



Documentation Tools for Android Development: A Review of Dokka

Author: Jubaer Ahmad Khan

Date: 15 September, 2024

Department of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka-1342

Contents

1	Introduction and Objective	2
1.1	Introduction	2
1.2	Objective	2
2	Installation and Working Process	3
2.1	Installation	3
2.1.1	Steps for installation:	3
2.2	Working Process	4
3	Pros and Cons	5
3.1	Pros	5
3.2	Cons	5

Chapter 1

Introduction and Objective

1.1 Introduction

Dokka is a powerful documentation tool specifically designed for Kotlin and Java projects. It generates documentation from comments written in Javadoc or KDoc, which makes it particularly useful for developers using Android Studio, Java, and XML in their projects. In this report, I evaluate Dokka as a potential documentation tool for my project development process using Android Studio.

1.2 Objective

The objectives of this report are as follows:

- Evaluate Dokka as a documentation tool for Android application development using Java and XML.
- Assess the ease of integration of Dokka with Android Studio.
- Analyze the documentation generation capabilities of Dokka, including its output formats.
- Understand how Dokka improves project management and collaboration in software development.
- Provide insights into the pros and cons of using Dokka for Android development projects.

Chapter 2

Installation and Working Process

2.1 Installation

Dokka can be integrated into an Android project by adding it to the Gradle build file.

2.1.1 Steps for installation:

1. Add the Dokka plugin to your `build.gradle` file:

```
plugins {  
    id 'org.jetbrains.dokka' version '1.4.32' // Use the latest version  
}
```

2. Configure the output formats (HTML, Markdown, Javadoc):

```
dokkaHtml {  
    outputDirectory.set(buildDir.resolve("dokka"))  
}
```

2.2 Working Process

Once integrated, you can generate documentation by running the following Gradle command:

```
./gradlew dokkaHtml // Generates HTML documentation
```

Dokka uses the comments from Javadoc or KDoc to generate detailed documentation. It supports output in HTML, GitHub Flavored Markdown, or the traditional Javadoc format, making it versatile for different platforms.

Chapter 3

Pros and Cons

3.1 Pros

- **Multi-format Support:** Dokka generates documentation in multiple formats, including HTML, Markdown, and Javadoc.
- **Easy Integration:** It integrates seamlessly into Android projects using Gradle.
- **KDoc and Javadoc Compatibility:** Supports both Kotlin's KDoc and Java's Javadoc, which is beneficial when working with mixed language projects.
- **Customizable Output:** The output formats are customizable to match specific project documentation needs.

3.2 Cons

- **Limited to Kotlin and Java:** It is not designed for use with other programming languages.
- **Customization Complexity:** Dokka can be somewhat complex to set up advanced configurations for large-scale projects.
- **Learning Curve:** For developers not familiar with Kotlin or advanced Gradle configurations, Dokka can have a steep learning curve.