JS code task

Front end (part 1)

You are building a flight search interface.

Create a search form with three fields:

1. From location (eg: Sydney)
2. To location (eg: Heathrow)
3. Travel date (eg: 2016-09-02)

When the user clicks search, your front end code needs to make an AJAX request to a small backend server you will build, which will in turn contact the Flight API.

Usability

Flights may be cheaper on some days than others;
If the user picks `2016-09-10` as their travel date, then query their chosen date and nearby dates (+/- 2):
`2016-09-08`, `2016-09-09`, **`2016-09-10`**, `2016-09-11`, `2016-09-12`

A great way to present all five days of results would be by using tabs.

You don't need to present all information returned by the flight search.
Show what you think is relevant to the user.

> **Important**
>
> - Don't use any frameworks based on React, Angular or similar.
>   You may however use utility libraries like jQuery/UI, Bootstrap, lodash & moment.js.
> - We use coffeescript, however you can also use plain javascript or typescript.
> - Do not render HTML on the server; send data to the frontend and render it there.

Back end (part 2)

Build a back end server which has three endpoints:

1. `/airlines`
   Lists all available airlines from the Flight API.
2. `/airports`
   Lists all matching airports from the Flight API.
3. `/search`
   Accepts all parameters from the search form above.
   For a single flight search, you will need to make multiple Flight API requests:
   - `/airlines` to get a list of airlines
   - `/flight_search/:airline_code` to search for flights for each airline

   The list of airlines may change, so caching is not an option.

Important

- Stick to small frameworks like expressjs, koa, sinatra, tornado or similar.
- We use node.js, however you're free to use ruby, python or similar.

Flight API (provided)

Please use our Flight API to complete this task.

It has three endpoints:

- `airlines`
  Provides a list of airlines. Takes no parameters.
  The airline codes from the response is required for the flight search.
  Example request: airlines
- `airports`
  Provides an airport search.
  Query parameters:
  - `q` - text based search param

  The `airportCode` from the response is required for the flight search.
  Example request: airports?q=Melbourne
- `flight_search/:airline_code`
  Provides a list of available flights for a single airline.
  URL parameters:

- `airline_code` - airline code from the airlines endpoint

Query parameters:

- `date` departure date, `YYYY-MM-DD`
- `from` origin airport code, eg: `SYD`
- `to` destination airport code, eg: `JFK`

Example request: `flight_search/QF?date=2018-09-02&from=SYD&to=JFK`

Expectations

- Documentation is not necessary; try to write self documenting code & document only where required.
- We're looking for well structured & architected code (but please don't build something turing complete).

Submission

Please send us a zip/gz/link to repository with your solution.

Provide a `start.sh` script in the root folder which will perform all required package installation, and launch the server on port 3000.