1. Introduction

1.1 Purpose

This document provides step-by-step instructions for building and deploying/publishing our Kotlin based IntelliJ plugin using Gradle as a build tool.

1.2 Scope

The scope of this document covers the entire process from cloning the repository to publishing the plugin to the marketplace.

2. Prerequisites

2.1 Software Dependencies

Ensure the following software is installed on your machine:

- Java JDK
- Gradle

3. Build Process

3.1 Clone Repository

```
git clone <repository-url>
cd amos2023ws02-pitest-ide-plugin/pitmutationmate
```

3.2 Gradle Build

```
./gradlew build
```

3.3 Run Tests

```
./gradlew test
```

4. Deploy/Publish

4.1 Publishing the PIT MutationMate plugin to the JetBrains Marketplace

To deploy the plugin to the JetBrains Marketplace, you need to supply the Personal Access Token, which you can find on this project's JetBrains account. The account details can be requested from the members of the project. Once you have the token, set the following environment variable:

```
export PUBLISH_TOKEN=<your_token>
```

and publish the plugin via the gradle task:

```
cd amos2023ws02-pitest-ide-plugin/pitmutationmate
./gradlew publishPlugin
```

Please check JetBrains Publishing a Plugin documentation for more information and other ways of publishing the plugin.

4.2 Publishing the companion Gradle plugin and our PITest dependency

Our IntelliJ plugin relies on a custom Gradle plugin as well as a PITest dependency which are also part of this project.

4.2.1 Publishing the Gradle companion plugin

The Gradle companion plugin can be found in `/pitmutationmate-override-plugin`. To publish it you will need a `gradle.publish.key` and a

`gradle.publish.secret`, which can be found in the the project's Gradle account. Account information can be requested from the members of the team. Once these have been retrieved, the Gradle plugin can be published via the gradle task:

```
cd pitmutationmate-override-plugin
./gradlew publishPlugins -Pgradle.publish.key=<key> -Pgradle.publish.secret=<secret>
```

Please check Gradle's Publishing Plugins documentation for more information.

Publishing the PITest dependency

The PITest dependency `coverage-reporter` can be found in `/coverage-reporter`.

To publish it you will need a username and a password, which can be found in the the project's Sonatype account. Account information can be requested from the members of the team. Once these have been retrieved, create a `gradle.properties` file and set the gradle properties `ossrhUsername` and `ossrhPassword` as shown below. Then PITest dependency can be published to the Sonatype maven repository via the gradle task:

```
cd coverage-reporter
echo 'ossrhUsername="<username>"\nossrhPassword="<password>"' > gradle.properties
./gradlew publish
```

Please check Sonatype's Publishing Plugins documentation for more information.

5. Troubleshooting

5.1 Build Issues

If the build fails, check the error messages and resolve any missing dependencies or configuration issues.

5.2 IDE Errors

1. If the companion plugin or the coverage-reporter dependency were changed, run the gradle task `publishToMavenLocal`.
2. File/Repair IDE (you can do this for the sandbox-ide and the one you have installed locally)
3. File/Invalidate Caches (you can do this for the sandbox-ide and the one you have installed locally)
4. Delete all build, .gradle and .idea folders. (Also in the projects that you open with the sandbox-ide)
5. Delete the ~/.gradle/caches folder.
6. Check for updates in your IDEA and install them
7. Install the Android plugin in the sandbox-ide (for testing android projects)
8. If the companion plugin or the coverage-reporter dependency were changed, they can be tested locally by running the gradle task `publishToMavenLocal`.