

PROMPT 1a: Email Assistance Agent

We have made an email assistant agent using Crew AI only to improve rough emails and draft them. It takes only rough emails as input and returns professional version with subject line.

- We have enabled Gmail API on Google Cloud Console
- We have completed OAuth process on Google Cloud Console and downloaded `credentials.json`
- We have installed google-api-python-client, google-auth-oauthlib and google-auth-httplib2

You are a Crew AI Agent Developer. Your task is to develop code to:

- access Gmail
- Integrate the output of email agent
- Parse subject and email body
- Draft it on Gmail

Generate the entire runnable code

Agent Code:

...

```
from crewai import LLM, Agent, Task, Crew
```

```
llm = LLM(
```

```
    model= "gemini/gemini-2.0-flash",
```

```
    api_key= GEMINI_API_KEY,
```

```
    temperature= 0.7
```

```
)
```

```
agent= Agent(
```

```
    role= "email assistant",
```

```
    goal= "Improve emails and make them sound more professional.",
```

```
    backstory= "A highly experienced communication expert and skilled in professional email writing and editing.",
```

```
    llm= llm
```

```
    # tools= [jt]
```

```
)
```

```
task= Task(
```

```
    description= f""" Take the following rough email and rewrite it into a professional and polished version. Expand Abbreviations:
```

```

    "{email}"",
    agent= agent,
    expected_output= "A professionally written email with proper formatting and content.",
)
crew= Crew(
    agents= [agent],
    tasks= [task],
)
result= crew.kickoff()
print(result)
'''

```

PROMPT 1b: Email Assistance Agent (debugging)

'CrewOutput' object has no attribute 'splitlines'

PROMPT 1c: Email Assistance Agent (debugging)

Cell In[6], line 137

```
134 print("Parsed Body:\n", body)
```

```
136 # Step 3: Gmail Authentication
```

```
--> 137 creds = gmail_authenticate()
```

```
138 service = build('gmail', 'v1', credentials=creds)
```

```
140 # Step 4: Draft Email in Gmail
```

Cell In[6], line 22

```
20 if not creds or not creds.valid:
```

```
21 if creds and creds.expired and creds.refresh_token:
```

```
---> 22 creds.refresh(Request())
```

```
23 else:
```

```
24 # credentials.json must be in the same folder 25
```

```
flow = InstalledAppFlow.from_client_secrets_file('credentials.json', SCOPES)
```

NameError: name 'Request' is not defined

PROMPT 2: Tkinter GUI

You are a Python developer. We have prepared the backend code for a business communication tool (email and cover letter). Your task is to generate code for the frontend using Tkinter.

- the interface should have following options
- Agent: Email, Cover Letter (dropdown)
- If email -> "enter rough email" in the window below.
- If cover letter -> "enter file path" for manual input and "enter rough content" in the window below
- Include a terminate button so the program runs even after execution and only closes when terminate button is pressed.
- integrate the code below, other dependencies are already included in main file:

```
'''
```

Email Agent Code:

```
'''
```

```
'''
```

Cover Letter Agent Code:

```
'''
```