

Developer Environment Setup Documentation

Objective

This document provides a detailed account of the steps taken to set up a developer environment for software engineering projects. It includes the installation of an operating system, IDE, version control system, programming languages, package managers, databases, and additional tools necessary for development.

1. Select Your Operating System (OS)

- **Action:** Downloaded the Windows 11 ISO file and created a bootable USB drive using the Windows Media Creation Tool.
- **Installation:** Installed Windows 11 by booting from the USB drive, followed on-screen instructions, and completed the installation process.
- **Configuration:** Activated Windows using a valid product key. Set up user profiles and configured system settings including updates and driver installations.
- **Troubleshooting:** Ensured compatibility of hardware drivers and resolved any driver-related issues.

2. Install a Text Editor or Integrated Development Environment (IDE)

- **Website:** Visual Studio Code Download
- **Action:** Downloaded and installed Visual Studio Code from the official website.
- **Configuration:** Customized the editor settings, such as enabling auto-save and configuring the integrated terminal. Installed key extensions like Python, and Prettier for enhanced functionality.
- **Troubleshooting:** Ensured VS Code was added to the system PATH for easy command-line access and verified the installation by launching the editor.

3. Set Up Version Control System

- **Website for Git:** Git Download
- **Website for GitHub:** GitHub Sign Up
- **Action:** Downloaded and installed Git. Created a GitHub account for repository management.
- **Configuration:** Configured Git with my name and email using the following commands:

  Bash:

  git config --global user.name "Your Name"
  git config --global user.email "youremail@example.com"

- **Initialization:** Initialized a local Git repository and made the first commit with:

  Bash:

  git init
  git add .
  git commit -m "Initial commit"

- **Troubleshooting:** Generated an SSH key pair and added the public key to my GitHub account to enable secure connections. Verified the setup by cloning a repository.

4. Install Necessary Programming Languages and Runtimes

- **Website:** Python Download
- **Action:** Downloaded and installed Python from the official website.
- **Configuration:** Added Python to the system PATH during installation. Verified the installation by checking the version:

Bash:

python --version

- **Troubleshooting:** Ensured pip was included in the installation and confirmed its functionality by installing a test package, such as requests.

5. Install Package Managers

- **Action:** Verified that pip, the Python package manager, was installed and operational:

Bash:

pip --version

- **Configuration:** Installed necessary packages for my project using pip and created a requirements.txt file to manage dependencies.

Bash:

pip install -r requirements.txt

- **Troubleshooting:** Configured proxy settings if required and resolved any issues with package installations.

6. Configure a Database (MySQL)

- **Website:** MySQL Download
- **Action:** Downloaded the MySQL installer and installed the database server.
- **Configuration:** During the setup, configured the root user password and set up MySQL Workbench for database management.

- **Verification:** Verified the installation by connecting to the MySQL server using MySQL Workbench.
- **Troubleshooting:** Addressed any connection issues and configured user permissions as necessary.

7. Set Up Development Environments and Virtualization (Optional)

- **Website:** Docker Download
- **Action:** Installed Docker Desktop for containerized development environments.
- **Configuration:** Configured Docker to start at system boot and pulled a basic image to test:

  Bash:

  docker pull hello-world

- **Troubleshooting:** Resolved issues related to Docker daemon and user permissions.

8. Explore Extensions and Plugins

- **Extensions Installed:**
  - **Python by Microsoft:** For Python development and debugging.
  - **Prettier:** For code formatting and consistency.
- **Configuration:** Customized settings for each extension to match my workflow.
- **Verification:** Tested each extension to ensure proper integration and functionality within VS Code.

- **Troubleshooting:** Managed extension conflicts and updated extensions to their latest versions.

Summary

This document outlines the complete process of setting up a robust developer environment. It includes the installation and configuration of essential tools such as the operating system, IDE, version control, programming languages, package managers, and databases. The setup is now equipped to support efficient development and collaboration.