

# Virtual Environment & Dependencies Setup Guide

## Step 1: Create Virtual Environment

### For Windows:

```
bash
# Create virtual environment
python -m venv venv

# Activate virtual environment
venv\Scripts\activate

# Verify activation (you should see (venv) in your prompt)
where python
```

### For Mac/Linux:

```
bash
# Create virtual environment
python3 -m venv venv

# Activate virtual environment
source venv/bin/activate

# Verify activation (you should see (venv) in your prompt)
which python
```

### Verification Commands:

```
bash
# Check Python version
python --version

# Check pip version
pip --version

# List currently installed packages (should be minimal)
```

```
pip list
```

## Step 2: Upgrade pip (Recommended)

```
bash
```

```
# Upgrade pip to latest version
```

```
python -m pip install --upgrade pip
```

```
# Verify upgrade
```

```
pip --version
```

## Step 3: Install Core Dependencies

**Install the packages from your list:**

```
bash
```

```
# Install all dependencies at once
```

```
pip install fastapi uvicorn pydantic sqlalchemy pytest requests passlib[bcrypt]
```

### Note about sqlite3:

**sqlite3 is built into Python**, so you don't need to install it separately. The command above will work without the sqlite3 package.

### Alternative - Install one by one (if you encounter issues):

```
bash
```

```
pip install fastapi
```

```
pip install uvicorn[standard] # [standard] includes extra dependencies
```

```
pip install pydantic
```

```
pip install sqlalchemy
```

```
pip install pytest
```

```
pip install requests
```

```
pip install "passlib[bcrypt]" # Quotes needed for bcrypt extra
```

## Step 4: Install Additional Recommended Dependencies

For a complete development environment, consider adding these:

```
bash
```

```
# Additional useful packages for ScoutConnect
```

```
pip install python-jose[cryptography] # For JWT tokens
```

```
pip install python-multipart      # For form data
pip install python-dotenv        # For environment variables
pip install alembic              # For database migrations
pip install httpx                # Better async HTTP client
pip install pytest-asyncio       # For async testing
pip install black                # Code formatting
pip install flake8               # Code linting
```

## Step 5: Verify Installation

Create a test script to verify everything works:

```
bash
# Create test file
touch test_imports.py
```

```
python
# test_imports.py
"""
```

Test script to verify all dependencies are installed correctly.

```
"""
```

```
def test_imports():
    """Test all critical imports."""

    print("🔧 Testing imports...")

    try:
        import fastapi
        print(f"✅ FastAPI {fastapi.__version__}")
    except ImportError as e:
        print(f"❌ FastAPI: {e}")

    try:
        import uvicorn
        print(f"✅ Uvicorn {uvicorn.__version__}")
    except ImportError as e:
        print(f"❌ Uvicorn: {e}")

    try:
        import pydantic
        print(f"✅ Pydantic {pydantic.__version__}")
    except ImportError as e:
```

```

print(f"❌ Pydantic: {e}")

try:
    import sqlalchemy
    print(f"✅ SQLAlchemy {sqlalchemy.__version__}")
except ImportError as e:
    print(f"❌ SQLAlchemy: {e}")

try:
    import sqlite3
    print(f"✅ SQLite3 {sqlite3.sqlite_version}")
except ImportError as e:
    print(f"❌ SQLite3: {e}")

try:
    import pytest
    print(f"✅ Pytest {pytest.__version__}")
except ImportError as e:
    print(f"❌ Pytest: {e}")

try:
    import requests
    print(f"✅ Requests {requests.__version__}")
except ImportError as e:
    print(f"❌ Requests: {e}")

try:
    import passlib
    print(f"✅ Passlib {passlib.__version__}")
except ImportError as e:
    print(f"❌ Passlib: {e}")

# Test optional packages
try:
    import jose
    print(f"✅ Python-JOSE (JWT support)")
except ImportError:
    print(f"⚠️ Python-JOSE not installed (recommended for JWT)")

try:
    import dotenv
    print(f"✅ Python-dotenv")
except ImportError:
    print(f"⚠️ Python-dotenv not installed (recommended for .env files)")

```

```
print("\n🎉 Import test complete!")

if __name__ == "__main__":
    test_imports()
```

Run the test:

```
bash
```

```
python test_imports.py
```

## Step 6: Create Requirements Files

### Basic requirements.txt (as requested):

```
bash
```

```
pip freeze > requirements.txt
```

### Enhanced Requirements Structure (Recommended):

Create organized requirements files:

```
bash
```

```
# Create requirements directory
```

```
mkdir requirements
```

```
# Create base requirements (core dependencies)
```

```
pip freeze | grep -E "(fastapi|uvicorn|pydantic|sqlalchemy|pytest|requests|passlib)" > requirements/base.txt
```

```
# Create development requirements
```

```
cat > requirements/dev.txt << 'EOF'
```

```
-r base.txt
```

```
pytest-asyncio==0.21.1
```

```
pytest-cov==4.1.0
```

```
httpx==0.25.2
```

```
black==23.11.0
```

```
flake8==6.1.0
```

```
python-dotenv==1.0.0
```

```
python-jose[cryptography]==3.3.0
```

```
python-multipart==0.0.6
```

```
alembic==1.12.1
```

EOF

```
# Create production requirements
cat > requirements/prod.txt << 'EOF'
-r base.txt
gunicorn==21.2.0
EOF
```

```
# Create full requirements.txt in root (combines everything)
pip freeze > requirements.txt
```

## Step 7: Test FastAPI Installation

Create a simple test API to verify everything works:

```
python
# test_api.py
from fastapi import FastAPI
import uvicorn

app = FastAPI(title="ScoutConnect Test API")

@app.get("/")
async def root():
    return {
        "message": "ScoutConnect API is working!",
        "status": "success",
        "environment": "development"
    }

@app.get("/health")
async def health_check():
    return {"status": "healthy"}

if __name__ == "__main__":
    print("🚀 Starting test server...")
    print("📍 Visit: http://localhost:8000")
    print("📖 Docs: http://localhost:8000/docs")
    uvicorn.run(app, host="127.0.0.1", port=8000, reload=True)
```

Run the test API:

bash

```
python test_api.py
```

Visit <http://localhost:8000> to see if it works!

## Step 8: Update Your Project Structure

Update your existing project files with the dependencies:

```
python
# app/main.py (update your existing file)
from fastapi import FastAPI, Depends
from fastapi.middleware.cors import CORSMiddleware
import uvicorn
import os
from dotenv import load_dotenv

# Load environment variables
load_dotenv()

app = FastAPI(
    title="ScoutConnect API",
    description="Smart Scouting Platform for Talent Discovery & Collaboration",
    version="0.1.0"
)

# CORS middleware
app.add_middleware(
    CORSMiddleware,
    allow_origins=["http://localhost:3000"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

@app.get("/")
async def root():
    return {
        "message": "Welcome to ScoutConnect API",
        "version": "0.1.0",
        "status": "active"
    }

@app.get("/health")
async def health_check():
```

```

    return {"status": "healthy", "database": "connected"}

if __name__ == "__main__":
    uvicorn.run(
        "main:app",
        host="0.0.0.0",
        port=8000,
        reload=True
    )

```

## Troubleshooting Common Issues

### Issue 1: Virtual Environment Not Activating

#### Windows:

```

bash
# If activation fails, try:
venv\Scripts\activate.bat
# or
venv\Scripts\Activate.ps1 # For PowerShell

```

#### Mac/Linux:

```

bash
# If activation fails, check the path:
ls venv/bin/
source venv/bin/activate

```

### Issue 2: pip Install Failures

```

bash
# If you get SSL errors:
pip install --trusted-host pypi.org --trusted-host pypi.python.org --trusted-host
files.pythonhosted.org <package_name>

# If you get permissions errors:
pip install --user <package_name>

# Update certificates (Mac):
/Applications/Python\ 3.x/Install\ Certificates.command

```



### Issue 3: Package Conflicts

bash

*# Clear pip cache*

pip cache purge

*# Reinstall packages*

pip uninstall <package\_name>

pip install <package\_name>

### Issue 4: sqlite3 Import Errors

sqlite3 is built into Python, but if you have issues:

bash

*# Check Python installation*

python -c "import sqlite3; print(sqlite3.sqlite\_version)"

*# If it fails, you may need to reinstall Python with sqlite support*

## Quick Command Reference

bash

*# Activate virtual environment*

*# Windows: venv\Scripts\activate*

*# Mac/Linux: source venv/bin/activate*

*# Deactivate virtual environment*

deactivate

*# Install package*

pip install <package\_name>

*# Uninstall package*

pip uninstall <package\_name>

*# List installed packages*

pip list

*# Show package info*

pip show <package\_name>

*# Install from requirements*

```
pip install -r requirements.txt
```

```
# Update requirements
```

```
pip freeze > requirements.txt
```

## Verification Checklist

- Virtual environment created (`venv/` directory exists)
- Virtual environment activated (prompt shows `(venv)`)
- pip upgraded to latest version
- All core dependencies installed successfully
- `requirements.txt` file created with `pip freeze`
- Import test script runs without errors
- Test FastAPI server starts and responds
- Dependencies verified with version numbers

## Next Steps

1. **Test your setup** by running the test API
2. **Commit your requirements.txt** to Git
3. **Start developing** your ScoutConnect API
4. **Add more dependencies** as needed for specific features

Your development environment is now ready for building ScoutConnect! 🎉