

MVP-Leadership Intelligence System - Complete Setup Guide

Overview

This guide will walk you through setting up the MVP-Leadership Intelligence System from scratch. Follow these steps carefully to ensure proper configuration.

Prerequisites Checklist

Before starting, ensure you have:

- ☐ Python 3.8 or higher installed
- ☐ OpenAI account with API access
- ☐ Limitless app installed and recording conversations
- ☐ Google Workspace account (Gmail, Calendar, Sheets)
- ☐ Administrative access to install Python packages

Step 1: API Keys and Credentials

1.1 OpenAI API Key

1. Create OpenAI Account

- Go to <https://platform.openai.com/>
- Sign up or log in to your account

2. Generate API Key

- Navigate to <https://platform.openai.com/api-keys>
- Click "Create new secret key"
- Name it "Leadership Intelligence System"
- Copy the key (starts with `sk-`)
- **Important:** Store this securely - you won't see it again

3. Set Up Billing

- Go to <https://platform.openai.com/account/billing>
- Add payment method
- Recommended: Set usage limit to \$50/month for safety

1.2 Limitless API Key

1. Open Limitless App

- Ensure you have conversations recorded
- Go to Settings or Preferences

2. Find API Section

- Look for "API", "Developer", or "Integration" settings
- Generate or copy your API key
- Test that lifelogs are being captured properly

1.3 Google Cloud Setup

1. Create Google Cloud Project

- Go to <https://console.cloud.google.com/>
- Click "New Project"
- Name: "Leadership Intelligence System"
- Note your Project ID

2. Enable Required APIs

Required APIs to enable:

- Gmail API
- Google Calendar API
- Google Sheets API
- Google Drive API

For each API:

- Go to "APIs & Services" → "Library"
- Search for the API name
- Click "Enable"

3. Create OAuth2 Credentials

- Go to "APIs & Services" → "Credentials"
- Click "Create Credentials" → "OAuth client ID"
- Application type: "Desktop application"
- Name: "Leadership Intelligence"
- Download the JSON file
- Rename it to `credentials.json`
- Save it in your MVP-Leadership Intelligence directory

Step 2: Environment Setup

2.1 Install Python Dependencies

```
bash

# Navigate to your directory
cd "G:\My Drive\Jarrett Folders\JD INC\Limitless3ss\MVP-Leadership Intelligence"

# Install required packages
pip install -r requirements.txt
```

2.2 Create Environment File

1. Copy the template

```
bash

copy .env.example .env
```

2. Edit `.env` file with your actual keys

```
env

# OpenAI API Key
OPENAI_API_KEY=sk-your-actual-openai-key-here

# Limitless API Key
LIMITLESS_API_KEY=your-actual-limitless-key-here
```

Step 3: Google Sheets Task Management Setup

3.1 Create Your Task Management Sheet

1. Create New Google Sheet

- Go to <https://sheets.google.com>
- Create new sheet
- Name it "Leadership Task Management"

2. Set Up Columns (exact headers required)

Column A: Task
Column B: Priority
Column C: Owner
Column D: Area
Column E: Status
Column F: Deliverable
Column G: Notes
Column H: Date
Column I: Due
Column J: Email Reminder
Column K: Frequency
Column L: Last Email Sent

3. Create Sheet Tabs

- Rename "Sheet1" to "Todo Tasks"
- Create another tab called "Completed Tasks"

4. Get Your Sheet ID

- Look at the URL: `https://docs.google.com/spreadsheets/d/SHEET_ID_HERE/edit`
- Copy the long string between `/d/` and `/edit`

3.2 Set Up Calendar Access

1. Find Your Calendar IDs

- Go to <https://calendar.google.com>
- Click settings gear → "Settings"
- In left sidebar, click on each calendar you want to include
- Scroll to "Calendar ID" section
- Copy the calendar ID (usually ends with @gmail.com or @group.calendar.google.com)

2. Primary Calendar

- Your main calendar ID is usually 'primary'
- For shared calendars, use the full email-like ID

Step 4: Script Configuration

4.1 Update the Config Class

Edit `leadshipintelligence script v7.0.py` and find the `Config` class:

python

```
class Config:
    def __init__(self):
        # UPDATE: Path to your Google credentials file
        self.credentials_path = r"G:\My Drive\Jarrett Folders\JD INC\Limitle3ss\MVP-Leadership

        # UPDATE: Where you want output files saved
        self.output_dir = r"G:\My Drive\Jarrett Folders\JD INC\Limitle3ss\MVP-Leadership Intell

        # UPDATE: Your Google Sheets spreadsheet ID
        self.spreadsheet_id = "your-actual-sheet-id-from-step-3"

        # UPDATE: Your Google Calendar IDs
        self.calendar_ids = [
            'primary', # Your main calendar
            'your-shared-calendar-id@group.calendar.google.com', # If you have shared calendar
            # Add more calendar IDs as needed
        ]
```

4.2 Create Output Directory

bash

```
mkdir output
```

Step 5: Email Label Setup (Optional)

5.1 Create Priority Labels in Gmail

1. Open Gmail

- Go to <https://gmail.com>

2. Create Labels

- Click settings gear → "See all settings"
- Go to "Labels" tab
- Create new labels:
 - ! - [YourName] (for high priority emails)
 - Any other priority labels you want

3. Update Script for Your Labels

```
python
```

```
# In EmailManager.fetch_emails() method, update this line:  
"priority": "label:\"! - YourActualName\" is:unread"
```

Step 6: First Run and Testing

6.1 Initial Test Run

```
bash
```

```
python "leadshipintelligence script v7.0.py"
```

6.2 OAuth Authorization

1. Browser Will Open

- Google will ask for permissions
- Sign in with your Google account
- Grant all requested permissions:
 - Gmail access
 - Calendar access
 - Sheets access
 - Drive access

2. Complete Authorization

- Click "Allow" for each permission
- Browser will show "Authorization successful"
- Return to your command prompt

6.3 Review First Output

The system will:

1. Fetch conversations from past 3 days
2. Analyze emails and calendar
3. Generate two output files
4. Ask if you want to add new tasks to your sheet

Expected Output Files:

- `output/YYYY-MM-DD_HHMM_detailed_lifelog_analysis.txt`
- `output/leadership_summary_YYYYMMDD_HHMM.txt`

Step 7: Verification and Troubleshooting

7.1 Verify Data Sources

Check Conversation Data:

Look for: "✅ Found X conversations with content"

If you see: "❌ NO CONVERSATION DATA" - check Limitless API key

Check Email Data:

Look for: "📧 Found X priority emails"

If you see errors - check Gmail API permissions

Check Calendar Data:

Look for: "📅 Found X valid events"

If you see errors - check Calendar API and calendar IDs

Check Task Data:

Look for: "📋 Loaded X active tasks"

If you see errors - check Sheets API and spreadsheet ID

7.2 Common Issues and Solutions

Issue: "No lifelog data found"

Solutions:

1. Check `LIMITLESS_API_KEY` in `.env` file
2. Verify Limitless app is recording conversations
3. Ensure you have conversations in past 3 days
4. Check that `_client.py` module exists and works

Issue: "Google credentials error"

Solutions:

1. Re-download credentials.json from Google Cloud
2. Verify file path in Config class
3. Delete any existing token_*.pickle files and re-authorize
4. Check that all required APIs are enabled

Issue: "OpenAI rate limit exceeded"

Solutions:

1. System automatically handles chunking for large datasets
2. Check your OpenAI usage at platform.openai.com
3. Consider upgrading OpenAI plan for higher limits
4. Reduce conversation window if needed

Issue: "No tasks found in sheet"

Solutions:

1. Verify spreadsheet ID in Config class
2. Check sheet name is exactly "Todo Tasks"
3. Ensure column headers match required format
4. Verify Sheets API permissions

Step 8: Customization Options

8.1 Adjust Analysis Window

python

```
# In LifeLogProcessor.fetch_LifeLogs(), change:
if start_date is None:
    start_date = (datetime.now() - timedelta(days=X)).date().isoformat() # Change X
```

8.2 Modify Task Priorities

python

```
# In TaskManager._parse_task_data(), change:
if priority in ['P1', 'P2', 'P3']: # Add P3 if you want to see P3 tasks
```

8.3 Customize AI Prompts

Edit the `PromptTemplates` class to modify:

- What constitutes a "commitment"
- How to categorize time vs action phrases
- What leadership insights to focus on

Step 9: Automation (Optional)

9.1 Windows Task Scheduler

1. Create Batch File

```
batch

@echo off

cd "G:\My Drive\Jarrett Folders\JD INC\Limitle3ss\MVP-Leadership Intelligence"

python "leadshipintelligence script v7.0.py"

pause
```

2. Schedule Daily Run

- Open Task Scheduler
- Create Basic Task
- Set to run daily at 7:00 AM
- Point to your batch file

9.2 Monitor and Maintain

- **Check output files daily** for quality
- **Review and approve new tasks** before adding to sheet
- **Monitor API usage** to avoid unexpected charges
- **Update conversation analysis window** based on your meeting patterns

Support

If you encounter issues:

1. Check the troubleshooting section above
2. Review log files for detailed error messages
3. Verify all API keys and permissions
4. Test each component individually

The system is designed to be robust and handle most edge cases automatically. With proper setup, it should provide reliable daily intelligence for your leadership workflow.

Security Best Practices

1. Protect API Keys

- Never commit .env file to version control
- Use environment variables in production
- Rotate keys periodically

2. Limit Permissions

- Google OAuth uses minimal required scopes
- Review and revoke unused permissions regularly
- Monitor API usage for unusual activity

3. Data Handling

- Output files contain sensitive conversation data
- Store in secure, access-controlled directories
- Consider encryption for highly sensitive information