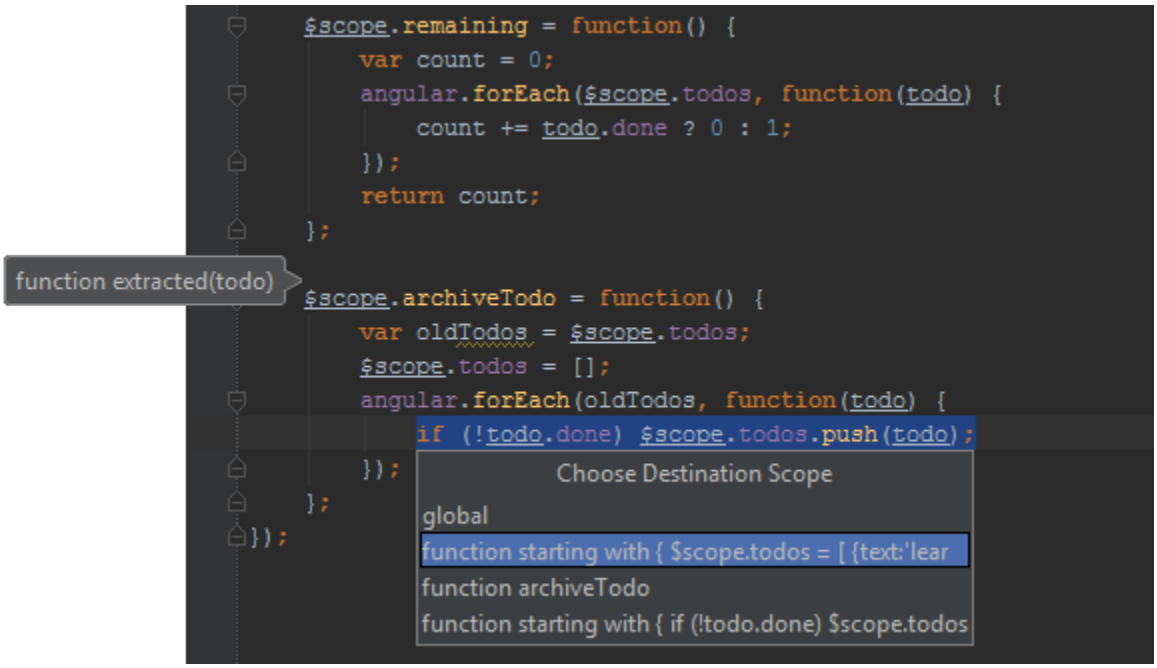


Check Web Help for full details.

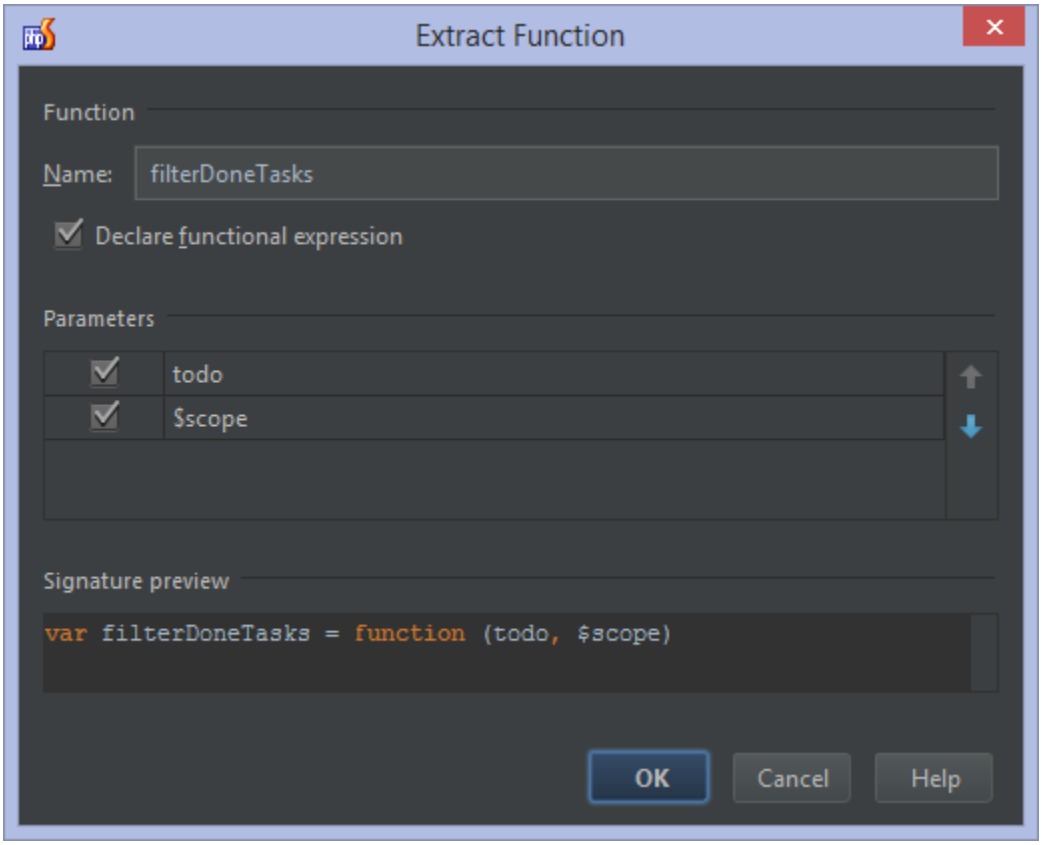
Extract Function Refactoring

Whenever we want to extract a piece of code into a separate function, we can make use of the Extract Function Refactoring by

pressing Ctrl+Alt+M (or Alt+CMD+M on MacOS X). Upon invocation, PhpStorm lets us select the scope in which the extracted function will be defined. The IDE will also show us exactly where the function will be injected into our code.



The refactoring will extract the selected block of code into a function, detecting parameters and return values. We can choose the name of the function, pick how it will be defined (using declaration or not) and select and name function parameters.

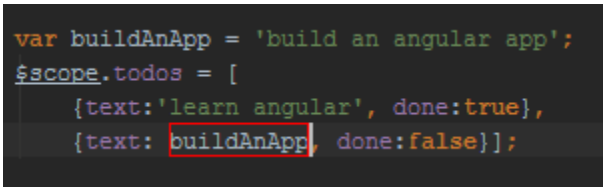


Check Web Help for full details.

Extract Variable/Constant Refactoring

We can extract a selected expression into a variable or constant by pressing Ctrl+Alt+V (Alt-CMD-V on Mac OS X). The original

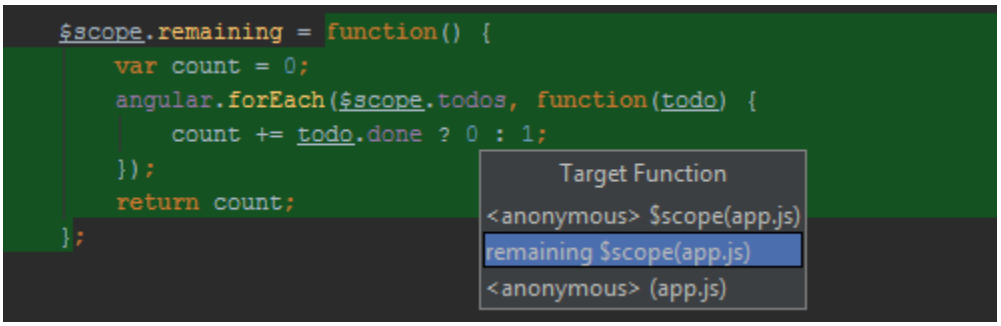
expression will be replaced with the new variable.constant.



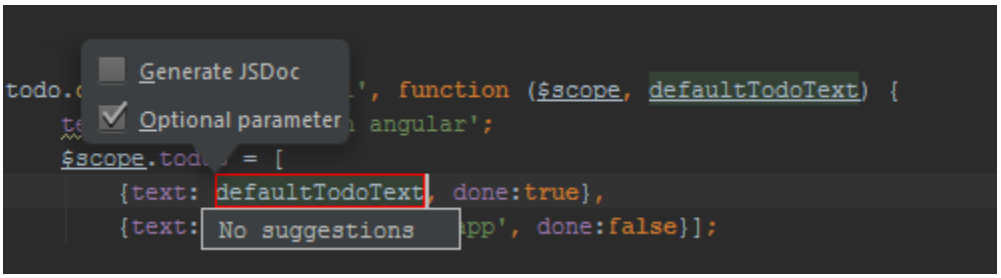
Check Web Help for full details.

Extract Parameter Refactoring

The Extract Parameter refactoring can be used to add a new parameter to a function declaration and to update the function calls accordingly. It can be invoked by pressing Ctrl+Alt+P (or Alt+CMD+P on MacOS X) on any variable or literal in our code. Upon invocation, PhpStorm lets us select the scope/function to which the extracted parameter will be added (if applicable).



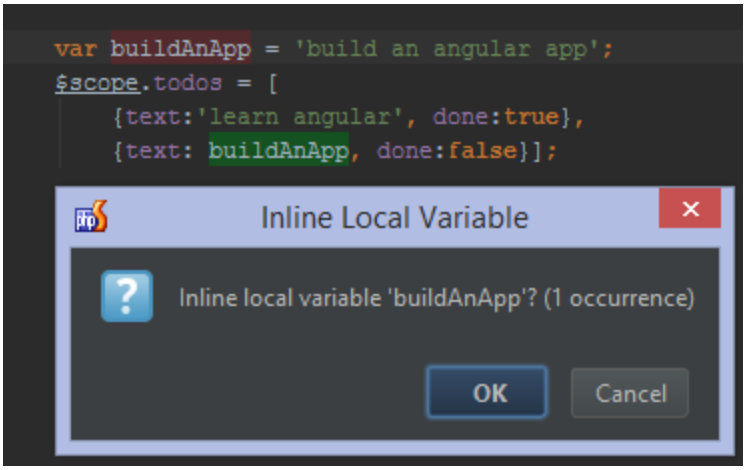
The refactoring will create a new function parameter which we can name, make optional (so a default value assignment will be generated as well) and generate JSDoc comments for.



Check Web Help for full details.

Inline Refactoring

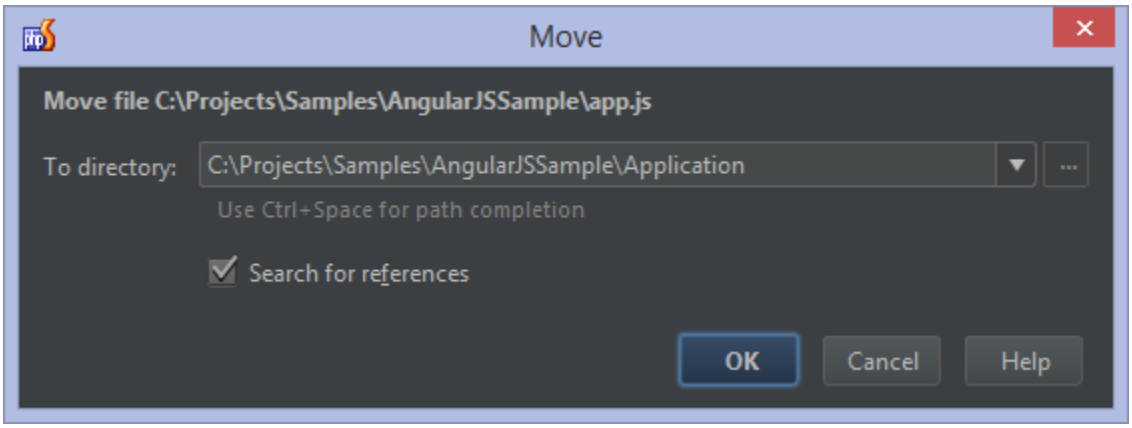
The Inline Refactoring Replace is the opposite of Extract Variable/Constant / Extract Function refactorings: it replaces redundant variables or functions with the full expression. We can select a variable and press Ctrl+Alt+N (or Alt-CMD-N on Mac OS X) to invoke it.



Check Web Help for full details.

Move Refactoring

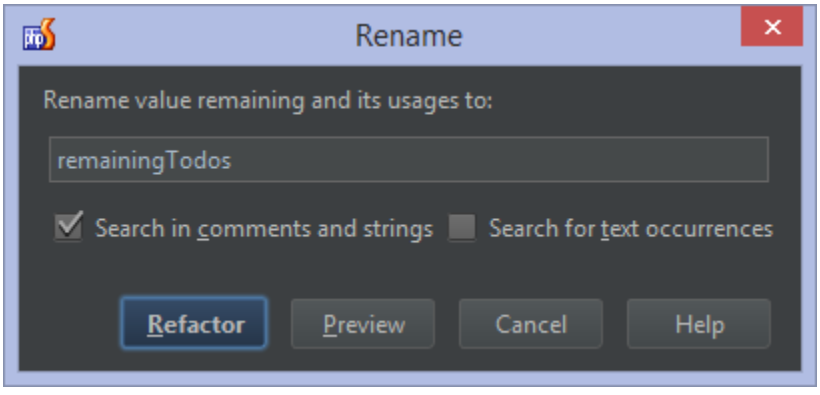
With the Move Refactoring (F6), we can change the location of a file or directory.



Check Web Help for full details.

Rename Refactoring

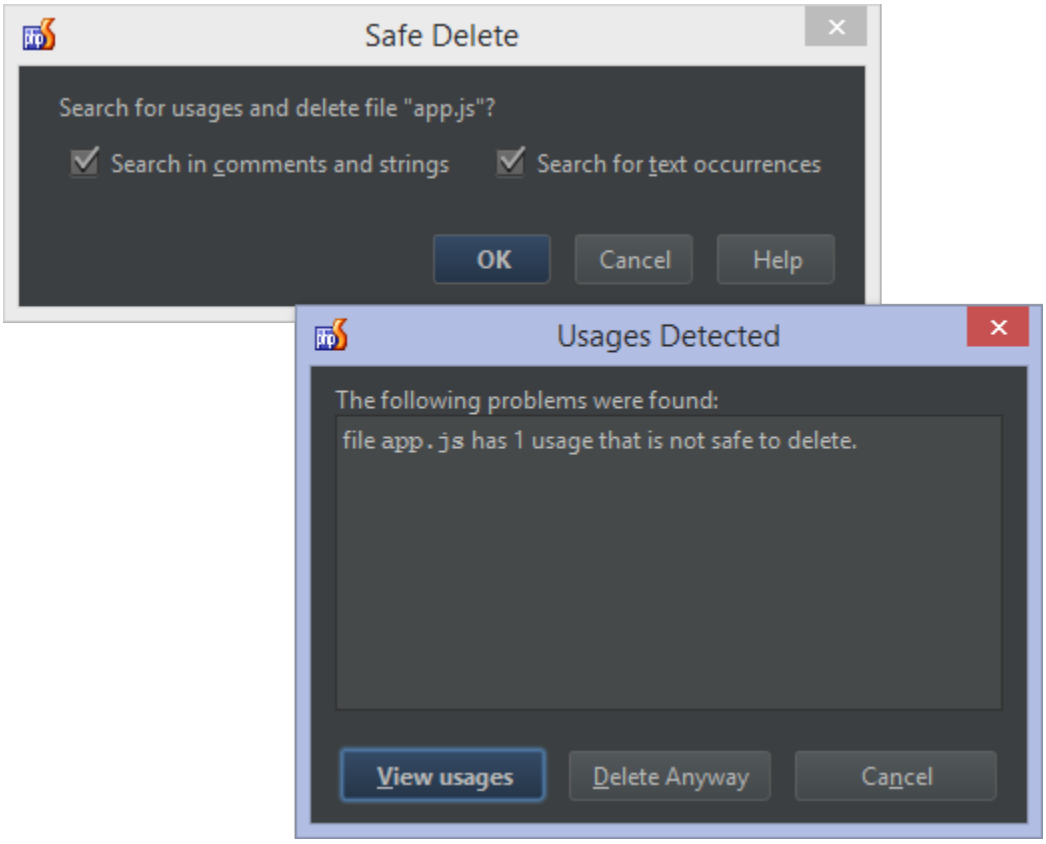
Using the Rename Refactoring (Shift+F6), we can rename symbols, automatically correcting all references in our codebase. We can rename Functions (and therefore, namespaces and classes), Variables, Parameters, CSS Color Values, Files and Directories.



Check Web Help for full details.

Safe Delete Refactoring

Not sure if a symbol or file is being used elsewhere but still want to delete it? Simply deleting it may break our application, but using the Safe Delete Refactoring (Alt+Delete or CMD-Delete on Mac OS X) we will get notified when references to it still exist. This allows us to either immediately delete the file or symbol when they are no longer used, but will give a warning if it is still in use.



Check Web Help for full details.

Tweet