Pazago Drive — Pre-requisites Setup Checklist

Welcome to the Pazago Drive! This checklist will help you set up your development environment and prepare for building AI agents with Mastra. **Please complete ALL items before your scheduled session.**

📋 Setup Checklist

✅ 1. Node.js and TypeScript Setup

Install Node.js

- [ ] Visit https://nodejs.org/
- [ ] Download the **LTS (Long Term Support)** version (recommended for most users)
- [ ] Run the installer and follow the installation wizard
- [ ] Verify installation by opening your terminal/command prompt and running:

```
node --versionnpm --version
```

- [ ] You should see version numbers displayed (Node.js 18+ recommended)

Install TypeScript

- [ ] Open your terminal/command prompt
- [ ] Install TypeScript globally using npm:

```
npm install -g typescript
```

- [ ] Verify TypeScript installation:

```
tsc --version
```

- [ ] You should see the TypeScript version number

Alternative: Using Package Managers

For Windows users (optional):

- [ ] Install using Chocolatey: `choco install nodejs`
- [ ] Install using Winget: `winget install OpenJS.NodeJS`

For macOS users (optional):

- [ ] Install using Homebrew: `brew install node`

For Linux users (optional):

- [ ] Install using package manager (e.g., `sudo apt install nodejs npm` for Ubuntu)

✅ 2. GitHub Account Setup

Create GitHub Account

- [ ] Visit https://github.com/
- [ ] Click "Sign up" and create your account
- [ ] Choose a professional username (this will be visible to employers)
- [ ] Verify your email address

Install Git

- [ ] Download Git from https://git-scm.com/downloads
- [ ] Install Git on your system
- [ ] Configure Git with your details:

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

- [ ] Verify Git installation: `git --version`

GitHub Authentication Setup

- [ ] Set up SSH keys for GitHub (recommended) by following GitHub's SSH guide
- [ ] OR configure personal access token for HTTPS authentication

✅ 3. Postman Setup

Install Postman

- [ ] Visit https://www.postman.com/downloads/
- [ ] Download Postman for your operating system
- [ ] Install and launch Postman
- [ ] Create a free Postman account (optional but recommended)
- [ ] Familiarize yourself with the interface

Postman Basics

- [ ] Learn how to create a new request
- [ ] Understand GET, POST, PUT, DELETE methods
- [ ] Practice adding headers and request bodies
- [ ] Learn how to save requests in collections

✅ 4. PostgreSQL and PgVector Setup

Install PostgreSQL

- [ ] Download PostgreSQL from https://www.postgresql.org/download/
- [ ] Install PostgreSQL 14+ (required for PgVector extension)
- [ ] During installation, remember your postgres user password
- [ ] Verify installation by running: `psql --version`

Install PgVector Extension

- [ ] **For macOS (Homebrew):**

```
brew install pgvector
```

- [ ] **For Ubuntu/Debian:**

```
sudo apt install postgresql-14-pgvector
```

- [ ] **For Windows:** Download pre-built binaries or compile from source
- [ ] **Alternative:** Use Docker with PgVector pre-installed:

```
docker run -d --name pgvector-db -p 5432:5432 -e POSTGRES_PASSWO
RD=password pgvector/pgvector:pg16
```

Database Setup

- [ ] Create a new database for your Mastra project:

```
CREATE DATABASE mastra_rag_db;
```

- [ ] Connect to your database and enable the vector extension:

```
\c mastra_rag_dbCREATE EXTENSION vector;
```

- [ ] Verify PgVector installation:

```
SELECT * FROM pg_extension WHERE extname = 'vector';
```

Database Client Tools

- [ ] Install a PostgreSQL client:
  - [ ] **pgAdmin** (GUI): https://www.pgadmin.org/
  - [ ] **DBeaver** (GUI): https://dbeaver.io/
  - [ ] **psql** (Command line - comes with PostgreSQL)
- [ ] Test connection to your database

- [ ] Practice basic SQL queries and vector operations

✅ 5. OpenAI Developer Account

Create OpenAI Account

- [ ] Visit https://platform.openai.com/
- [ ] Click "Sign up" and create your account
- [ ] Verify your email address and phone number
- [ ] Complete the account setup process

API Key Setup

- [ ] Navigate to https://platform.openai.com/api-keys
- [ ] **If you have credits/payment method:** Click "Create new secret key" and securely store it
- [ ] **If using provided keys:** Skip this step - keys will be provided during the session
- [ ] **Important:** Never share your API key or commit it to version control

Billing Setup (Required)

- [ ] **Add a payment method** to your OpenAI account (credit/debit card required)
- [ ] **Important:** OpenAI no longer provides free credits for new accounts
- [ ] Set a low spending limit (e.g., $10-20) to control costs during the session
- [ ] Review the pricing page to understand costs
- [ ] **Estimated session cost:** $2-5 for typical API usage during learning
- [ ] Check your usage at https://platform.openai.com/usage

Alternative Option

- [ ] **If you don't receive free credits or prefer not to add a payment method:**
  - [ ] Still create your OpenAI account and verify it
  - [ ] We will provide API keys on the day of your session

- [ ] No additional setup required - just have your account ready

✅ 5. Development Environment

Code Editor Setup

- [ ] Install **Visual Studio Code** from https://code.visualstudio.com/
- [ ] Install recommended extensions:
  - [ ] TypeScript and JavaScript Language Features
  - [ ] Prettier - Code formatter
  - [ ] ESLint
  - [ ] GitLens
  - [ ] REST Client (for API testing)
  - [ ] **PostgreSQL** (by Chris Kolkman) - for database management
  - [ ] **SQL Tools** - for database queries and connections

Terminal Setup

- [ ] Ensure you have a good terminal application
- [ ] Windows: Use PowerShell, Command Prompt, or Windows Terminal
- [ ] macOS: Use Terminal or iTerm2
- [ ] Linux: Use your preferred terminal emulator

✅ 6. Mastra Documentation and Resources

Mastra Documentation

- [ ] Bookmark the Mastra documentation: https://docs.mastra.ai/
- [ ] Read the "Getting Started" guide
- [ ] Explore the RAG (Retrieval-Augmented Generation) section
- [ ] Review code examples and tutorials
- [ ] Join the Mastra community Discord/forum if available

Essential Reading

- [ ] **Read the complete "Principles of Building AI Agents" book** (PDF provided)
  - Focus especially on:
  - [ ] Part I: Prompting a Large Language Model (LLM)
  - [ ] Part II: Building an Agent
  - [ ] Part IV: Retrieval-Augmented Generation (RAG)
  - [ ] Part VII: Development & Deployment

Mastra Framework Preparation

- [ ] Understand what Mastra is and its core concepts
- [ ] Learn about agents, tools, and workflows
- [ ] Familiarize yourself with RAG pipelines
- [ ] Review vector databases and embedding concepts

✅ 7. Additional Preparations

Knowledge Prerequisites

Core Programming Concepts:

- [ ] Basic understanding of JavaScript/TypeScript
- [ ] Familiarity with async/await and Promises
- [ ] Understanding of REST APIs and HTTP methods
- [ ] Basic knowledge of JSON data format
- [ ] Understanding of environment variables
- [ ] Object-oriented programming concepts

AI/ML and LLM Concepts:

- [ ] **Large Language Models (LLMs)** - Understanding what they are and how they work
- [ ] **Prompt Engineering** - How to craft effective prompts for AI models

- [ ] **Token limits and context windows** - Understanding input/output constraints
- [ ] **Temperature and sampling parameters** - How they affect AI responses
- [ ] **Retrieval-Augmented Generation (RAG)** - Core concept and architecture
- [ ] **Vector embeddings** - How text is converted to numerical representations
- [ ] **Vector databases** - Storage and retrieval of embeddings
- [ ] **Semantic search** - Finding relevant information based on meaning
- [ ] **Chunking strategies** - Breaking down documents for processing
- [ ] **AI agents and tools** - How agents use external tools and APIs

Mastra-Specific Concepts:

- [ ] **Agents** - Autonomous AI entities that can perform tasks
- [ ] **Workflows** - Structured sequences of AI operations
- [ ] **Tools integration** - How agents interact with external services
- [ ] **Memory systems** - How agents maintain context across interactions
- [ ] **Evaluation frameworks** - Testing and measuring AI performance

Data and Search Concepts:

- [ ] **Information retrieval** - Finding relevant information from large datasets
- [ ] **Document processing** - Parsing and extracting content from various formats
- [ ] **Indexing strategies** - Organizing data for efficient search
- [ ] **Similarity scoring** - Measuring relevance between queries and documents
- [ ] **Metadata handling** - Managing additional information about documents

PostgreSQL and PgVector Concepts:

- [ ] **PostgreSQL basics** - Understanding relational databases and SQL
- [ ] **Vector data types** - How PgVector stores and handles vector data
- [ ] **Vector operations** - Similarity search, distance calculations (L2, cosine, inner product)
- [ ] **Indexing vectors** - HNSW and IVFFlat indexes for efficient vector search

- [ ] **Query optimization** - Writing efficient vector similarity queries
- [ ] **Database schema design** - Structuring tables for RAG applications
- [ ] **Connection pooling** - Managing database connections efficiently
- [ ] **Hybrid search** - Combining vector similarity with traditional SQL filtering

Recommended Pre-Assignment Learning

- [ ] Learn about RAG pipeline components (ingestion, indexing, retrieval, generation)
- [ ] Familiarize yourself with vector similarity concepts (cosine similarity, etc.)
- [ ] Understand the trade-offs between different embedding models
- [ ] **Practice SQL and PostgreSQL**:
  - [ ] Basic SQL queries (SELECT, INSERT, UPDATE, DELETE)
- [ ] **Learn PgVector specifics**:
  - [ ] Vector similarity search queries
  - [ ] Creating and managing vector indexes
  - [ ] Understanding distance metrics (L2, cosine, inner product)
  - [ ] Hybrid search patterns (vector + metadata filtering)

System Requirements Check

- [ ] Ensure you have at least 8GB RAM
- [ ] Check available disk space (minimum 5GB free)
- [ ] Stable internet connection for API calls
- [ ] Administrative privileges to install software

Create Project Directory

- [ ] Create a dedicated folder for your Mastra projects
- [ ] Example: `~/Projects/Mastra` or `C:\Projects\Mastra`

🚀 Ready for Your Session?

Once you've completed all items above, you're ready for your Mastra development session!

✅ Final Verification Checklist:

- [ ] Node.js and TypeScript are installed and working
- [ ] PostgreSQL and PgVector are set up and tested
- [ ] OpenAI account is created and verified
- [ ] **Either:** OpenAI API key created and secured **OR** ready to use provided keys
- [ ] GitHub account is ready with Git configured
- [ ] Postman is installed and functional
- [ ] VS Code is set up with recommended extensions
- [ ] You've read the "Principles of Building AI Agents" book
- [ ] You understand basic AI/ML concepts (LLMs, RAG, vector embeddings)
- [ ] You're familiar with SQL and PostgreSQL basics
- [ ] All tools have been tested and verified working

🎯 What to Bring to Your Session:

- **Laptop with all software installed** (as per this checklist)
- **OpenAI account credentials** (we'll provide API keys if needed)
- **GitHub credentials** ready for use
- **Database client** (pgAdmin/DBeaver) configured and tested
- **Questions or issues** you encountered during setup

📚 Additional Resources

- **Mastra Examples**: GitHub Repository with Examples
- **TypeScript Handbook**: https://www.typescriptlang.org/docs/
- **Node.js Documentation**: https://nodejs.org/en/docs/

- **OpenAI API Documentation**: https://platform.openai.com/docs
- **Prompt Engineering Guide**: https://www.promptingguide.ai/
- **RAG Papers and Tutorials**: Academic papers on Retrieval-Augmented Generation
- **Vector Database Concepts**: Pinecone Learning Center
- **PostgreSQL Documentation**: https://www.postgresql.org/docs/
- **PgVector GitHub Repository**: https://github.com/pgvector/pgvector
- **PgVector Documentation**: https://github.com/pgvector/pgvector#getting-started
- **SQL Tutorial**: https://www.w3schools.com/sql/
- **PostgreSQL Tutorial**: https://www.postgresqltutorial.com/

❓ Need Help?

If you encounter any issues during setup:

1. Check the official documentation for each tool
2. Search for solutions on Stack Overflow
3. Ask questions in the Mastra community
4. **Contact us before your session** if you have setup issues

📞 Support Contact

If you're unable to complete any part of this setup, please reach out **at least 24 hours before your scheduled session** so we can assist you.

Important: Come prepared with everything installed and tested. This will ensure we can focus entirely on learning Mastra and building amazing AI applications during your session! 🎯

Setup Checklist - Version 1.0