In [1]:
```
#!pip install requests plotly dash jupyter-dash
```

In [ ]:
```
import requests
import plotly.graph_objs as go
from jupyter_dash import JupyterDash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
```

In [3]:
```
points = []

# Fetch ISS location data
def fetch_iss_location():
    url = "http://api.open-notify.org/iss-now.json"
    response = requests.get(url)
    data = response.json()

    if data["message"] == "success":
        latitude = float(data["iss_position"]["latitude"])
        longitude = float(data["iss_position"]["longitude"])
        return latitude, longitude
    else:
        return None

# Initialize the Dash app
app = JupyterDash(__name__)

app.layout = html.Div([
    html.Div(id='coordinates', style={'whiteSpace': 'pre-line'}),
    dcc.Graph(id='iss-track'),
    dcc.Interval(id='interval-component', interval=5 * 1000, n_intervals=0)
])

# Update the ISS track and coordinates in real-time
@app.callback(
    [Output('iss-track', 'figure'), Output('coordinates', 'children')],
    [Input('interval-component', 'n_intervals')])
def update_iss_track(n):
    global points

    # Fetch the ISS location every 5 seconds
    location = fetch_iss_location()

    if location:
        points.append(location)

        # Update the map with the new location
        fig = go.Figure(go.Scattergeo(
            lat=[p[0] for p in points],
            lon=[p[1] for p in points],
            mode='lines+markers',
            marker=dict(size=5, color='red'),
            line=dict(width=2, color='blue')
        ))
```

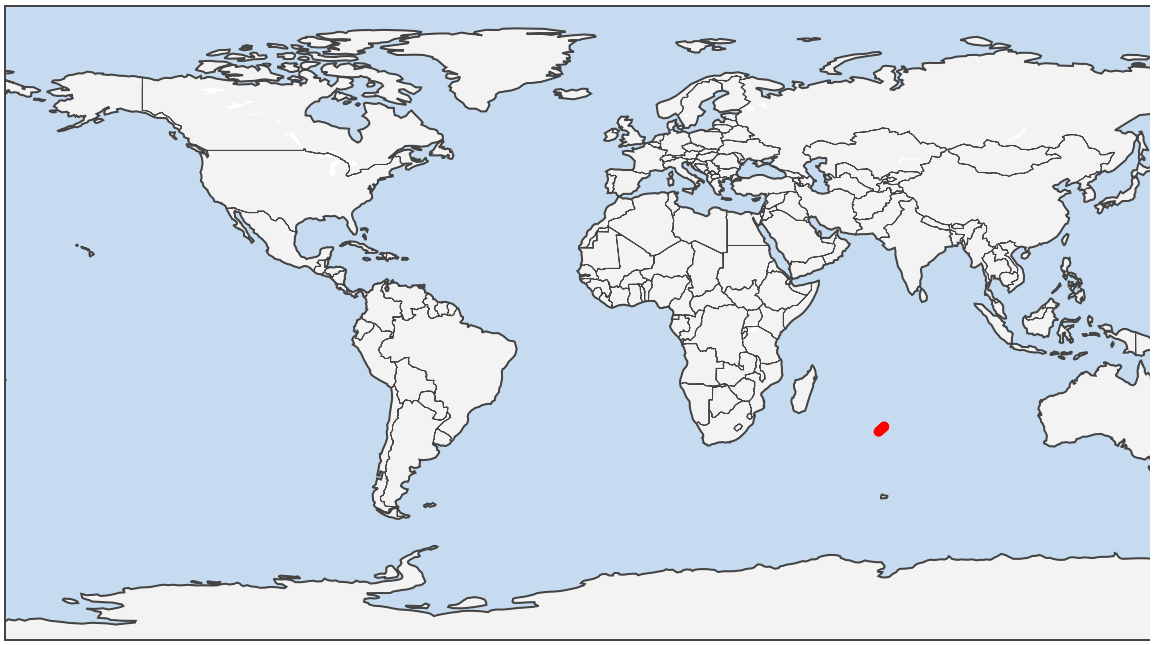
```
        fig.update_layout(
            title="Real-time ISS Tracking",
            geo=dict(
                showland=True,
                showcountries=True,
                showocean=True,
                countrywidth=0.5,
                landcolor="rgb(243, 243, 243)",
                oceancolor="rgb(198, 219, 239)",
                projection_type="equirectangular"
            ),
            margin=dict(l=10, r=10, t=40, b=10),
            autosize=True,
            showlegend=False
        )

        coordinates_text = "\n".join([f"Latitude: {p[0]:.2f}, Longitude: {p[1]:.2f}
        return fig, coordinates_text

    return go.Figure(), "No coordinates available"

# Run the app in the Jupyter Notebook
app.run_server(mode='inline', debug=False)
```

```
Dash is running on http://127.0.0.1:8050/
```

```
WARNING: This is a development server. Do not use it in a production deployment. Use
a production WSGI server instead.
 * Running on http://127.0.0.1:8050
Press CTRL+C to quit
127.0.0.1 - - [14/Apr/2023 04:44:28] "GET /_alive_fa505f42-7820-4e57-90b2-21e1476a41
1a HTTP/1.1" 200 -
```

Latitude: -30.85, Longitude: 68.04
Latitude: -30.60, Longitude: 68.31
Latitude: -30.40, Longitude: 68.54
Latitude: -30.16, Longitude: 68.81
Latitude: -29.95, Longitude: 69.03
Latitude: -29.66, Longitude: 69.35
Latitude: -29.46, Longitude: 69.57
Latitude: -29.41, Longitude: 69.62

Real-time ISS Tracking



```
127.0.0.1 - - [14/Apr/2023 04:44:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Apr/2023 04:44:28] "GET /_dash-layout HTTP/1.1" 200 -
127.0.0.1 - - [14/Apr/2023 04:44:28] "GET /_dash-dependencies HTTP/1.1" 200 -
127.0.0.1 - - [14/Apr/2023 04:44:28] "GET /_dash-component-suites/dash/dcc/async-gra
ph.js HTTP/1.1" 304 -
127.0.0.1 - - [14/Apr/2023 04:44:28] "GET /_dash-component-suites/dash/dcc/async-plo
tlyjs.js HTTP/1.1" 304 -
127.0.0.1 - - [14/Apr/2023 04:44:29] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [14/Apr/2023 04:44:34] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [14/Apr/2023 04:44:39] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [14/Apr/2023 04:44:44] "POST /_dash-update-component HTTP/1.1" 200 -
127.0.0.1 - - [14/Apr/2023 04:44:49] "POST /_dash-update-component HTTP/1.1" 200 -
```

In [ ]: