Windows Setup Guide for Ruby on Rails Development

Ruby 3 + Rails 8 + VS Code

This guide will walk you through setting up a complete Ruby on Rails development environment on **Windows 11** from scratch.

Estimated Time: 45-60 minutes

Note: This guide is specifically for Windows 11. If you're using Windows 10, the steps are similar but you may need to enable WSL2 manually through Windows Features.

Table of Contents

Why WSL2?

Ruby on Rails works best in a Unix-like environment. WSL2 (Windows Subsystem for Linux) gives you a real Linux environment inside Windows, making Rails development much smoother and closer to production environments.

Step 1: Install WSL2

1.1 Enable WSL

Method 1 - Using Search:

1. Press the **Windows key** on your keyboard
2. Type **"Terminal"**
3. Right-click on **Terminal** in the results
4. Select **"Run as administrator"**

Method 2 - Using Windows Terminal directly:

1. Press **Windows key + X**
2. Select **"Terminal (Admin)"** from the menu

In the Terminal window, run this command:

```powershell
wsl --install
```

Important: You may see one of two outcomes:

Outcome A: The command installs WSL and Ubuntu together, then asks you to restart.

- If this happens, **restart your computer** and skip to step 1.2

Outcome B: The command only installs WSL features, with a message like "The requested operation is successful. Changes will not be effective until the system is rebooted."

- If this happens:
  1. **Restart your computer**
  2. After restart, open **Terminal as administrator** again (using Method 1 or 2 above)
  3. Run the command again:

```powershell
    wsl --install -d Ubuntu
```

4. This will now install Ubuntu specifically
5. **You may need to restart again** if prompted

1.2 Complete Ubuntu Setup

1. After your final restart, Ubuntu should automatically open and begin installation
   - **If Ubuntu doesn't open automatically:** Press **Windows key**, type **"Ubuntu"**, and click it

    - You'll see "Installing, this may take a few minutes..." - be patient!

2. Once installation completes, you'll be asked to create a username and password:
    - Choose a username (lowercase, no spaces - e.g., `john` or `developer`)
    - Choose a password (you won't see it as you type - this is normal)
    - Re-enter your password to confirm
    - **Remember these credentials!** You'll need them frequently

Tip for Windows 11: You can now access Ubuntu anytime by:

- Pressing **Windows key** and typing **"Ubuntu"**, OR
- Opening **Windows Terminal** and clicking the dropdown arrow (∨), then selecting **"Ubuntu"**

1.3 Update Ubuntu

Once you see your username in the terminal (like `john@DESKTOP-ABC:~$`), run:

```bash
sudo apt update && sudo apt upgrade -y
```

Enter your password when prompted.

Step 2: Install Ruby 3.3

We'll use **rbenv**, which makes managing Ruby versions easy.

2.1 Install Dependencies

Copy and paste this entire block into your Ubuntu terminal:

```bash
sudo apt install -y git curl libssl-dev libreadline-dev zlib1g-dev \
  autoconf bison build-essential libyaml-dev libreadline-dev \
  libncurses5-dev libffi-dev libgdbm-dev
```

2.2 Install rbenv and ruby-build

```bash
curl -fsSL https://github.com/rbenv/rbenv-installer/raw/HEAD/bin/rbenv-installer | bash
```

2.3 Configure Your Shell

Add rbenv to your shell by running:

```bash
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(rbenv init -)"' >> ~/.bashrc
source ~/.bashrc
```

2.4 Install Ruby 3.3

```bash
rbenv install 3.3.6
rbenv global 3.3.6
```

This will take 5-10 minutes. Be patient!

2.5 Verify Ruby Installation

```bash
ruby -v
```

You should see something like: `ruby 3.3.6 (2024-11-05 revision ...)`

Step 3: Install Rails 8

3.1 Install Rails

```bash
gem install rails -v 8.0.1
```

3.2 Rehash rbenv

```bash
rbenv rehash
```

3.3 Verify Rails Installation

```bash
rails -v
```

You should see: `Rails 8.0.1`

Step 4: Install SQLite

SQLite is a lightweight database that's perfect for development. It requires minimal setup!

4.1 Install SQLite

```bash
sudo apt install -y sqlite3 libsqlite3-dev
```

4.2 Verify SQLite Installation

```bash
sqlite3 --version
```

You should see the SQLite version (e.g., `3.37.2 2022-01-06...`)

That's it! SQLite is ready to use. Unlike MySQL, there's no service to start or configure - Rails will create and manage the database files automatically.

Step 5: Install VS Code and Extensions

5.1 Install VS Code on Windows

1. Download VS Code from: https://code.visualstudio.com/
2. Run the installer
3. **Important:** During installation, check these boxes:
   - ✅ Add "Open with Code" action to Windows Explorer file context menu
   - ✅ Add "Open with Code" action to Windows Explorer directory context menu
   - ✅ Add to PATH

5.2 Install WSL Extension

1. Open VS Code
2. Click the Extensions icon (or press `Ctrl+Shift+X`)
3. Search for "WSL"
4. Install the **"WSL"** extension by Microsoft

5.3 Connect VS Code to WSL

1. In VS Code, press `Ctrl+Shift+P`
2. Type "WSL: Connect to WSL"
3. Press Enter
4. A new VS Code window will open connected to your Ubuntu environment
5. **IMPORTANT:** Look at the bottom-left corner of VS Code - you should see a green icon that says **"WSL: Ubuntu"**
   - If you see this, you're connected correctly!
   - If you don't see this, repeat steps 1-3

From now on, always make sure you see "WSL: Ubuntu" in the bottom-left corner when working on Rails projects.

5.4 Install Recommended Extensions (in WSL)

With VS Code connected to WSL, install these extensions:

1. **Ruby LSP** (by Shopify) - Ruby language support
2. **ERB Helper Tags** (by Raymon Schouwenaar) - For Rails views
3. **Rails** (by bung87) - Rails snippets and helpers
4. **SQLite Viewer** (by Florian Klampfer) - View SQLite databases (optional but helpful)
5. **GitLens** (by GitKraken) - Git integration (optional but helpful)

To install:

1. Press `Ctrl+Shift+X` to open Extensions
2. Search for each extension name
3. Click **"Install"**
   - If you see "Install in WSL: Ubuntu" as an option, click that

- If you only see "Install", that's fine - click it (as long as you see "WSL: Ubuntu" in the bottom-left corner of VS Code, the extension will install in WSL)

Note: As long as VS Code shows "WSL: Ubuntu" in the bottom-left corner, all extensions you install will automatically be installed in the WSL environment.

Step 6: Verify Your Setup

Let's make sure everything is working!

6.1 Open Ubuntu Terminal in VS Code

1. In VS Code (connected to WSL), press `` Ctrl+` `` (backtick) to open the terminal
2. You should see your Ubuntu prompt

6.2 Check All Versions

Run these commands:

```bash
ruby -v
rails -v
sqlite3 --version
node -v
```

Expected output:

- Ruby: 3.3.x
- Rails: 8.0.x
- SQLite: 3.x.x
- Node: (if installed, if not, it's okay - Rails 8 includes its own JS runtime)

6.3 Check SQLite

```bash
sqlite3
```

If you see `sqlite>`, it's working! Type `.quit` to exit.

Step 7: Creating Your First Rails App

Let's create a test app to make sure everything works!

7.1 Navigate to Your Projects Folder

```bash
cd ~
mkdir projects
cd projects
```

7.2 Create a New Rails App

```bash
rails new test_app
```

This creates a new Rails app called `test_app`. SQLite is the default database for Rails, so we don't need to specify it!

7.3 Open the App in VS Code

```bash
cd test_app
code .
```

A new VS Code window will open with your Rails app!

7.4 Create the Database

In the VS Code terminal (`` Ctrl+` ``):

```bash
rails db:create
```

You should see:

```
Created database 'db/development.sqlite3'
Created database 'db/test.sqlite3'
```

Note: With SQLite, the databases are just files in your `db/` folder - much simpler than MySQL!

7.5 Start the Rails Server

```bash
rails server
```

7.6 View Your App

1. Open your Windows web browser
2. Go to: http://localhost:3000
3. You should see the Rails welcome page! 🎉

Press `Ctrl+C` in the terminal to stop the server.

Troubleshooting

WSL installation issues

If `wsl --install` doesn't work or gives errors:

1. **Check Windows version:** You need Windows 11 (or Windows 10 version 2004 or higher)
   - Press **Windows key + R**, type `winver`, press Enter
2. **Enable Virtualization in BIOS:** WSL requires CPU virtualization
   - Restart your computer and enter BIOS (usually press F2, F10, Del, or Esc during startup)
   - Look for "Virtualization Technology" or "Intel VT-x" or "AMD-V"
   - Enable it, save, and restart
3. **Manual installation:** If automatic installation fails:

```powershell
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

Restart, then run `wsl --install -d Ubuntu`

Ubuntu doesn't appear after installation

1. Press **Windows key**, type **"Ubuntu"**, and launch it manually
2. Or open **Terminal** and select **Ubuntu** from the dropdown
3. If still not found, run in Terminal (Admin): `wsl --install -d Ubuntu`

SQLite errors when creating database

Make sure SQLite is installed:

```bash
sudo apt install -y sqlite3 libsqlite3-dev
```

"Could not find gem 'sqlite3'" error

Install the sqlite3 gem:

```bash
gem install sqlite3
bundle install
```

Rails server won't start - "Address already in use"

Something is already running on port 3000:

```bash
rails server -p 3001
```

Then visit http://localhost:3001

VS Code can't find Ruby/Rails

Make sure you've:

1. Connected to WSL (look for "WSL: Ubuntu" in bottom-left corner)
2. Opened the terminal in WSL (`Ctrl+``)
3. Run `source ~/.bashrc` in the terminal

Extensions not working or can't find Ruby

1. Check that "WSL: Ubuntu" is shown in the bottom-left corner of VS Code
2. Close and reopen VS Code, then reconnect to WSL (`Ctrl+Shift+P` → "WSL: Connect to WSL")

3. Reinstall the Ruby LSP extension while connected to WSL
4. Restart VS Code completely (close all windows and reopen)

Ruby version is wrong

```bash
rbenv global 3.3.6
rbenv rehash
ruby -v
```

Quick Reference Commands

Starting Your Work Day

```bash
# Navigate to your project
cd ~/projects/your_project_name

# Open in VS Code
code .

# Start Rails server
rails server
```

Common Rails Commands

```bash
# Create a new Rails app (SQLite is default)
rails new app_name

# Create a new Rails app with specific database
rails new app_name -d postgresql  # if you want PostgreSQL instead

# Create database
rails db:create

# Run migrations
rails db:migrate

# Seed database
rails db:seed

# Start server
rails server

# Generate a model
rails generate model ModelName

# Generate a controller
rails generate controller ControllerName

# Open Rails console
rails console

# Run tests
rails test
```

SQLite Commands

```bash
# Open SQLite console for development database
sqlite3 db/development.sqlite3

# View tables (in SQLite console)
.tables

# View schema (in SQLite console)
.schema

# Exit SQLite console
.quit
```

Working with Files

Understanding WSL File System

Your Ubuntu files are located at:

- **In Windows File Explorer:** Type `\\wsl$\Ubuntu\home\YOUR_USERNAME\` in the address bar

    - Or: Open File Explorer, look for **"Linux"** in the left sidebar (Windows 11)
- **In Ubuntu:** `~/` (which is `/home/YOUR_USERNAME/`)

Important: Always edit Rails files using VS Code connected to WSL, not directly from Windows!

Opening VS Code from Terminal

```bash
# Open current directory
code .

# Open specific file
code filename.rb

# Open specific directory
code ~/projects/my_app
```

Tips for Success

1. **Always work in WSL** - Your Ubuntu terminal and VS Code connected to WSL
2. **No database service needed** - SQLite doesn't require starting any services, it just works!
3. **Use `source ~/.bashrc`** - If commands aren't found, run this to reload your environment
4. **Git is pre-installed** - You can use Git commands immediately in Ubuntu
5. **Copy/Paste in Terminal:**
    - Copy: Select text, then right-click
    - Paste: Right-click in terminal
6. **Database files** - Your SQLite databases are stored as `.sqlite3` files in the `db/` folder
7. **Google is your friend** - If you get an error, copy it and search for it!

Next Steps

Now you're ready to start the technical assignment! Remember:

1. Create a new Rails app: `rails new your_app_name`
2. Run `rails db:create` to create the SQLite databases
3. Start coding!

Good luck with your assignment! 🚀

Additional Resources

- **Rails Guides:** https://guides.rubyonrails.org/
- **Ruby Documentation:** https://ruby-doc.org/
- **Rails API Documentation:** https://api.rubyonrails.org/
- **Stack Overflow:** https://stackoverflow.com/questions/tagged/ruby-on-rails

---

If you encounter any issues not covered here, don't hesitate to reach out for help!