

# Intention Actions

IntelliJ IDEA helps you handle the situations when you use classes that haven't been imported, or methods that haven't been written etc., which can result in errors. When a possible problem is suspected, IntelliJ IDEA suggests a solution, and in certain cases can implement this solution (properly assign variables, create missing references and more). Besides syntax problems, IntelliJ IDEA recognizes code constructs that can be optimized or improved, and suggests appropriate *intention actions*, denoted with the special icons.

On this page:

- [Intention action icons](#)
- [Intention action types](#)

In this section:

- [Applying Intention Actions](#)
- [Configuring Intention Actions](#)
- [Disabling Intention Actions](#)

## Intention action icons

Item	Icon	Description
Intention actions suggested		A yellow bulb indicates that IntelliJ IDEA just proposes to alter your code. It covers a range of situations from warning correction to suggestions for code improvement (like micro-refactorings).
Specific intention action		This sign appears in the suggestion list before each specific intention action. If an intention action alert is disabled, the sign turns to  . Disabled intention action is still available and can be enabled again.
Quick-fix suggested		A red bulb with an exclamation mark indicates that IntelliJ IDEA suggests a way to fix an error. It is related to Create from usage intentions and Quick fixes.
Disabled		Alert is disabled, but the intention action is still available and can be enabled again.

## Intention action types

Find descriptions of specific intention actions on the [Intentions page](#) of the **Settings** dialog, where they are grouped according to the areas of their usage. Generally, intention actions can be divided into several categories, for example:

### Create from usage

This type of intention action creates new code items: classes, methods, etc. They are smart enough to analyze your code and provide actions suitable for a particular case. The main concept behind this type is that you can begin using new things without declaring them first. You are not taken away from your current task for mundane minutiae like creating declarations, new files, etc. which IntelliJ IDEA handles while you keep focused.

For example, *Create Constant Field* is suggested if the reference is uppercase, or *Create class* appears when a name is typed after the `new` keyword, or when an identifier starts with a capitalized letter, etc.

### Quick fixes

This type of intention action responds to common coding mistakes: using an improper access modifier, or an expression of the wrong type, or missing resources, etc. IntelliJ IDEA catches these kinds of problems as you type, and provides a quick way to fix them

using Intentions Actions with appropriate suggestions for the error.



### Micro-refactorings

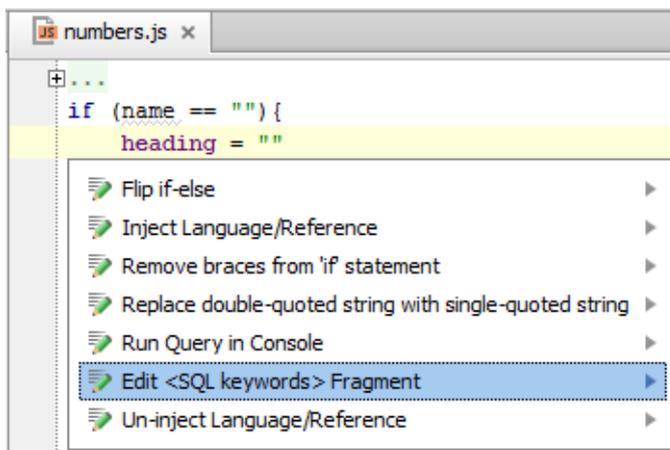
These intention actions appear for code that is syntactically correct, but can be structurally improved by such things as:

- Converting code constructs.
- Splitting declarations and assignments.
- Splitting or merging statements and tags, etc.



### Edit <Injected Language> Fragment

For string literals that represent [language injections](#), the Edit <Injected Language> Fragment intention action is available. You can use this intention action to open the corresponding code fragment in a separate editor.



### See Also

Concepts:

- [Code Inspection](#)

Reference:

- [Intentions](#)

External Links:

- [Check lambda support in IntelliJ IDEA 12 EAP](#)
- <http://www.jetbrains.com/idea/documentation/intentions.jsp>

Web Resources:

- [Developer Community](#)