Quick Start Guide

TL;DR - Get Running in 5 Minutes

1. Build and Run

```
# Navigate to backend
cd backend-spring

# Build
./gradlew build

# Run
./gradlew bootRun
```

Server starts at: `http://localhost:8080`

2. Test API

```
# Get all folders
curl http://localhost:8080/api/folders

# Get app state
curl http://localhost:8080/api/app-state

# Create a folder
curl -X POST http://localhost:8080/api/folders \
  -H "Content-Type: application/json" \
  -d '{
    "name": "My Folder",
    "ordering": 0,
    "createdAt": 1698765432000.0,
    "createdAtHk": "2024-10-30 15:30:32"
  }'
```

3. Make Schema Changes

Step 1: Edit entity (e.g., add a field)

Step 2: Create migration file:

```
# Create: src/main/resources/db/migration/V2__Your_change.sql
echo "ALTER TABLE shell_script ADD COLUMN description TEXT;" > \
  src/main/resources/db/migration/V2__Add_description.sql
```

Step 3: Restart app (migration auto-applies)

```
./gradlew bootRun
```

4. Build Production JAR

```
./gradlew bootJar

# Output: build/libs/script-manager-backend-0.0.1-SNAPSHOT.jar
```

Project Structure

```
backend-spring/
├── src/main/kotlin/com/scriptmanager/
│   ├── entity/          # Database models
│   ├── repository/      # Data access
│   ├── controller/      # REST endpoints
│   └── Application.kt   # Main app
├── src/main/resources/
│   ├── application.yml  # Configuration
│   └── db/migration/    # Flyway SQL files
└── build.gradle.kts     # Dependencies
```

Key Configuration

Database: `application.yml`

```
spring:
  datasource:
    url: jdbc:sqlite:../src-tauri/database.db
  jpa:
    hibernate:
      ddl-auto: validate  # Never auto-update schema!
```

Common Commands

```
# Run app
./gradlew bootRun

# Run tests
./gradlew test

# Build JAR
```

```
./gradlew bootJar

# Clean build
./gradlew clean build

# Check dependencies
./gradlew dependencies
```

REST Endpoints

Method	Endpoint	Description
GET	/api/folders	Get all folders
GET	/api/folders/{id}	Get folder by ID
POST	/api/folders	Create folder
PUT	/api/folders/{id}	Update folder
DELETE	/api/folders/{id}	Delete folder
GET	/api/scripts	Get all scripts
GET	/api/scripts/{id}	Get script by ID
POST	/api/scripts	Create script
PUT	/api/scripts/{id}	Update script
DELETE	/api/scripts/{id}	Delete script
GET	/api/app-state	Get app state
PUT	/api/app-state	Update app state

Need More Details?

See 3_COMPLETE_SETUP_GUIDE.md for complete documentation.

See FLYWAY_WORKFLOW.md for detailed migration guide.