

General Guidelines and Requirements	1
Writing General Guidelines	2
Article Structure	2
Code Snippets in Articles	3
Code Guidelines	4
Payment	4

**This is a live document. We are continuously updating it. Please continue referring to this document to make sure your article follows the guidelines and checklist before submission.**

## General Guidelines and Requirements

1. Highly recommended using Grammarly(<https://www.grammarly.com/>) as your writing tool (especially if you're not a native speaker). It will help minimize revision in the final document.
2. You can also use ProWritingAid as an alternative to Grammarly.
3. All articles should be run through Copyscape Premium or similar premium services to check for plagiarism and the report should be part of final article submission. **We will not accept articles without the report. We will also run an internal check for plagiarism** and will reject the article in case we find any discrepancy.
4. All articles should have a complete and running example and we should submit it to our GitHub repository.
5. **We will not count source code in the final article's length.**
6. Run the article through spell check to avoid unnecessary revisions.

## Writing General Guidelines

1. All Title and Subheading should follow the Title Capitalization rule.
2. Articles should follow the below structure consistently.
  - a. Main Heading should be H2.
  - b. Sub-Heading should use H4.
3. There should be no sub-sub heading unless we require it.
4. Articles should be a combination of code and explanation.
5. **We should take code examples from real-world problems and it should not pick simple programs using numbers or alphabets.**
6. Articles should use the keywords (will provide the keywords) at appropriate positions. We should not overuse the keywords. E.g. Articles about Spring Boot should not use it over 4-5 times.
7. Keep the paragraphs small and simple. Use multiple paragraphs if we need to provide more details.
8. **Always use We** to address in your article, **avoid using You**. e.g. **"We will add required dependencies"** in place of **"You should add required dependencies"**.
9. Check with the team with article headings and subheading before starting on the article.
10. Submit the images used in the article separately. We will not accept an article with embedded images only.

## Article Structure

All articles should follow the below structure and workflow.

1. Overview (What the articles about and what we will learn). - H2
2. Application Setup. (Refer to the following section for more details - <https://www.javadevjournals.com/spring-boot/spring-boot-configuration-properties/#2-Setting-up-Application>)
  - a. Maven dependencies.
  - b. Overview of the dependencies and why we need those.
  - c. Final pom.xml file.
  - d. Screenshot for the project generation.
3. Configurations overview if required.
  - a. Application.properties file if you are working on Spring article.
  - b. Overview of the properties.

4. Overview of the application.
5. Other sections and subsections based on articles.
6. For recapping or highlighting some important points in the section, use the bullet points in the articles.
7. Use images only for the following section in your articles.
  - a. To show the project generation (e.g. Spring Initializr.)
  - b. To show the project layout in your IDE.
  - c. To display the layout or workflow (Refer to <https://www.javadevjournal.com/spring-boot/spring-boot-oauth2/> for Oauth2 workflow.).
  - d. To display the output e.g. if you are using Postman or similar REST client too. (e.g. <https://www.javadevjournal.com/spring-boot/spring-boot-with-mongodb/> for REST API output).
  - e. HTML layout. (Refer to the following article <https://www.javadevjournal.com/spring/password-encoding-in-spring-security/#8-Testing-Password-Encoding-With-Spring>)

## Code Snippets in Articles

Our articles should contain texts and code snippets. Use the following guidelines to add code snippets in the articles.

1. Heading and section should contain the code example for better clarity.
2. Subsections should contain code snippets relevant to the sections and not the complete example. Put the complete example or class at the end of the section.

E.g. If we are explaining some configuration, put the relevant configuration section code in the subheading.

```
@Override
    public void configure(AuthorizationServerSecurityConfigurer
oauthServer) throws Exception {
        oauthServer.tokenKeyAccess("permitAll()")
            .checkTokenAccess("isAuthenticated()");
    }
```

Subheading contains only the relevant code snippet.

3. Final code base should contain unit test cases to validate individual methods. **We will not accept article submission without unit test cases.**
4. **Code should be correctly indented:**
  - a. Use the Google Java code formatter (<https://github.com/google/google-java-format>).
5. Fluent API should be formatted correctly for better readability. Split the Fluent API into multiple lines for better reliability rather than putting it in one line. Here is a sample example.

```
public void configure(ClientDetailsServiceConfigurer clients) throws Exception
{
    clients
        .inMemory()
        .withClient(clientID)
        .secret(passwordEncoder.encode(clientSecret))
        .authorizedGrantTypes("password", "authorization_code",
"refresh_token")
        .scopes("user_info")
        .authorities("READ_ONLY_CLIENT")
        .redirectUris(redirectURLs)
        .accessTokenValiditySeconds(accessTokenValidity)
        .refreshTokenValiditySeconds(refreshTokenValidity);
}
```

## Code Guidelines

1. Code should not contain System.out.println.
2. No default constructors.
3. Java 8 syntax is recommended and should be used when possible.
4. Code package should always start with com.javadevjournals

## Payment

We follow a predefined process to release the payment. **We do not hold payment till we publish the article.**

1. Before we start, we discuss the article topic.
2. Once the topic is finalised, we will discuss and complete the article layout and topics to be covered in the article.
3. We will start a milestone with a timeline for the submission.
4. On submission, we will review the article and provide the feedback for any revision.
5. Once an article is approved, please submit the article with the required checklist (refer to this document). We will release the payment.