Java Setup on Linux: A Complete Tutorial

This tutorial will guide you through checking for Java installation and setting up the JAVA_HOME environment variable on Linux (Ubuntu/Debian-based systems).

Prerequisites

- A Linux system (Ubuntu/Debian)
- Terminal access
- sudo privileges (for installation only)

Step 1: Check if Java is Installed

First, check if Java Runtime Environment (JRE) is installed:

```bash
java -version
```

Expected output if Java is installed:

```
openjdk version "21.0.10" 2026-01-20
OpenJDK Runtime Environment (build 21.0.10+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 21.0.10+7-Ubuntu-124.04, mixed mode, sharing)
```

If Java is not installed, you'll see:

```
Command 'java' not found
```

Step 2: Check if Java Development Kit (JDK) is Installed

To compile Java programs, you need the JDK (not just JRE):

```bash
javac -version
```

Expected output if JDK is installed:

```
javac 21.0.10
```

If only JRE is installed:

```
javac: command not found
```

Step 3: Install Java (if needed)

Install JRE only (for running Java applications):

```bash
sudo apt update
sudo apt install default-jre
```

Install JDK (for developing Java applications):

```bash
sudo apt update
sudo apt install default-jdk
```

Install a specific OpenJDK version:

```bash
sudo apt install openjdk-21-jdk
```

Step 4: Locate Your Java Installation

Find where Java is installed on your system:

```bash
which java
```

This shows the executable location (usually `/usr/bin/java`).

To find the actual installation directory (following symlinks):

```bash
readlink -f $(which java)
```

Example output:

```
/usr/lib/jvm/java-21-openjdk-amd64/bin/java
```

The JAVA_HOME path is everything **before** `/bin/java`, so in this case:

```
/usr/lib/jvm/java-21-openjdk-amd64
```

Step 5: Check if JAVA_HOME is Set

```bash
echo $JAVA_HOME
```

If nothing is displayed, JAVA_HOME is not set.

Step 6: Set JAVA_HOME Environment Variable

Method 1: Using echo commands

```bash
echo 'export JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64' >> ~/.bashrc
echo 'export PATH=$JAVA_HOME/bin:$PATH' >> ~/.bashrc
source ~/.bashrc
```

Method 2: Manual editing

Open your `.bashrc` file:

```bash
nano ~/.bashrc
```

Add these lines at the end:

```bash
export JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
export PATH=$JAVA_HOME/bin:$PATH
```

Save and exit (Ctrl+X, then Y, then Enter).

Apply the changes:

```bash
source ~/.bashrc
```

Step 7: Verify JAVA_HOME is Set

```bash
echo $JAVA_HOME
```

Expected output:

```
/usr/lib/jvm/java-21-openjdk-amd64
```

Useful Java Commands Reference

Command	Purpose
`java -version`	Check Java runtime version
`javac -version`	Check Java compiler version
`which java`	Find Java executable location
`readlink -f $(which java)`	Find actual Java installation path
`echo $JAVA_HOME`	Check JAVA_HOME variable
`update-alternatives --list java`	List all Java installations
`update-java-alternatives -l`	List Java alternatives (Debian/Ubuntu)

When Do You Need JAVA_HOME?

You need to set JAVA_HOME if you're using:

- **Maven** - Build automation tool
- **Gradle** - Build automation tool
- **Tomcat** - Web server
- **Android Studio** - Android development
- **Eclipse/IntelliJ IDEA** - IDEs (sometimes)
- **Jenkins** - CI/CD tool
- Other Java development tools

For simply running Java applications, JAVA_HOME is usually optional.

Troubleshooting

Multiple Java Versions Installed

List all installed versions:

```bash
update-alternatives --config java
```

Select your preferred version from the menu.

JAVA_HOME Not Persisting After Reboot

Make sure you added the export commands to `~/.bashrc` (for bash) or `~/.zshrc` (for zsh), not just ran them in the terminal.

Permission Denied Errors

If you get permission errors when installing Java:

```bash
bash

sudo apt update
sudo apt install openjdk-21-jdk
```

Make sure to use `sudo` for installation commands.

Summary

You've successfully:

1. ✓ Checked for Java installation
2. ✓ Located your Java installation path
3. ✓ Set the JAVA_HOME environment variable
4. ✓ Verified your Java setup

Your Linux system is now properly configured for Java development!