Flight Data Management Application

Step 1: Create a RESTful API for Flight Data

You are tasked with building a simple application that provides RESTful APIs to manage flight data including insert, update, delete, and retrieve data. Each flight record includes the following fields:

- **Airline**: The name of the airline.
- **Supplier**: The supplier providing the flight data.
- **Fare**: The total price of the flight.
- **Departure Airport**: 3-letter airport code.
- **Destination Airport**: 3-letter airport code.
- **Departure Time**: ISO_DATE_TIME format (UTC timezone).
- **Arrival Time**: ISO_DATE_TIME format (UTC timezone).

Clients can also search and filter flight data based on origin, destination, airline, departure, and arrival time.

Step 2: Integrating CrazySupplier Data

Recently, this company partnered with a new supplier called CrazySupplier. Unlike other suppliers, we cannot store their data in our database; it is only accessible via an API. We need to combine their data with our existing flight data in the search API.

CrazySupplier API Details

Endpoint: `POST` https://api.crazy-supplier.com/flights
Request Body:

- from: 3-letter airline code
- to: 3-letter airline code
- outboundDate: ISO_LOCAL_DATE format – **CET timezone**
- inboundDate: ISO_LOCAL_DATE format – **CET timezone**

Response:

- carrier: Airline name
- basePrice: price without tax
- tax: tax which should be added to the basePrice
- departureAirportName: 3-letter airport name
- arrivalAirportName: 3-letter airport name
- outboundDateTime: ISO_LOCAL_DATE format – **CET timezone**
- inboundDateTime: : ISO_LOCAL_DATE format – **CET timezone**

The implementation should be almost ready for production.
Use MAVEN and mention any assumptions if needed.
Do not forget the tests.