

This page describes the noticeable changes made to the default code formatting settings in IntelliJ IDEA. This is also a chance to review some of these changes to validate or tweak them if necessary:

Issues

These are the reported issues that we still discuss (or simply things that the formatter does not support). The first is what we use.

- Space in annotation parameter assignment: `@Foo(name="foo")` vs. `@Foo(name = "foo")`
- Space in annotation array parameter: `@Target({ ElementType.METHOD, ElementType.TYPE })` vs. `@Target({ElementType.METHOD, ElementType.TYPE})`

General

- Default indent option to use tab character instead of space

Java

Tabs and indent

- Use tab character

Spaces

- Add a space before the left brace of an array initializer

Wrapping and braces

- Keep when reformatting: *multiple expressions in one line, simple blocks in one line, simple classes in one line*
- `else`, `catch` and `finally` on new line
- Method declaration parameters: do not align when multiline

Blank lines

- Keep one space before `}` (solely use to keep the space between the end of the last method and the end of the class)
- Minimum blank line after class header 0 (instead of 1)

Javadoc

- Disabled

Imports

- Use **100** for "Class count to use import with `*`" so that imported classes are always listed individually.
- Use **0** for "Names count to use static imports with `*`" since static imports are only allowed in test sources where they can and should be aggregated, e.g. `import static org.junit.Assert.*; .`
- Use the following import sequence order:
 - `java.*`
 - `<blank line>`
 - `javax.*`
 - `<blank line>`
 - all other imports
 - `<blank line>`
 - `org.springframework.*`
 - `import static` all other methods

