AgroTrack-Lite v2.0 - Complete Installation Guide

📁 Project Structure

```
agrotrack-lite-v2/
├── src/
│   ├── agents/
│   │   ├── base.ts          # Base agent class with execution modes
│   │   ├── intent.ts        # NLP parser (LangChain + OpenAI)
│   │   ├── risk.ts          # Risk assessment via Mirror Node
│   │   ├── market.ts        # Price discovery
│   │   ├── escrow.ts        # HTS token lock (RETURN_BYTE)
│   │   └── settlement.ts    # Payment release (RETURN_BYTE)
│   ├── hedera/
│   │   ├── client.ts        # Hedera client singleton
│   │   ├── hcs.ts           # Consensus Service operations
│   │   ├── hts.ts           # Token Service operations
│   │   └── mirror.ts        # Mirror Node queries
│   ├── orchestrator/
│   │   └── workflow.ts      # Multi-agent coordination
│   ├── api/
│   │   └── server.ts        # Express API + routes
│   ├── sms/
│   │   └── gateway.ts       # SMS send/receive
│   ├── types/
│   │   ├── agents.ts        # Agent interfaces
│   │   └── intents.ts       # Zod schemas
│   └── index.ts             # Main entry point
├── scripts/
│   ├── setup-hedera.ts      # Initialize HCS + HTS
│   ├── demo-journey.ts      # Automated testing
│   └── test-agents.ts       # Unit tests
├── dashboard/
│   └── index.html           # React dashboard (single-file)
├── docker/
│   ├── Dockerfile
│   └── docker-compose.yml
├── .env.example
├── .gitignore
├── package.json
├── tsconfig.json
└── README.md
```

🛠️ Step-by-Step Setup

Step 1: Prerequisites

Install Node.js 20+

bash

```bash
# macOS (via Homebrew)
brew install node

# Ubuntu
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt-get install -y nodejs

# Windows
# Download from https://nodejs.org
```

Get Hedera Testnet Account

1. Go to portal.hedera.com
2. Create account
3. Fund with testnet HBAR (free)
4. Note your Account ID and Private Key

Get OpenAI API Key

1. Go to platform.openai.com
2. Create API key
3. Add $5 credits (for testing)

Step 2: Create Project

bash

```
# Create project directory
mkdir agrotrack-lite-v2
cd agrotrack-lite-v2

# Initialize npm
npm init -y

# Install dependencies
npm install @hashgraph/sdk @langchain/openai @langchain/core \
  express cors axios zod dotenv

# Install dev dependencies
npm install --save-dev typescript @types/node @types/express \
  @types/cors tsx
```

Step 3: Configure TypeScript

Create `tsconfig.json`:

```json
{
  "compilerOptions": {
    "target": "ES2022",
    "module": "ESNext",
    "moduleResolution": "node",
    "lib": ["ES2022"],
    "outDir": "./dist",
    "rootDir": "./src",
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true,
    "resolveJsonModule": true
  },
  "include": ["src/**/*"],
  "exclude": ["node_modules", "dist"]
}
```

Step 4: Create Directory Structure

```bash
mkdir -p src/{agents,hedera,orchestrator,api,sms,types}
mkdir -p scripts dashboard docker
```

Step 5: Configure Environment

Create `.env`:

```bash
# Hedera Configuration
HEDERA_NETWORK=testnet
HEDERA_ACCOUNT_ID=0.0.YOUR_ACCOUNT_ID
HEDERA_PRIVATE_KEY=YOUR_PRIVATE_KEY_HERE

# Leave empty - will be created by setup script
HCS_TOPIC_ID=
ESCROW_TOKEN_ID=

# Custodial Accounts (can all be the same for testing)
ESCROW_ACCOUNT_ID=0.0.YOUR_ACCOUNT_ID
BUYER_ACCOUNT_ID=0.0.YOUR_ACCOUNT_ID
FARMER_ACCOUNT_ID=0.0.YOUR_ACCOUNT_ID

# AI Configuration
OPENAI_API_KEY=sk-proj-YOUR_KEY_HERE

# SMS Gateway (start with stub)
SMS_MODE=stub
AT_USERNAME=sandbox
AT_API_KEY=
AT_SENDER=AgroTrack

# Server
PORT=3000
MIRROR_NODE_URL=https://testnet.mirrornode.hedera.com/api/v1
```

Create `.env.example` (for version control):

```bash
```

```
cp .env .env.example
# Edit .env.example to remove sensitive values
```

Step 6: Copy Source Files

Copy all the TypeScript files I provided earlier into their respective directories:

1. **Base Agent & Types** → `src/agents/base.ts`, `src/types/`
2. **Intent Agent** → `src/agents/intent.ts`
3. **Risk Agent** → `src/agents/risk.ts`
4. **Market Agent** → `src/agents/market.ts`
5. **Escrow Agent** → `src/agents/escrow.ts`
6. **Settlement Agent** → `src/agents/settlement.ts`
7. **Hedera Integration** → `src/hedera/`
8. **Orchestrator** → `src/orchestrator/workflow.ts`
9. **API Server** → `src/api/server.ts`
10. **SMS Gateway** → `src/sms/gateway.ts`
11. **Main Entry** → `src/index.ts`

Step 7: Update package.json

Add these scripts:

```
json
```

```
{
  "type": "module",
  "scripts": {
    "dev": "tsx watch src/index.ts",
    "build": "tsc",
    "start": "node dist/index.js",
    "setup": "tsx scripts/setup-hedera.ts",
    "demo": "tsx scripts/demo-journey.ts",
    "test": "tsx scripts/test-agents.ts"
  }
}
```

Step 8: Run Setup

```bash
# This will:
# 1. Create HCS topic
# 2. Create escrow token
# 3. Associate token with accounts
npm run setup
```

Expected Output:

```
🔧 AgroTrack-Lite v2.0 Setup

✅ Connected to testnet
   Operator: 0.0.xxxxx

📋 Setting up HCS topic...
✅ Topic ready: 0.0.xxxxx

🪙 Creating escrow token...
✅ Token created: 0.0.xxxxx

🔗 Associating token with accounts...
   ✅ Associated with 0.0.xxxxx
   ✅ Associated with 0.0.xxxxx

✅ Setup complete!

Update your .env with:
HCS_TOPIC_ID=0.0.xxxxx
ESCROW_TOKEN_ID=0.0.xxxxx
```

Update `.env` with the printed values.

Step 9: Start Development Server

```bash
npm run dev
```

Expected Output:

```
🚀 AgroTrack-Lite v2.0
   Server: http://localhost:3000
   Webhook: http://localhost:3000/webhook/sms
   Health: http://localhost:3000/health

✅ Ready for SMS messages
```

Step 10: Test the System

Terminal 1 - Keep server running:

```bash
npm run dev
```

Terminal 2 - Send test SMS:

```bash
# Create offer
curl -X POST http://localhost:3000/webhook/sms \
  -d "from=+254700000001" \
  -d "text=Maize 200kg Kisumu"

# Wait 5 seconds, check console for OTP

# Accept offer (replace OTP from console)
curl -X POST http://localhost:3000/webhook/sms \
  -d "from=+254700000001" \
  -d "text=YES 123456"

# Deliver
curl -X POST http://localhost:3000/webhook/sms \
  -d "from=+254700000001" \
  -d "text=Delivered 198kg Grade B OTP 123456"
```

You should see:

- ✅ Parallel agent execution (Risk + Market)
- ✅ Agent decision logs
- ✅ Escrow lock transaction
- ✅ Settlement release transaction
- ✅ SMS responses (in console if mode=stub)

Step 11: View in Dashboard

1. Open `dashboard/index.html` in browser

2. See real-time event stream
3. View agent status
4. Monitor statistics

Or run as dev server:

```bash
# Install http-server globally
npm install -g http-server

# Serve dashboard
cd dashboard
http-server -p 5173
```

Open `http://localhost:5173`

Step 12: Verify on Hedera

1. Go to hashscan.io/testnet
2. Search for your topic ID (from .env)
3. See all logged messages
4. Search for token ID
5. View token transfers

🧪 Running Automated Demo

```bash
npm run demo
```

This simulates a complete farmer journey with proper timing and logging.

🐳 Docker Deployment

```bash
```

```
# Build
docker-compose build

# Run
docker-compose up -d

# View logs
docker-compose logs -f api

# Stop
docker-compose down
```

🔧 Troubleshooting

Issue: "Topic not found"

Solution: Run `npm run setup` to create topic

Issue: "Token association failed"

Solution: Ensure all account IDs in .env are valid and funded

Issue: "OpenAI API error"

Solution: Check OPENAI_API_KEY and account balance

Issue: "Mirror Node timeout"

Solution: Network issue - retry after few seconds

Issue: Import errors in TypeScript

Solution: Ensure `"type": "module"` in package.json

📝 Common Tasks

Reset Everything

```
bash
```

```
# Remove node_modules
rm -rf node_modules package-lock.json

# Remove build
rm -rf dist

# Reinstall
npm install

# Re-run setup
npm run setup
```

Switch to Live SMS

1. Sign up at africastalking.com
2. Get API key
3. Update .env:

```
SMS_MODE=live
AT_USERNAME=your_username
AT_API_KEY=your_key
```

4. Set up ngrok:

```bash
ngrok http 3000
```

5. Copy ngrok URL to Africa's Talking webhook

Deploy to Production

1. Get production Hedera account
2. Update .env with mainnet credentials
3. Change HEDERA_NETWORK=mainnet
4. Deploy to Railway/Render/AWS
5. Run setup once on production

🎯 Next Steps

1. ✅ Complete installation
2. ✅ Run test journey
3. ✅ View events in dashboard
4. ✅ Verify on Hashscan
5. 🎨 Customize for your use case
6. 🚀 Deploy to production
7. 📱 Connect real SMS gateway
8. 💰 Integrate M-Pesa for cash-out

🆘 Support

- **GitHub Issues:** Create an issue
- **Hedera Discord:** Join community
- **Documentation:** docs.hedera.com

Installation complete! 🎉

You now have a fully functional multi-agent system running on Hedera with SMS integration, token escrow, and real-time monitoring.