# Zip File Setup Guide: Data Integration Hub

This guide provides a comprehensive, step-by-step process for setting up and running the Data Integration Hub from the provided zip file.

### 🎯 Quick Start (5 Minutes)

1. **Extract the Zip File**: Extract the contents of the zip file to a folder on your computer, for example, C:\Users\YourName\Desktop\integrations\_technical\_assessment.
2. **Open Command Prompt**: Navigate to the extracted folder in your command prompt.  
   Bash  
   cd C:\Users\YourName\Desktop\integrations\_technical\_assessment

### 📋 Prerequisites Check

Make sure you have the following software installed on your system. You can check the versions by running the commands below.

* **Python 3.11+**: [Download here](https://www.python.org/downloads/)  
  Bash  
  python --version
* **Node.js 16+**: [Download here](https://nodejs.org/en/download/)  
  Bash  
  node --version
* **WSL (Ubuntu)**: Required for running Redis. [Install guide](https://learn.microsoft.com/en-us/windows/wsl/install)  
  Bash  
  wsl --version
* **Git**: [Download here](https://git-scm.com/downloads)  
  Bash  
  git --version

### 🚀 Complete Setup Guide

#### Phase 1: Redis Setup (WSL)

Redis is used by the backend to temporarily store OAuth state tokens during the authentication flow.

1. **Open WSL Terminal**: You can do this by either searching for "Ubuntu" in the Start Menu or by typing wsl in Command Prompt or PowerShell.  
   Bash  
   # From Command Prompt  
   wsl
2. **Install and Start Redis**: In the WSL terminal, run these commands to install and start the Redis server.  
   Bash  
   # Update packages  
   sudo apt update  
     
   # Install Redis  
   sudo apt install -y redis-server  
     
   # Start Redis  
   sudo service redis-server start  
     
   # Enable Redis to automatically start on boot  
   sudo systemctl enable redis-server
3. **Verify Redis is Running**: To confirm Redis is active, run the following commands. The output should confirm that the service is running and listening on port 6379.  
   Bash  
   # Check Redis status  
   sudo service redis-server status  
     
   # Check if Redis is listening on port 6379  
   ss -ltnp | grep 6379  
     
   # Test Redis CLI  
   redis-cli ping  
   # Expected output: PONG

#### Phase 2: Backend Setup

The backend is built with FastAPI and handles all API requests and OAuth flows.

1. **Navigate to Backend Directory**: In a new Command Prompt window, go to the backend folder.  
   Bash  
   cd C:\Users\YourName\Desktop\integrations\_technical\_assessment\backend
2. **Create and Activate Python Environment**: Create a virtual environment and activate it. This isolates the project's dependencies.  
   Bash  
   # Create virtual environment  
   python -m venv .venv  
     
   # Activate virtual environment  
   .venv\Scripts\activate  
     
   # Your command prompt should now show "(.venv)" at the beginning.
3. **Install Backend Dependencies**: Install all the necessary Python packages.  
   Bash  
   # Upgrade pip to the latest version  
   python -m pip install --upgrade pip  
     
   # Install dependencies from requirements.txt  
   pip install -r requirements.txt
4. **Create Environment File**: The application uses environment variables for sensitive data like API keys and secrets. Create a new file named .env based on the provided example.  
   Bash  
   # Create a copy of the example file  
   copy .env.example .env  
     
   # Open the .env file with Notepad (or your preferred text editor)  
   notepad .env
5. **Add Your HubSpot Credentials**: Open the newly created .env file and replace the placeholder values with your actual HubSpot credentials. You can get these by following the steps in the **"How to Use the Application"** section.  
   Ini, TOML  
   # HubSpot OAuth (REPLACE WITH YOUR VALUES)  
   HUBSPOT\_CLIENT\_ID=your\_hubspot\_client\_id\_here  
   HUBSPOT\_CLIENT\_SECRET=your\_hubspot\_client\_secret\_here  
   HUBSPOT\_REDIRECT\_URI=http://localhost:8000/integrations/hubspot/oauth2callback  
     
   # Redis (keep as is)  
   REDIS\_HOST=localhost
6. **Start Backend Server**: Start the FastAPI backend server. It will run on port 8000.  
   Bash  
   # Load environment variables into the current session  
   set -a; source .env; set +a  
     
   # Start the server  
   uvicorn main:app --host 0.0.0.0 --port 8000 --reload  
     
   You should see output indicating the server is running. You can verify it by navigating to http://localhost:8000/ in your browser.

#### Phase 3: Frontend Setup

The frontend is a React application that provides the user interface.

1. **Open New Command Prompt**: Keep the backend server running. Open a **new Command Prompt** window and navigate to the frontend folder.  
   Bash  
   cd C:\Users\YourName\Desktop\integrations\_technical\_assessment\frontend
2. **Install Frontend Dependencies**: Install all the necessary Node.js packages.  
   Bash  
   # Install Node.js packages  
   npm install
3. **Start Frontend Server**: Start the React development server. It will run on port 3000.  
   Bash  
   npm start  
     
   Your browser should automatically open to http://localhost:3000/, where you will see the **Data Integration Hub** interface.

### 🎯 How to Use the Application

Before you can use the HubSpot integration, you need to set up an app in the HubSpot Developer Portal.

#### Step 1: Get HubSpot Credentials

1. **Create a HubSpot App**:
   * Go to the [HubSpot Developer Portal](https://developers.hubspot.com/).
   * Click "Create app".
   * Give your app a name (e.g., "My Integration App").
   * Click "Create app" to continue.
2. **Configure OAuth**:
   * In the app's settings, go to the "Auth" tab.
   * Add the following Redirect URL: http://localhost:8000/integrations/hubspot/oauth2callback.
   * Under "Scopes", find and enable the crm.objects.contacts.read scope. This gives your app permission to read contact data.
   * Save your changes.
3. **Get Credentials**:
   * On the "Auth" tab, you'll find the **"Client ID"** and **"Client Secret"**.
   * Copy these values and paste them into the HUBSPOT\_CLIENT\_ID and HUBSPOT\_CLIENT\_SECRET fields in your backend/.env file.

#### Step 2: Test the Integration

1. **Open the Application**: Open your browser and navigate to http://localhost:3000. You should see the application's dark-themed interface.
2. **Connect to HubSpot**:
   * From the **"Integration Type"** dropdown, select **"HubSpot"**.
   * Click the **"Connect to HubSpot"** button. This will open a new popup window.
   * In the popup, authorize your application to access your HubSpot account. The window will close automatically.
3. **Load Data**:
   * Once connected, click the **"Load Data"** button.
   * The application will fetch contact data from your HubSpot account and display it in the text area below.

### 🔧 Redis Management in WSL

Here are some useful commands for managing your Redis instance from the WSL terminal.

#### Redis Commands

| Command | Description |
| --- | --- |
| sudo service redis-server start | Starts the Redis service. |
| sudo service redis-server stop | Stops the Redis service. |
| sudo service redis-server restart | Restarts the Redis service. |
| sudo service redis-server status | Checks the current status of Redis. |
| redis-cli | Opens the Redis command-line interface. |
| redis-cli ping | Tests the connection to Redis. |
| redis-cli KEYS "\*" | Lists all keys currently stored in Redis. |
| redis-cli GET "hubspot\_state:\*" | Retrieves the value for a specific key. |
| redis-cli FLUSHALL | **CAUTION**: Deletes all keys in all databases. |
| redis-cli MONITOR | Displays all commands processed by Redis in real time. |

#### Redis Configuration

You can edit the Redis configuration file to change settings like memory limits or binding addresses.

Bash

sudo nano /etc/redis/redis.conf

After making any changes, remember to restart the Redis service with sudo service redis-server restart.