ASMC API - Quick Start Guide

Get your ASMC API up and running in under 10 minutes! This guide covers the essential steps to set up the backend API for local development.

Prerequisites

Before you begin, ensure you have:

- **Node.js 18+** installed
- **MongoDB 4.4+** installed and running
- **Git** installed
- **Code editor** (VS Code recommended)

Quick Setup (5 Minutes)

Step 1: Clone and Install

```
# Clone the repository
git clone <repository-url>
cd asmc-api

# Install dependencies
npm install
```

Step 2: Environment Setup

```
# Copy environment template
cp .env.example .env.development

# Edit environment variables
nano .env.development
```

Minimum required configuration:

```
PORT=7055
NODE_ENV=development
MONGO_URI=mongodb://localhost:27017/asmc
JWT_SECRET=your-super-secret-jwt-key-minimum-32-characters
```

Step 3: Start the Server

```
# Start development server
npm run dev
```

Step 4: Verify Installation

Open your browser and visit:

- **API Health Check**: http://localhost:7055/health
- **API Documentation**: http://localhost:7055/api-docs

You should see:

- Health check returns "ok"
- Swagger UI documentation interface

First API Call

Test the API with a simple authentication call:

```bash
# Test admin login (use default credentials or create admin user)
curl -X POST http://localhost:7055/auth/admin-login \
  -H "Content-Type: application/json" \
  -d '{
    "username": "admin@asmc.com",
    "password": "password123"
  }'
```

Expected response:

```json
{
    "success": true,
    "message": "Login successful",
    "result": {
        "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
        "user": {
            "id": "user_id",
            "username": "admin@asmc.com",
            "role": "admin"
        }
    }
}
```

Development Workflow

Available Scripts

```bash
# Development
npm run dev                 # Start with nodemon (auto-restart)

# Production
npm start                   # Start production server
npm run start:staging       # Start staging environment
npm run start:prod          # Start with PM2

# Database
npm run backup              # Backup database
npm run restore             # Restore database

# Documentation
npm run generate-docs       # Generate API documentation
```

Project Structure Overview

```
asmc-api/
├── app/
│   ├── controller/         # API endpoints logic
│   ├── models/            # Database models
│   ├── middlewares/       # Authentication, validation
│   ├── helpers/           # Utility functions
│   └── routes/            # Route definitions
├── docs/                  # Swagger documentation
├── cron/                  # Scheduled jobs
└── public/                # Static files
```

▯ Database Setup

MongoDB Installation

Ubuntu/Debian:

```
wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
sudo apt-get update
sudo apt-get install -y mongodb-org
sudo systemctl start mongod
sudo systemctl enable mongod
```

macOS:

```
brew tap mongodb/brew
brew install mongodb-community
brew services start mongodb/brew/mongodb-community
```

Windows:

Download and install from [MongoDB Download Center]

Verify MongoDB

```
# Check if MongoDB is running
mongo --eval "db.adminCommand('ismaster')"

# Should return: { "ismaster" : true, ... }
```

▯ Environment Configuration

Complete Environment Variables

```
# Server Configuration
PORT=7055
NODE_ENV=development

# Database
```

```
MONGO_URI=mongodb://localhost:27017/asmc

# Authentication
JWT_SECRET=your-super-secret-jwt-key-minimum-32-characters
JWT_EXPIRE=7d

# Email (Optional for development)
SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_USER=your-email@gmail.com
SMTP_PASS=your-app-password

# Image Processing (Optional for development)
IMAGEKIT_PUBLIC_KEY=your-imagekit-public-key
IMAGEKIT_PRIVATE_KEY=your-imagekit-private-key
IMAGEKIT_URL_ENDPOINT=https://ik.imagekit.io/your-imagekit-id

# Payment Gateway (Optional for development)
CCAVENUE_MERCHANT_ID=your-merchant-id
CCAVENUE_ACCESS_CODE=your-access-code
CCAVENUE_WORKING_KEY=your-working-key
```

API Testing

Using Swagger UI

1. Go to http://localhost:7055/api-docs
2. Click "Authorize" button
3. Enter your JWT token: `Bearer your-token-here`
4. Test any endpoint

Using cURL

```
# Get all members (requires authentication)
curl -X GET http://localhost:7055/members \
  -H "Authorization: Bearer YOUR_JWT_TOKEN"

# Create a new member
curl -X POST http://localhost:7055/members \
  -H "Authorization: Bearer YOUR_JWT_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "firstName": "John",
    "lastName": "Doe",
    "email": "john.doe@example.com",
    "phone": "9876543210",
    "address": "123 Main St"
  }'
```

Using Postman

1. Import the API collection (if available)
2. Set base URL: `http://localhost:7055`

3. Add authorization header: `Authorization: Bearer YOUR_JWT_TOKEN`

Authentication Flow

1. Admin Login

```
curl -X POST http://localhost:7055/auth/admin-login \
  -H "Content-Type: application/json" \
  -d '{
    "username": "admin@asmc.com",
    "password": "password123"
  }'
```

2. Use Token in Subsequent Requests

```
# Extract token from login response
TOKEN="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

# Use token in API calls
curl -X GET http://localhost:7055/members \
  -H "Authorization: Bearer $TOKEN"
```

Common Issues & Solutions

Issue 1: Port Already in Use

```
# Error: EADDRINUSE: address already in use :::7055
# Solution:
sudo lsof -i :7055
sudo kill -9 <PID>
# Or use different port:
PORT=7056 npm run dev
```

Issue 2: MongoDB Connection Failed

```
# Error: MongoNetworkError: failed to connect to server
# Solution:
sudo systemctl start mongod
# Or check MongoDB status:
sudo systemctl status mongod
```

Issue 3: JWT Token Invalid

```
# Error: JsonWebTokenError: invalid token
# Solution: Check JWT_SECRET in .env file
echo $JWT_SECRET
# Should be minimum 32 characters
```

Issue 4: Module Not Found

```
# Error: Cannot find module
# Solution: Reinstall dependencies
rm -rf node_modules package-lock.json
npm install
```

Next Steps

After successful setup:

1. **Explore API Documentation**: Visit http://localhost:7055/api-docs
2. **Test Core Endpoints**: Try authentication, member creation, and data retrieval
3. **Set up Frontend**: Connect your admin panel or mobile app
4. **Configure Production**: Review deployment guides for production setup

Additional Resources

- **Full Documentation**: README.md
- **Architecture Guide**: ARCHITECTURE_OVERVIEW.md
- **API Reference**: API_REFERENCE.md
- **Deployment Guide**: DEPLOYMENT_GUIDE.md

Getting Help

If you encounter issues:

1. **Check Logs**: Look at console output for error messages
2. **Verify Environment**: Ensure all required environment variables are set
3. **Test Dependencies**: Verify Node.js, MongoDB, and npm versions
4. **Review Documentation**: Check the full documentation for detailed guides

Verification Checklist

- □ Node.js 18+ installed
- □ MongoDB running and accessible
- □ Dependencies installed (`npm install`)
- □ Environment variables configured
- □ Server starts without errors (`npm run dev`)
- □ Health check returns "ok" (http://localhost:7055/health)
- □ API documentation loads (http://localhost:7055/api-docs)
- □ Can authenticate and get JWT token
- □ Can make authenticated API calls

Congratulations! Your ASMC API is now running locally and ready for development!

Next: Set up the admin panel or mobile app to connect to your API.