

Kotlin Development Laptop Setup Based on GitVerse

1. Purpose

This document describes how to prepare a laptop for Kotlin development using GitVerse as the main code hosting and collaboration platform [1].

2. Hardware and OS Requirements

- 64-bit operating system (Windows 10/11, modern Linux distribution, or current macOS)[2].
- Minimum 8 GB RAM (16 GB recommended for Android development and emulators)[2].
- At least 30–40 GB of free disk space for IDEs, SDKs, and project sources[2][3].

3. Base Software

1. Update the operating system and system packages to the latest stable versions.
2. Install a current Java Development Kit (JDK 17 or later is recommended for new Kotlin projects)[4][5].
3. Verify Java installation in a terminal or command prompt with the `java -version` and `javac -version` commands[6].

4. Install Development Tools

4.1 IntelliJ IDEA (Recommended)

- Download and install IntelliJ IDEA Community or Ultimate Edition from JetBrains[6][5].
- During first launch, configure the JDK for projects so Kotlin can compile to JVM[6][4].

4.2 Android Studio (For Android Apps)

- Download and install Android Studio, then install the required Android SDK components and tools[2].
- Verify that an Android Virtual Device (AVD) or physical device is available for running Kotlin Android applications[2].

4.3 Optional: GitVerse-related IDEs and Plugins

- Optionally install tools such as GigaIDE if available in your environment for deeper GitVerse ecosystem integration[7][8].

5. Install and Configure Git

1. Install Git from the official distribution for your operating system[9].
2. Configure user identity:

```
git config --global user.name "Your Name"  
git config --global user.email "you@example.com"
```
3. Generate an SSH key pair (for example with `ssh-keygen`) and keep the private key secure.
4. Test Git installation with `git --version` and by cloning any public repository[9].

6. Connect Laptop to GitVerse

1. Create or log in to a GitVerse account in your browser.
2. Add the generated SSH public key to your GitVerse profile settings for SSH access[9].
3. Create a new repository in GitVerse for Kotlin development or use an existing one[10][11].
4. Clone the repository to the laptop:

```
git clone git@gitverse.ru:org/project.git
```

(SSH) or

```
git clone https://gitverse.ru/org/project.git
```

(HTTPS)[9][10]

7. Configure IDE with GitVerse Repository

1. In IntelliJ IDEA or Android Studio, open the cloned project directory as an existing project[6][5].
2. Ensure Git is enabled as Version Control in IDE settings so that commits and pushes work directly with the GitVerse remote[6][12].
3. Use the built-in VCS tools to commit, create branches, and push changes to GitVerse[9][10].

8. Create and Run a Kotlin Project

8.1 JVM / Desktop Kotlin

1. In IntelliJ IDEA: create a new Kotlin project (JVM, Gradle or IntelliJ build system)[6][4][5].
2. Confirm the project SDK is set to the installed JDK and that the Kotlin plugin is enabled (bundled by default)[3][6].
3. Create a Kotlin source file under `src/main/kotlin` and implement a simple `main` function to validate the setup[4][5].

8.2 Android + Kotlin

1. In Android Studio: create a new project and select Kotlin as the language[2][13].
2. Configure minimum SDK and templates as needed, then run the app on an emulator or device to confirm everything works[2].

9. Working with GitVerse from Terminal

Typical daily commands:

- `git status` – check working tree state[9][11].
- `git add .` – stage changes.
- `git commit -m "Meaningful message"` – commit staged changes.
- `git push origin <branch>` – send commits to the GitVerse repository[9][10].

Use these commands together with IDE tools to keep local code synchronized with GitVerse.

10. Code Quality and Documentation

- Follow official Kotlin coding conventions for formatting and naming in all GitVerse repositories[14].
- For larger projects, integrate documentation tools like Dokka in Gradle to generate HTML or Markdown API documentation from Kotlin sources[15][16].

References

[1] GitVerse Features. GigaIDE – Professional Development Environment.
<https://gitverse.ru/features/gigaide/>

[2] Android Developers. Download and Install Android Studio. (2025).
<https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio>

[3] JetBrains. Set up a Kotlin/JS project. Kotlin Language Documentation. (2025).
<https://kotlinlang.org/docs/js-project-setup.html>

- [4] JetBrains. Tutorial: Create Your First Kotlin Application. IntelliJ IDEA Documentation. (2025). <https://www.jetbrains.com/help/idea/create-your-first-kotlin-app.html>
- [5] JetBrains. Get Started with Kotlin. IntelliJ IDEA Documentation. (2025). <https://www.jetbrains.com/help/idea/get-started-with-kotlin.html>
- [6] GeeksforGeeks. Kotlin Environment Setup with IntelliJ IDEA. (2019). <https://www.geeksforgeeks.org/kotlin/kotlin-environment-setup-with-intellij-idea/>
- [7] GitVerse. Что такое IDE (Интегрированная среда разработки) в программировании. Blog. (2024). <https://gitverse.ru/blog/articles/development/131-что-такое-ide-integrirovannaya-sreda-razrabotki-v-programmirovanii>
- [8] GitVerse. IntelliJ IDEA: Обзор и настройка популярной среды разработки. Blog. (2025). <https://gitverse.ru/blog/articles/development/560-intellij-idea-что-это-за-среда-разработки>
- [9] GitVerse. Работа с терминалом. Knowledge Base. (2024). https://gitverse.ru/docs/knowledge-base/working_with_terminal/
- [10] GitVerse. vxll03/GitVerse_guide. Repository. (2025). https://gitverse.ru/vxll03/GitVerse_guide
- [11] GitVerse Wiki. Памятка по GIT (основы, командная строка). (2025). <https://gitverse.ru/Beetlebassist/Beetlebassist/wiki>
- [12] Stack Overflow. Configure Kotlin to an Existing Project in IntelliJ. (2018). <https://stackoverflow.com/questions/52961491/configure-kotlin-to-an-existing-project-in-intellij>
- [13] GitVerse Blog. Программирование на Android: какие языки использовать. (2025). <https://gitverse.ru/blog/articles/development/557-yazyki-dlya-android-razrabotchikov>
- [14] JetBrains. Coding Conventions. Kotlin Documentation. (2025). <https://kotlinlang.org/docs/coding-conventions.html>
- [15] GitHub. Kotlin/dokka: API Documentation Engine for Kotlin. (2014–present). <https://github.com/Kotlin/dokka>
- [16] DEV Community. How To Document A Kotlin Project. (2019). <https://dev.to/cjbrooks12/how-to-document-a-kotlin-project-edc>