

Groovy

This part concerns the issues related to Groovy, and IntelliJ IDEA's specific techniques that become available when Groovy support is enabled.

In this section:

- Groovy
 - [Groovy Support](#)
- [Creating Modules with Groovy and Groovy-Based Frameworks Support](#)
- [Groovy Procedures](#)
- [Groovy-Specific Refactorings](#)
- [Launching Groovy Interactive Console](#)
- [Launching Groovy Shell](#)

Make sure that the desired SDK is downloaded and installed on your computer, and the libraries are properly configured.

Also, make sure that Groovy plugin is enabled in IntelliJ IDEA. The plugin is bundled with IntelliJ IDEA and activated by default. If the plugin is not enabled, [enable the plugin](#).

Groovy Support

IntelliJ IDEA supports Groovy up to version 2.0.

Groovy files are marked with  icon.

Groovy support includes:

1. Possibility to [create Java modules with Groovy facet](#).
2. Ability to create an [extension module](#)  to support custom extension methods.
3. Coding assistance:
 - [Code completion](#) for keywords, labels, variables, parameters and functions.
 - [Chained expression completion](#) .
 - Error and syntax highlighting.
 - Code [formatting](#) and [folding](#).
For example, all closures that have their parameters described in the separate line are formatted the following way:

```
def c1 = {
    param1, param2 ->
        body()
}
```
4. [Code generation](#)
 - Inserting, expanding, and generating code blocks using [live templates](#).
 - Creating various applications elements via [intention actions](#).
 - Possibility to create [line and block comments](#) (Ctrl+Slash or Ctrl+NumPad //Ctrl+Shift+Slash or Ctrl+Shift+NumPad /).

5. **Common** and **specific** refactorings.
6. Numerous ways to **navigate** through the source code, among them:
 - **Navigating with Structure View**.
 - **Navigate | Declaration** (Ctrl+B, Ctrl+Button1 Click or Button2 Click).
 - **Navigate | Implementation** (Ctrl+Alt+B or Ctrl+Alt+Button1 Click) from overridden method / subclassed class.
7. Advanced facilities to **search through the source code**.
8. **Viewing reference** information.
9. Possibility to **generate** documentation, created according to GroovyDoc syntax.
10. Possibility to **compile** mixed Groovy and Java code.
11. **Running and debugging**.
12. Possibility **create**, and **perform** tests.
13. Access to a **Groovy console** in any Java project (**Tools | Groovy Shell**).

See Also

Concepts:

- [Facet](#)

External Links:

- <http://groovy.codehaus.org/> 

Web Resources:

- [Developer Community](#) 