

# Setup Guide

---

# Intermediate Development with Node.js

---

## System Requirements

### Minimum Requirements

- **Operating System:** Windows 10, macOS 10.15, or Ubuntu 18.04 (or equivalent Linux distribution)
- **RAM:** 8GB minimum, 16GB recommended
- **Storage:** 10GB free disk space
- **Internet:** Stable broadband connection for downloading dependencies

### Recommended Specifications

- **RAM:** 16GB or more
- **CPU:** Multi-core processor (4+ cores)
- **Storage:** SSD for better performance

## Required Software Installation

### 1. Node.js and npm

#### Installation:

1. Visit [nodejs.org](https://nodejs.org)
2. Download the LTS version (18.x or higher)
3. Run the installer and follow the setup wizard
4. Verify installation:

```
node --version  
npm --version
```

#### Expected Output:

```
v18.17.0 (or higher)  
9.6.7 (or higher)
```

## 2. Code Editor

### VS Code (Recommended):

1. Download from [code.visualstudio.com](https://code.visualstudio.com)
2. Install recommended extensions:
  - JavaScript (ES6) code snippets
  - Node.js Extension Pack
  - Thunder Client (for API testing)
  - Docker (for containerization labs)

### Alternative Editors:

- WebStorm
- Sublime Text
- Atom

## 3. Git Version Control

### Installation:

- **Windows:** Download from [git-scm.com](https://git-scm.com)
- **macOS:** Install via Homebrew: `brew install git` or download from website
- **Linux:** `sudo apt-get install git` (Ubuntu/Debian) or equivalent

### Verification:

```
git --version
```

## 4. Docker Desktop

### Installation:

1. Visit [docker.com/products/docker-desktop](https://docker.com/products/docker-desktop)
2. Download for your operating system
3. Install and start Docker Desktop
4. Verify installation:

```
docker --version  
docker-compose --version
```

## 5. MongoDB

### Option A: MongoDB Atlas (Cloud - Recommended for Course)

1. Create free account at [mongodb.com/atlas](https://mongodb.com/atlas)
2. Create a free cluster
3. Get connection string for later use

### Option B: Local MongoDB Installation

1. Download from [mongodb.com/try/download/community](https://mongodb.com/try/download/community)
2. Follow installation instructions for your OS
3. Start MongoDB service

### Verification:

```
mongosh --version
```

## 6. Redis

### Option A: Redis Cloud (Recommended for Course)

1. Create free account at [redis.com/try-free](https://redis.com/try-free)
2. Create a free database
3. Get connection details for later use

### Option B: Local Redis Installation

### Windows:

```
# Using Chocolatey  
choco install redis-64
```

### macOS:

```
# Using Homebrew
brew install redis
brew services start redis
```

## Linux (Ubuntu/Debian):

```
sudo apt update
sudo apt install redis-server
sudo systemctl start redis-server
```

## Verification:

```
redis-cli ping
# Expected response: PONG
```

## Course Repository Setup

### 1. Clone the course repository:

```
git clone https://github.com/chrisminnick/intermediate-node-dev.git
cd intermediate-node-dev
```

### 2. Run the setup test:

```
cd setup-test
npm install
npm start
```

### 3. Verify setup:

- Open browser to `http://localhost:3000`
- You should see a JSON response with system information
- Press `Ctrl+C` to stop the test server

## Environment Configuration

### 1. Create Global .env Template

Create a file named `.env.example` in your projects folder:

```
# Database Configuration
MONGODB_URI=mongodb://localhost:27017/intermediate-node-dev
MONGODB_TEST_URI=mongodb://localhost:27017/intermediate-node-dev-test

# Redis Configuration
REDIS_HOST=localhost
REDIS_PORT=6379
REDIS_PASSWORD=

# Application Configuration
NODE_ENV=development
PORT=3000

# JWT Configuration
JWT_SECRET=your-super-secret-jwt-key-here
JWT_EXPIRES_IN=24h

# API Keys (for external services)
API_KEY=your-api-key-here
```

## 2. Development Tools Setup

### Install useful global packages:

```
npm install -g nodemon pm2 loadtest artillery
```

- **nodemon** : Automatic server restart during development
- **pm2** : Production process manager
- **loadtest** : Simple load testing tool
- **artillery** : Advanced load testing and performance tool

## Troubleshooting Common Issues

### Node.js Installation Issues

#### Error: "node command not found"

- Restart your terminal/command prompt
- Check if Node.js was added to your system PATH
- Try reinstalling Node.js with administrator privileges

#### Permission Errors with npm (macOS/Linux):

```
# Fix npm permissions
mkdir ~/.npm-global
npm config set prefix '~/.npm-global'
# Add to ~/.bashrc or ~/.zshrc:
export PATH=~/.npm-global/bin:$PATH
```

## MongoDB Connection Issues

### Connection refused errors:

- Ensure MongoDB service is running
- Check firewall settings
- Verify connection string format

## Redis Connection Issues

### Connection timeout:

- Verify Redis server is running: `redis-cli ping`
- Check Redis configuration file
- Ensure proper network access

## Docker Issues

### Docker daemon not running:

- Start Docker Desktop application
- Ensure Docker service is running in background
- Check Docker Desktop troubleshooting guide

## Performance Optimization

### Development Environment

#### 1. Increase Node.js memory limit:

```
node --max-old-space-size=4096 app.js
```

#### 2. Use SSD storage for better I/O performance

#### 3. Configure VS Code for Node.js development:

- Install Node.js debugger

- Configure auto-save
- Set up integrated terminal

## Testing Your Setup

Run this complete verification script:

```
#!/bin/bash

echo "🔍 Verifying Node.js development environment..."

# Check Node.js
node --version && echo "✅ Node.js installed" || echo "❌ Node.js missing"

# Check npm
npm --version && echo "✅ npm installed" || echo "❌ npm missing"

# Check Git
git --version && echo "✅ Git installed" || echo "❌ Git missing"

# Check Docker
docker --version && echo "✅ Docker installed" || echo "❌ Docker missing"

# Check MongoDB (if local)
mongosh --version && echo "✅ MongoDB client installed" || echo "❌ MongoDB client missing"

# Check Redis (if local)
redis-cli ping > /dev/null && echo "✅ Redis running" || echo "❌ Redis not accessible"

echo "🎉 Setup verification complete!"
```

## Getting Help

If you encounter issues during setup:

1. Check the course repository issues: [GitHub Issues](#)
2. Review official documentation for each tool
3. Ask in course discussion forums
4. Contact the instructor for technical support

## Next Steps

Once your environment is set up:

1. Review the course schedule and materials
2. Familiarize yourself with the lab structure



3. Complete the setup test successfully

4. Be ready for Day 1 of the course!

**Course Repository:** <https://github.com/chrisminnick/intermediate-node-dev>

**Support Email:** [support@watzthis.com](mailto:support@watzthis.com)