TaskSlinger Installation and Operating Guide

PRACTICE MODULE FOR CERTIFICATE IN:

INTELLIGENT SOFTWARE AGENTS (ISA)

Task Slinger

Task management make simple for people on the go!

Installation and Operating Guide

Team Members: Richard Chai

Contents

1.0 Overview of this Guide

This guide will cover the following:

- Installation of TaskSlinger Chatbot Agent
- Running TaskSlinger AI Models Locally on your laptop
- Running TaskSlinger AI Models on Cloud

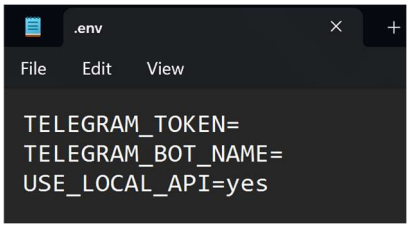
2.0 Installation of TaskSlinger Chatbot Agent

Regardless of whether you run TaskSlinger AI models on cloud or locally, you must complete the installation of TaskSlinger Chatbot Agent (this section)

- Install python 3.11
    - https://www.python.org/downloads/

- Install all the packages found in requirements.txt, which are:
    - aiogram==3.4.1
    - asyncio~=3.4.3
    - python-dateutil==2.9.0.post0
    - gradio_client~=0.14.0
    - gTTS==2.5.1
    - librosa==0.10.1
    - pydub==0.25.1
    - python-dotenv==1.0.1
    - pytz~=2024.1
    - spacy==3.7.4
    - date-spacy==0.0.1
    - soundfile==0.12.1
    - transformers==4.38.2
    - requests==2.31.0
- Download Spacy modules
    - python -m spacy download en_core_web_sm
    - python -m spacy download en_core_web_lg

- Set up the .env file



    - The .env file is found in the same folder as TaskSlinger bot python files.
        - You should insert your own Telegram Token and bot name in this file.
        - If using AI Models locally, set USE_LOCAL_API=yes
        - If using AI Models on cloud, set USE_LOCAL_API=no

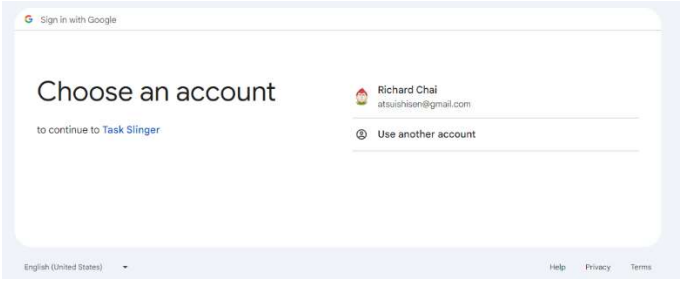
- Install FFMPEG.exe in the same folder as main.py
    - https://www.geeksforgeeks.org/how-to-install-ffmpeg-on-windows/

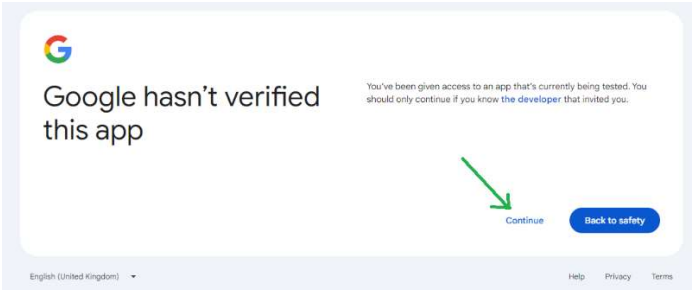
- ***Before you proceed further, please follow the steps in Section 3 and/or Section 4 to install AI Models locally and/or on Cloud.***

- Download the code in this folder:
    - https://github.com/atsui888/Intelligent-Software-Agents/tree/main/Code/01_TaskSlinger_Bot

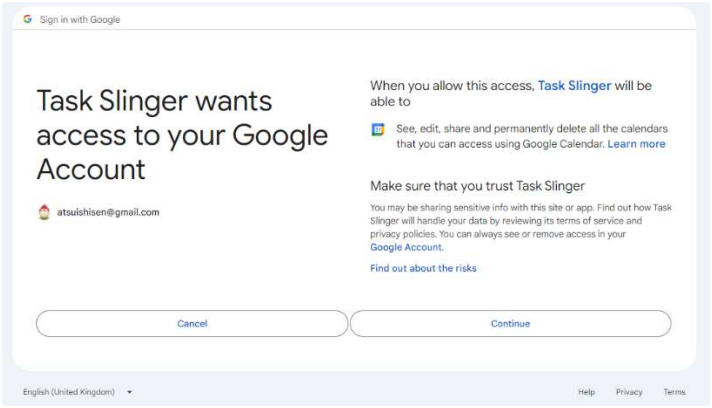
- CD into the folder which holds the code you downloaded, main.py is in it.

- Start TaskSlinger by using the command "python main.py" at the command line. Do remember to first activate your python virtual environment if appropriate.

- Next, you will be prompted to allow the bot to manage your calendar, you must authorise it if you wish to test the bot.
    - If not yet authenticated to allow Task Slinger to use your Google Calendar, the screen below will pop up in your browser.
        - 

        - Click to select the account you want to use.
    - You will now see the following screen when Google informs you that the app has not been verified.
        - 

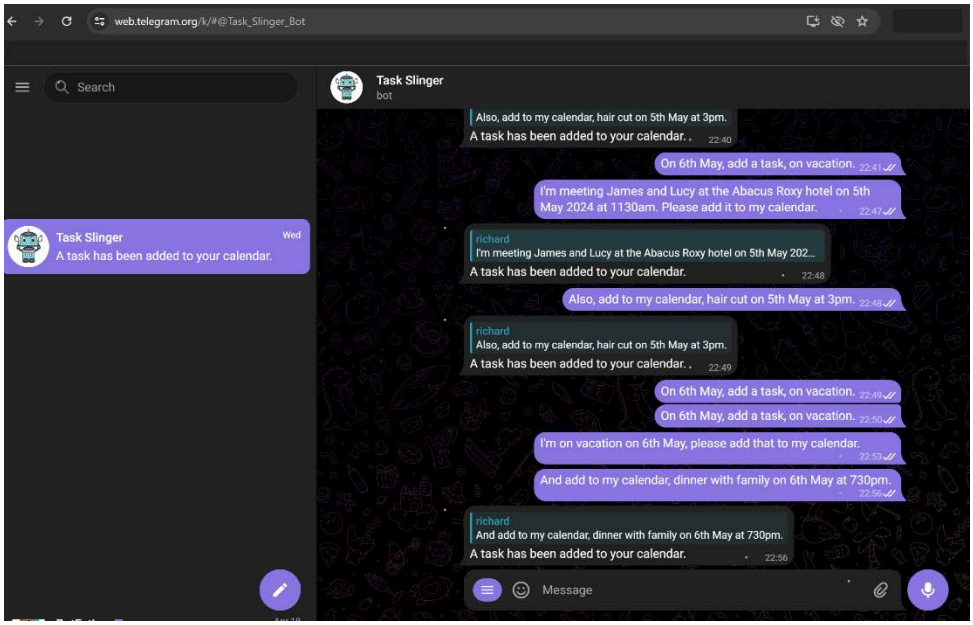
            - Apps that go onto Google Play Store need to be verified but Task Slinger is still in Testing phase.
            - Please click on "continue" to proceed if you wish to test the app. Otherwise, you have the option to watch the videos to see Task Slinger in operation.
    - After you selected to continue, Google will inform you that Task Slinger requires access to your Google Account. (Ideally you would have created a test google account to use).



- Select “continue”.

- When you see the screen below, the Google Authentication for Task Slinger is completed.

- Next, go to your Telegram account and look for TaskSlinger Bot or your own bot which you used your own token and name in the .env file.



- You can now interact with the chatbot!

3.0 Running TaskSlinger AI Models Locally

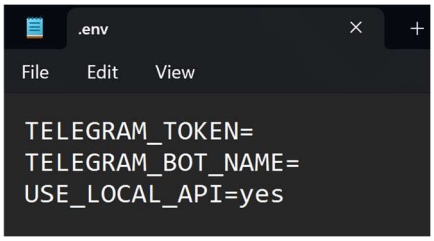
- Install Docker Desktop
    - https://docs.docker.com/desktop/install/windows-install/

- Start running Docker Desktop

- Use the docker pull command to pull the following AI models into your laptop or desktop

    - $> docker image pull rchai/convo_summariser:latest
    - $> docker image pull rchai/isa_intent_classifier:latest
    - $> docker image pull rchai/isa_qna:latest
    -

- Set up the .env file

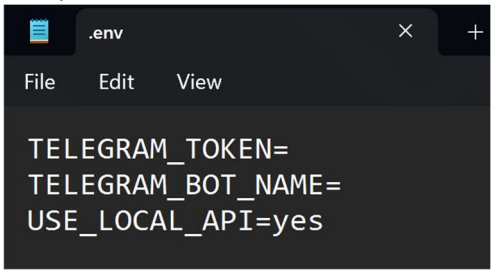


    - The .env file is found in the same folder as TaskSlinger bot python files.
        - You should insert your own Telegram Token and bot name in this file.
        - If you wish to use the docker models you just pulled, set USE_LOCAL_API=yes
- Next, go into the folder where you downloaded the TaskSlinger code
    - At the command line, type “docker compose up”
        - This will spin up the 3 images you pulled into running containers.
    - To shut them down, use the command “docker compose down”.

4.0 Running TaskSlinger AI Models on Cloud

To use the TaskSlinger AI Models on Cloud,

- Set up the .env file



    - The .env file is found in the same folder as TaskSlinger bot python files.
        - You should insert your own Telegram Token and bot name in this file.
        - If you wish to use the AI Models in Cloud, set USE_LOCAL_API=no

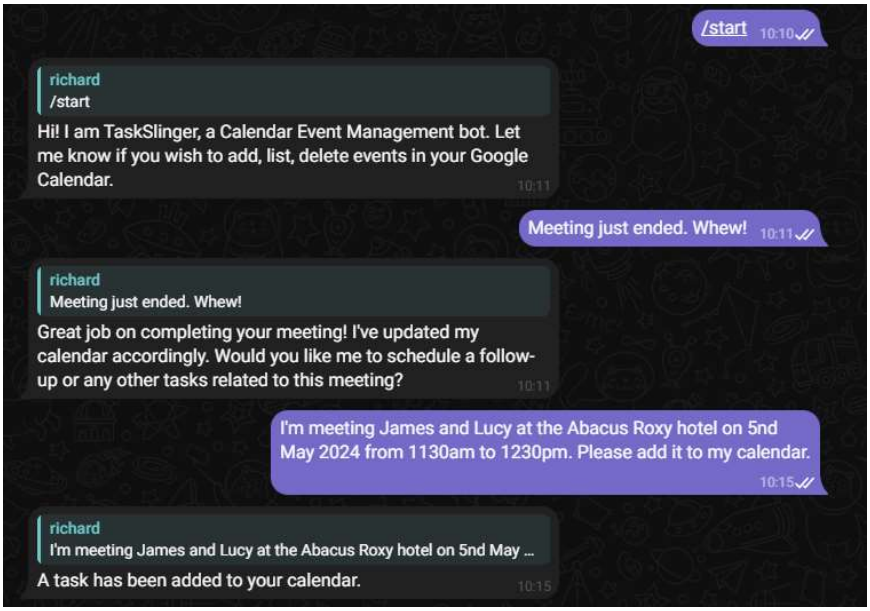
- Please note that when the cloud models are not in use, they go into sleep mode and will take some time to warm up for use.
- Also, to save cost, the models use either free CPU or low cost CPU in cloud, hence the performance may not be that fast.

If you have any questions on installation, please contact Richard at https://www.linkedin.com/in/richardchai/

4.0 Using TaskSlinger Intelligent Chatbot

TaskSlinger does not use scripted dialog flows, instead it tries to understand your intent and respond accordingly.

Example:



And this is what TaskSlinger is thinking: