Python Session Prerequisites - Windows Setup Guide

Overview

This document provides step-by-step setup instructions for Windows users Complete all installations and preparations before the session to maximize learning time.

Required Software Installation

1. Python 3.13 Installation

Install via Microsoft Store (Recommended)

1. Open Microsoft Store
2. Search for "Python 3.13"
3. Install the official Python package from Python Software Foundation
4. The installation will automatically handle PATH configuration

Verify Installation

Open Command Prompt (cmd) and run:

```
python --version
pip --version
```

Expected output: Python 3.13.x and pip version information

Why Microsoft Store?

- Automatic PATH configuration
- Easy updates and management
- No administrator privileges required
- Cleaner installation process

2. Podman Installation

Download Podman Desktop

1. Visit: https://podman.io/getting-started/installation
2. Download "Podman Desktop for Windows"
3. Run the installer as Administrator

4. Follow the installation wizard
5. Restart your computer after installation

Verify Podman Installation

Open Command Prompt and run:

```
podman --version
```

Expected output: Podman version information

Initialize Podman Machine

```
podman machine init
podman machine start
```

3. PostgreSQL Setup via Podman

Pull PostgreSQL Image

```
podman pull postgres:15
```

Create and Start PostgreSQL Container

```
podman run --name postgres-dev -e POSTGRES_PASSWORD=mypassword -e
POSTGRES_DB=fastapi_db -p 5432:5432 -d postgres:15
```

Verify PostgreSQL is Running

```
podman ps
```

You should see the postgres-dev container running

Commands to Remember

- **Start PostgreSQL**: podman start postgres-dev
- **Stop PostgreSQL**: podman stop postgres-dev
- **Check status**: podman ps
- **View logs**: podman logs postgres-dev

4. pgAdmin Installation

Download pgAdmin

1. Visit: https://www.pgadmin.org/download/pgadmin-4-windows/

2. Download the latest Windows installer
3. Run the installer as Administrator
4. Follow the installation wizard

Setup pgAdmin

1. Launch pgAdmin from Start Menu
2. Set a master password when prompted
3. Add a new server connection:
    a. **Name**: Local PostgreSQL
    b. **Host**: localhost
    c. **Port**: 5432
    d. **Username**: postgres
    e. **Password**: mypassword (or what you set above)

5. Postman Installation

Download Postman

1. Visit: https://www.postman.com/downloads/
2. Download Postman for Windows
3. Run the installer
4. Create a free Postman account (recommended)

Basic Postman Setup

1. Create a new workspace called "FastAPI Learning"
2. Create a new collection called "FastAPI Endpoints"
3. Test with a simple GET request to https://httpbin.org/get

6. Code Editor Setup (Optional but Recommended)

Visual Studio Code

1. Download from: https://code.visualstudio.com/
2. Install the following extensions:
    a. Python
    b. Python Debugger
    c. REST Client (alternative to Postman)
    d. PostgreSQL (for database queries)

7. Jupyter Notebook Installation

Install Jupyter via pip

Open Command Prompt and run:

```
pip install jupyter notebook
```

Verify Jupyter Installation

```
jupyter --version
```

Expected output: Jupyter version information

Launch Jupyter Notebook

```
jupyter notebook
```

This will:

- Start the Jupyter server
- Automatically open your default browser
- Display the Jupyter dashboard at http://localhost:8888

Install JupyterLab (Optional - Modern Interface)

JupyterLab is a more feature-rich interface:

```
pip install jupyterlab
```

Launch JupyterLab:

```
jupyter lab
```

Essential Jupyter Extensions

Install useful extensions for better experience:

```
pip install jupyter_contrib_nbextensions
jupyter contrib nbextension install --user
```

Jupyter in VS Code (Recommended)

If you're using VS Code:

1. Install the "Jupyter" extension from VS Code marketplace
2. Open or create a .ipynb file
3. Select Python 3.13 as your kernel

4. You can now run Jupyter notebooks directly in VS Code

Test Your Jupyter Setup

1. Open Jupyter Notebook or JupyterLab
2. Create a new Python 3 notebook
3. Run this test code:

```
print("Hello, Jupyter!")
import sys
print(f"Python version: {sys.version}")
```

4. Verify the output displays correctly

Useful Jupyter Commands

- **Start notebook**: jupyter notebook
- **Start lab**: jupyter lab
- **List running servers**: jupyter notebook list
- **Stop server**: Press Ctrl+C in the terminal where Jupyter is running

Self-Study Materials (Complete Before Session)

1. Client-Server Architecture

Study Topics:

- What is client-server architecture?
- Role of client vs server
- How web browsers communicate with servers
- Request-response cycle

Recommended Resources:

- Read: "Client-Server Architecture Explained" (search online)
- Watch: YouTube videos on "Client Server Architecture basics"
- Understand: How your browser loads a webpage

2. Backend Development Fundamentals

Key Concepts to Understand:

- What is backend development?
- Difference between frontend and backend
- Role of APIs in connecting frontend and backend
- What is a web server?
- What is an application server?

3. HTTP Protocol Basics

Must Know:

- HTTP methods: GET, POST, PUT, DELETE
- HTTP status codes: 200, 201, 400, 404, 500
- What are HTTP headers?
- Request and response structure

4. Database Fundamentals

Basic Concepts:

- What is a database?
- Relational vs Non-relational databases

- What is SQL?
- Basic database operations (CRUD)
- Primary keys and foreign keys

5. API Concepts

Understanding APIs:

- What is an API?
- REST API principles
- JSON data format
- API endpoints and routes

☑ Pre-Session Verification Checklist

Environment Check

Run these commands in Command Prompt to verify your setup:

```
# Check Python
python --version

# Check pip
pip --version

# Check Podman
podman --version

# Check if PostgreSQL container is running
podman ps

# Test PostgreSQL connection (should connect without error)
podman exec -it postgres-dev psql -U postgres -d fastapi_db

# Check Jupyter
jupyter --version
```

Database Connection Test

1. Open pgAdmin
2. Connect to your Local PostgreSQL server
3. Verify you can see the fastapi_db database
4. Create a test table to ensure write permissions

Postman Test

1. Open Postman
2. Create a GET request to https://httpbin.org/get
3. Send the request and verify you get a 200 response

Jupyter Notebook Test

1. Open Command Prompt
2. Run jupyter notebook or jupyter lab

3. Verify browser opens automatically with Jupyter interface
4. Create a new Python 3 notebook
5. Run a test cell with print("Setup complete!")
6. Verify the output displays correctly

🔧 Troubleshooting Common Issues

Python Installation Issues

Problem: "python is not recognized as internal or external command" **Solution**:

1. Reinstall Python with "Add to PATH" checked
2. Or manually add Python to PATH in Environment Variables

Podman Issues

Problem: "podman machine start" fails **Solution**:

1. Ensure Hyper-V is enabled (Windows Features)
2. Run Command Prompt as Administrator
3. Try: podman machine stop then podman machine start

PostgreSQL Connection Issues

Problem: Can't connect to PostgreSQL from pgAdmin **Solution**:

1. Verify container is running: podman ps
2. Check if port 5432 is available: netstat -an | findstr 5432
3. Restart container: podman restart postgres-dev

Postman Issues

Problem: Requests timing out **Solution**:

1. Check Windows Firewall settings
2. Try disabling antivirus temporarily
3. Use "Postman Agent" if web version doesn't work