1. **Install Flask and spaCy**

pip install Flask

pip install spacy

1. **Download the English language model for spaCy**

python -m spacy download en\_core\_web\_sm

1. **Import the required libraries and load the spaCy model**

import spacy

from spacy.matcher import Matcher

from flask import Flask, request, jsonify

nlp = spacy.load("en\_core\_web\_sm")

matcher = Matcher(nlp.vocab)

1. **Define the pattern for matching the scheduling command**

pattern = [{"LOWER": "schedule"}, {"LOWER": "a"}, {"POS": "NOUN"}, {"LOWER": "on"}, {"LOWER": {"IN": ["at", "on"]}}, {"ENT\_TYPE": "DATE", "OP": "?"}, {"ENT\_TYPE": "TIME", "OP": "?"}, {"LOWER": {"IN": ["with", "between"]}}, {"ENT\_TYPE": "PERSON", "OP": "+"}, {"LOWER": {"IN": ["for", "about", "regarding"]}, "OP": "?"}, {"LOWER": {"IN": ["a", "an"]}, "OP": "?"}, {"LOWER": {"IN": ["hour", "hours", "minute", "minutes"]}, "OP": "?"}]

matcher.add("Schedule", [pattern], on\_match=None)

1. **Define the API endpoint that handles the scheduling command**

app = Flask(\_\_name\_\_)

def jls\_extract\_def(matcher, doc, i, matches):

    return matches

@app.route('/schedule', methods=['POST'])

def schedule():

    text = request.get\_json()['text']

    doc = nlp(text)

    matches = matcher(doc)

    if len(matches) == 0:

        return jsonify({'error': 'No matches found.'})

    match\_id, start, end = matches[0]

    task = doc[start+2:end].text

    date = None

    time = None

    for ent in doc.ents:

        if ent.label\_ == 'DATE':

            date = ent.text

        elif ent.label\_ == 'TIME':

            time = ent.text

    attendees = [ent.text for ent in doc.ents if ent.label\_ == 'PERSON']

    duration = None

    for i, token in enumerate(doc):

        if token.lower\_ in ['hour', 'hours', 'minute', 'minutes']:

            duration = float(doc[i-1].text)

    return jsonify({

        'task': task,

        'date': date,

        'time': time,

        'attendees': attendees,

        'duration': duration

    })

1. **Run the API**
2. if \_\_name\_\_ == '\_\_main\_\_':
3. app.run()

**Now, we can test the API using tools like Postman or cURL by sending a POST request to the /schedule endpoint with the scheduling command in the request body as a JSON object:**

{

    "text": "Schedule a meeting next week between John, Mary, and Jane to discuss the new product launch. We'll need 1 hour."

}

**The API will return the extracted information in JSON format:**

{

    "attendees": ["John", "Mary", "Jane"],

    "date": "next week",

    "duration": 1.0,

    "task": "meeting to discuss the new product launch",

    "time": null

}