Here’s a \*\*summary of all the steps\*\* involved in creating the \*\*backend-only weather app\*\* using FastAPI:

---

### \*\*1. Set Up the Project\*\*

- Install dependencies:

```bash

pip install fastapi uvicorn requests

```

- Create a project structure:

```

weather\_app\_backend/

│

└── main.py

```

---

### \*\*2. Integrate OpenWeatherMap API\*\*

- Sign up at [OpenWeatherMap](https://openweathermap.org/api) to get an API key.

- Use the API key in the code to fetch weather data.

---

### \*\*3. Create API Endpoints\*\*

- \*\*Initialize FastAPI\*\*:

```python

from fastapi import FastAPI, HTTPException

import requests

app = FastAPI()

```

- \*\*Define the `/weather` endpoint\*\*:

- Accepts a `city` parameter.

- Fetches weather data from OpenWeatherMap API.

- Returns weather details (city, country, temperature, humidity, wind speed, description).

```python

@app.get("/weather")

def get\_weather(city: str):

params = {

"q": city,

"appid": API\_KEY,

"units": "metric"

}

response = requests.get(WEATHER\_URL, params=params)

if response.status\_code != 200:

raise HTTPException(status\_code=response.status\_code, detail="Error fetching weather data")

weather\_data = response.json()

return {

"city": weather\_data["name"],

"country": weather\_data["sys"]["country"],

"temperature": weather\_data["main"]["temp"],

"humidity": weather\_data["main"]["humidity"],

"wind\_speed": weather\_data["wind"]["speed"],

"description": weather\_data["weather"][0]["description"]

}

```

- \*\*Define the root endpoint\*\*:

- Returns a welcome message.

```python

@app.get("/")

def read\_root():

return {"message": "Welcome to the Weather App! Use /weather?city=<city\_name> to get weather data."}

```

---

### \*\*4. Run the App\*\*

- Start the FastAPI app using `uvicorn`:

```bash

uvicorn main:app --reload

```

- Access the app at `http://127.0.0.1:8000`.

---

### \*\*5. Test the API\*\*

- Use the `/weather` endpoint to fetch weather data:

```

http://127.0.0.1:8000/weather?city=London

```

- Example response:

```json

{

"city": "London",

"country": "GB",

"temperature": 15.5,

"humidity": 72,

"wind\_speed": 4.6,

"description": "clear sky"

}

```

---

### \*\*6. Optional: Add News API Integration\*\*

- Define a `/news` endpoint to fetch news articles:

```python

@app.get("/news")

def get\_news(query: str = "Apple"):

params = {

"q": query,

"apiKey": NEWS\_API\_KEY

}

response = requests.get(NEWS\_URL, params=params)

if response.status\_code != 200:

raise HTTPException(status\_code=response.status\_code, detail="Error fetching news data")

return response.json()

```

- Test the `/news` endpoint:

```

http://127.0.0.1:8000/news?query=Apple

```

---

### \*\*7. Deploy the Backend\*\*

- Deploy the app using platforms like:

- \*\*Heroku\*\*

- \*\*Vercel\*\*

- \*\*AWS Lambda\*\*

- \*\*Docker\*\*

---

### \*\*Summary of Key Steps\*\*

1. \*\*Install Dependencies\*\*: FastAPI, Uvicorn, and Requests.

2. \*\*Set Up FastAPI\*\*: Initialize the app and define endpoints.

3. \*\*Integrate OpenWeatherMap API\*\*: Fetch weather data using the API key.

4. \*\*Create Endpoints\*\*:

- `/weather`: Fetch and return weather data.

- `/`: Welcome message.

- `/news` (optional): Fetch news articles.

5. \*\*Run the App\*\*: Use `uvicorn` to start the server.

6. \*\*Test the API\*\*: Use a browser, Postman, or cURL to test endpoints.

7. \*\*Deploy\*\*: Deploy the backend to a cloud platform.

---

This summary provides a clear, step-by-step overview of the entire process. Let me know if you need further clarification or assistance!