

Clean Code Cheat Sheet for Java/Spring Boot

1. Project structure (as prescribed by Maven):

- a. `src/main/java`: For source files
- b. `src/main/resources`: For resource files, like properties
- c. `src/test/java`: For test source files
- d. `src/test/resources`: For test resource files, like properties

2. Consistent naming conventions:

- a. Classes: uppercase
- b. Variables: lowercase
- c. Methods: lowercase and starting with a verb
- d. CamelCase

3. Using clear and speaking names for all constructs (i.e. trying to establish a ubiquitous language), in general making code as readable and understandable:

- a. Trying to keep it close to natural language where possible (e.g. by using method chaining)

4. Usage of @Annotations to reduce boilerplate code (this one might be controversial!):

- a. e.g. Lombok's `@Getter` and `@Setter` instead of e.g.

```
public String getName() {  
    return name;  
}
```
- b. e.g. Spring's `@Service` for autodetection by SpringBoot

5. Avoiding hardcoding:

- a. Using Enums for constants
- b. Setting values of variables with methods

6. Avoiding repetition (DRY):

- a. Using existing libraries where available and fitting
- b. Writing functions/methods

7. JavaDoc/documentation:

- a. Writing JavaDoc for every method which may be used by yourself or other developers
- b. Using comments (very sparingly!) where further context is necessary

8. Testing the code well:

- a. Unit Tests
- b. Integration Tests
- c. Regression Tests

9. Having somebody else look over your code (static code analysis):

- a. Code reviews (e.g. with merge requests)
- b. Code walkthroughs

10. Constant refactoring (also of other people's code!)

11. Clean code tools

- a. SonarQube
- b. SonarLint