

# API Documentation—Weather Stylist

## API Overview:

The function of this API is to power the AI Weather Stylist application. It then provides endpoints to retrieve real-time weather data and generate outfit recommendations using Cohere, and it also enforces a secure credit-based system for user access and functionality.

- [API Documentation Page for Streamlit](#)

```
import streamlit as st
```

### 1. Set up the page.

```
st.set_page_config(page_title="API Documentation", layout="wide")
```

```
st.title("Weather Stylist API Documentation")
```

### 2. Introduction

```
st.markdown("""
```

Welcome to the Weather Stylist API!

This API allows users to get weather updates and receive outfit recommendations powered by **Cohere AI**.

```
---
```

- Authentication

All API calls require a login. Once logged in, your app will use a Supabase token to call the API.

**\*\*Header Example:\*\***

**Authorization: Bearer YOUR\_SUPABASE\_JWT\_TOKEN**

---

**Endpoint 1: `/weather`**

**\*\*Use:\*\*** Get the current weather and suggestions.

**\*\*Method:\*\*** `GET`

**\*\*Example Request:\*\***

```bash

```
curl -H "Authorization: Bearer YOUR_TOKEN" "https:// http://127.0.0.1:8000  
/weather?city=Lincoln"
```

**Response**

```
{  
  
  "city": "London",  
  
  "temperature": 15,  
  
  "condition": "Rainy",  
  
  "suggestion": "Wear a waterproof jacket and boots"  
}
```

## **Endpoint 2: /user/credits**

Use: Check how many styling credits you have.

Method: GET

### **Response:**

```
{  
  
  "email": "user@example.com",  
  
  "credits": 3  
}
```

## **Endpoint 3: /generate-style**

Use: Generate a clothing suggestion based on weather

Method: POST

### **Body:**

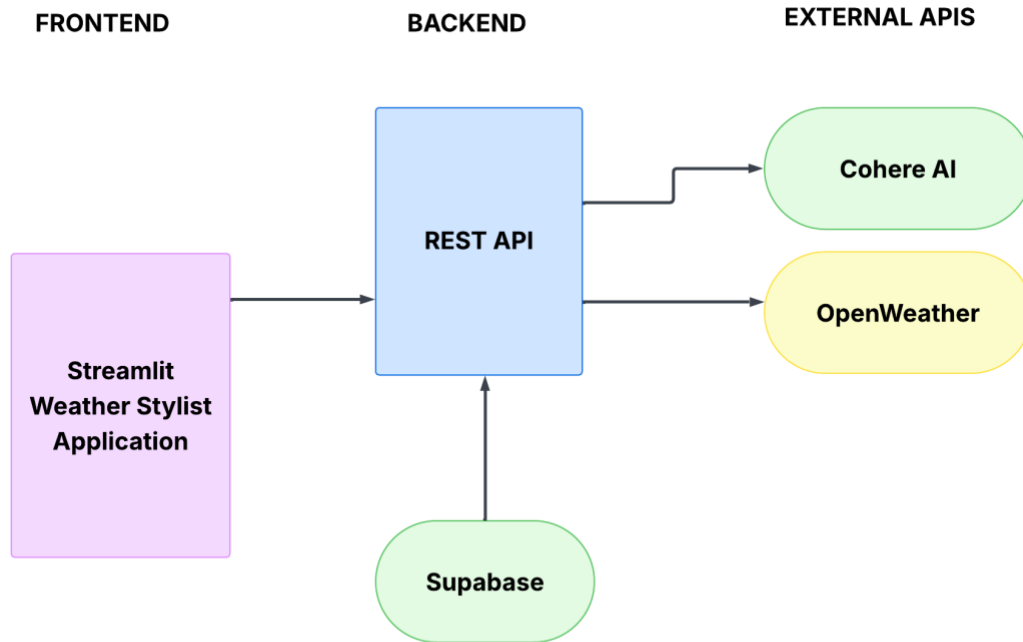
```
{  
  
  "temperature": 25,  
  
  "weather_condition": "Sunny"  
}
```

### **Response**

```
{  
  
  "Outfit": "Try light linen shirts and shorts with sandals"  
}
```

**Note:** every call to this endpoint requires 1 credit.

## Architecture Diagram



## Weather Stylist Application

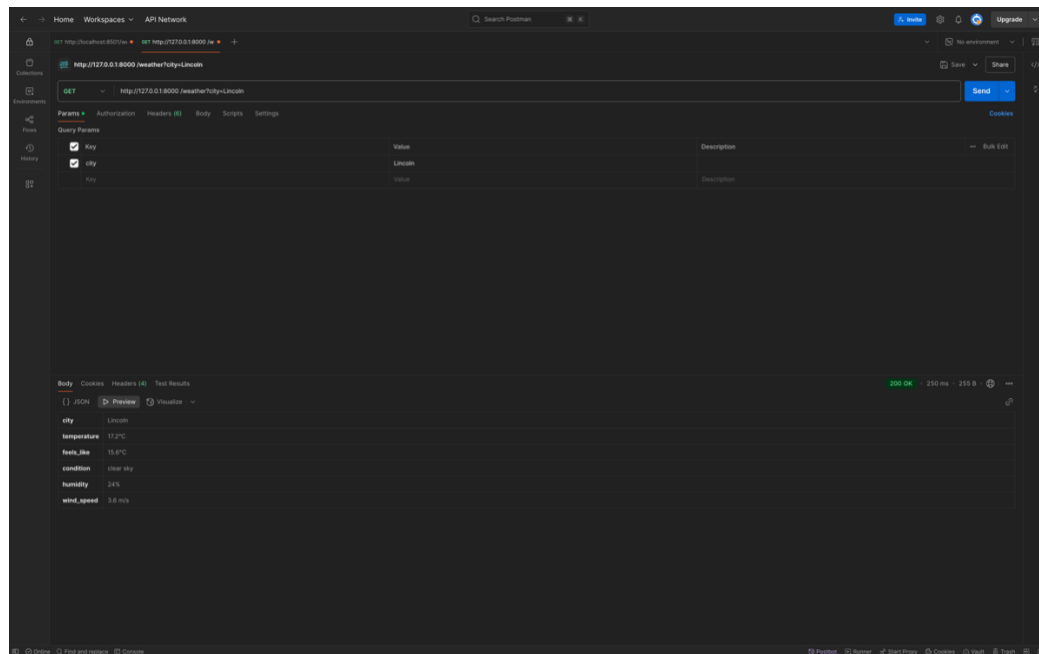
### Technology Used

- Supabase (Authentication and Database)
- Cohere (Outfit Generation)
- Streamlit (Frontend)
- Python + FastAPI
- Postman (Testing)
- Pytest & Unittest (Test code)

## Endpoint Documentation:

| Method | Endpoint          | Description                         | Auth | Rate Limit    |
|--------|-------------------|-------------------------------------|------|---------------|
| GET    | /weather?city=... | Fetches live weather info           | Yes  | 3 credits/day |
| POST   | /ask-ai           | Gets AI suggestion based on weather | Yes  | 1 credit/use  |

## Example Request and Response (Postman)



## Authentication:

Supabase is used to secure authentication. Users must create an account, then log in and receive a session token, which is included in the Authorization header for every request made by the user. Only registered users will be authorized to access the functionality of checking the weather updates based on their credit availability and getting outfit recommendations from the AI.

## Credit Limiting:

When a user successfully creates an account, the user is given a limited credit. If credit runs out, the user is directed to purchase more credits, and further API calls are blocked until they reset every 24 hours or the user makes a credit purchase.

Credit Deduction Screenshot

