

```
!pip install dash
```

```
Requirement already satisfied: dash in /usr/local/lib/python3.10/dist-packages (2.16.1)
Requirement already satisfied: Flask<3.1,>=1.0.4 in /usr/local/lib/python3.10/dist-packages (from dash) (2.2.5)
Requirement already satisfied: Werkzeug<3.1 in /usr/local/lib/python3.10/dist-packages (from dash) (3.0.2)
Requirement already satisfied: plotly>=5.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (5.15.0)
Requirement already satisfied: dash-html-components==2.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (2.0.0)
Requirement already satisfied: dash-core-components==2.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (2.0.0)
Requirement already satisfied: dash-table==5.0.0 in /usr/local/lib/python3.10/dist-packages (from dash) (5.0.0)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.10/dist-packages (from dash) (7.1.0)
Requirement already satisfied: typing-extensions>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from dash) (4.11.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from dash) (2.31.0)
Requirement already satisfied: retrying in /usr/local/lib/python3.10/dist-packages (from dash) (1.3.4)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.10/dist-packages (from dash) (1.6.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from dash) (67.7.2)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (3.1.3)
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-packages (from Flask<3.1,>=1.0.4->dash) (8.1.7)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly>=5.0.0->dash) (8.2.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly>=5.0.0->dash) (24.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from Werkzeug<3.1->dash) (2.1.5)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata->dash) (3.18.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->dash) (3.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->dash) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->dash) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->dash) (2024.2.2)
Requirement already satisfied: six>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from retrying->dash) (1.16.0)
```

```
import requests
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
```

```
# Create a Dash application
app = dash.Dash(__name__)
```

```
# Define the layout of the dashboard
app.layout = html.Div([
    html.H1("Flight Information Dashboard"),
    html.Label('Enter Flight Number:'),
    dcc.Input(id='flight-input', type='text', value='DL993'),
    html.Button('Get Flight Information', id='get-flight-button', n_clicks=0),
    html.Div(id='flight-output')
])
```

```
# Define a function to fetch flight information for a specific flight number
def get_flight_information(flight_number):
    api_key = 'dd57d0d1c2cc386972d8ec3c6c7826ca'
    url = f'http://api.aviationstack.com/v1/flights?access_key={api_key}&flight_iata={flight_number}'
    response = requests.get(url)
    data = response.json()
    return data
```

```
# Define callback function to update flight information based on user input
```

```
@app.callback(
    Output('flight-output', 'children'),
    [Input('get-flight-button', 'n_clicks')],
    [dash.dependencies.State('flight-input', 'value')]
)
def update_flight_information(n_clicks, flight_number):
    if n_clicks > 0:
        flight_data = get_flight_information(flight_number)
        if 'error' in flight_data:
            return f"Error: {flight_data['error']['info']}"
        else:
            flight = flight_data['data'][0]
            departure = flight['departure']
            arrival = flight['arrival']
            airline = flight['airline']
            flight_number = flight['flight']['number']
            flight_status = flight['flight_status']
```

```
    return html.Div([
        html.H2(f"Flight Information for {airline['name']} {flight_number}"),
        html.P(f"Flight Status: {flight_status}"),
        html.H3("Departure"),
        html.P(f"Airport: {departure['airport']} ({departure['iata']})")
    ])
```

```

    name = f"Airport: {departure['airport']} ({departure['iata']})",
    html.P(f"Terminal: {departure['terminal']}, Gate: {departure['gate']}",),
    html.P(f"Scheduled Departure: {departure['scheduled']}",),
    html.P(f"Estimated Departure: {departure['estimated']}",),
    html.H3("Arrival"),
    html.P(f"Airport: {arrival['airport']} ({arrival['iata']})",),
    html.P(f"Terminal: {arrival['terminal']}, Gate: {arrival['gate']}, Baggage: {arrival['baggage']}",),
    html.P(f"Scheduled Arrival: {arrival['scheduled']}",),
    html.P(f"Estimated Arrival: {arrival['estimated']}",)
  ])

# Run the Dash application
if __name__ == '__main__':
    app.run_server(debug=True)

```

Flight Information Dashboard

Enter Flight Number:

Get Flight Information

Flight Information for IndiGo 928

Flight Status: scheduled

Departure

Airport: Hyderabad Airport (HYD)

Terminal: None, Gate: 119

Scheduled Departure: 2024-04-17T06:50:00+00:00

Estimated Departure: 2024-04-17T06:50:00+00:00

Arrival

Airport: Surat (STV)

Terminal: None, Gate: None, Baggage: None

Scheduled Arrival: 2024-04-17T08:15:00+00:00

Estimated Arrival: 2024-04-17T08:15:00+00:00

