TUTORIALS _ Using Intellij Idea

version **#0.5**

YET ANOTHER KIND OF ASSISTANTS 2024 <assistants@tickets.assistants.epita.fr>

Copyright

This document is for internal use at EPITA (website) only.

Contents

*https://intra.forge.epita.fr

1 Usage Tips

Like vim and emacs, IDEA is a configurable IDE and you should feel free adjust it to your preferences.

IDEA supports plugins through a marketplace. Most of them are free and provide useful features and appearance tweaks to your environment. To see the list of available plugins, you can go to `Files -> Settings -> Plugins`, or by default `Ctrl+Alt+S`.

1.1 Shortcuts

This section is designed to provide you with some of the most useful shortcuts using this IDE. These tips and the associated shortcuts are the ones set by default. If you decide to change them from the default configuration, the shortcuts will still appear in their respective menus and in your settings. Finally, these features are available through the user interface, try to play around and find them.

Be careful!

Some of the shortcuts described refer to specific *Java* notions that you do not know yet of. Keep this cheatsheet in a tab for later!

1.1.1 Must-know

These ones can be useful all the time:

- The `Alt+Enter` shortcut can be used to show quick fix options. When you have a piece of code that is causing an error or warning, you can place your cursor on the problematic code and press `Alt+Enter` to bring up a list of quick fix options. These options can be used to automatically fix the error or warning, or to help you understand what might be causing the issue.
- The `Alt+Insert` shortcut can be used to generate code. For example, if you want to create a new class, you can use this shortcut to bring up a list of options for generating various types of code, such as a class, interface, or enum. You can then select the type of code you want to generate and follow the prompts to complete the code generation process.
- The `Ctrl+Alt+L` shortcut reformats your code. Keeping your code clean is essential. This shortcut does everything for you, and the assistant reading your code will appreciate formatted code.

1.1.2 Object-oriented navigation

When creating a complex tree of object inheritance, the following shortcuts can ease the navigation between different classes:

- `Ctrl+U`: Navigate to the super-method/super-class. This will take you to the method or class that is being overridden or extended by the current method or class.
- `Ctrl+Alt+B`: Navigate to implementation(s). This will take you to all of the places where the current method or class is being implemented or overridden.
- `Ctrl+N`: Go to class. Type the name of a class and press `Ctrl+N` to navigate to it.
- `Ctrl+Shift+N`: Go to file. Type the name of a file and press `Ctrl+Shift+N` to navigate to it.

- `Ctrl+Shift+Alt+N`: Go to symbol. Type the name of a symbol (such as a method or variable) and press `Ctrl+Shift+Alt+N` to navigate to its declaration.
- `Ctrl+Alt+F7`: Find usages. Place your cursor on a symbol and press `Ctrl+Alt+F7` to find all of the places where it is being used.

1.1.3 General navigation

Navigating between many tabs can be a hassle. These shortcuts can help you in that context:

- `Ctr+Alt+[Arrow]`: Navigate between the previous or the next position of your cursor in the history.
- `Shift` quickly tapped twice: Open the symbol search window. Allows you to find any class or method in a project.

1.1.4 Refactoring options

These ones will be useful from time to time to refactor your code:

- `Shift+F6`: Rename. Place your cursor on a symbol and press `Shift+F6` to rename it throughout your project. This one works almost everywhere.
- `Ctrl+T`, `Ctrl+Alt+T`, `Ctr+Alt+Shift+T`: Bring up a menu with a list of refactoring options.

The last one provides a complete set of tools:

- Change the signature of a method (add, remove a parameter, change its return type).
- Extract method: Extracts a selected part of a function and creates an auxiliary function
- Create variable `Ctr+Alt+V`: Creates from a selected expression or a line a variable. It automatically determines its type.

Using any of these options will update or eventually replace every occurence of an usage automatically.

These tips are here to help you get started with the IDE. You might find your own way and shortcuts for using IDEA later on. Keep in mind during this workshop that IDEA can generate code that can save you a lot of time.

Going further...

Feel free to browse through the official Jetbrains documentation for a complete overview of the available features.

You mean it's working? For real this time?