EatClub Backend Developer Technical Challenge

Your mission, should you choose to accept it, is to build an API that consumes and transforms data about restaurant deals.

- You must use the Java programming language
- Use Maven as necessary to manage packages

Submit your full Maven project in the form of a public GitHub repository link. Push a few commits showing how you incrementally built your solution and solved problems.

How we will test your solution

We will have a look at the code on GitHub, and some of your commits.

We will test your API by checking out (cloning) your code and running it in a development environment.

EatClub

Task 1

Write an API that returns a list of all the available restaurant deals that are active at a specified time of day.

The API will take a single parameter named **timeOfDay** as a string parameter.
Eg: **timeOfDay** = "10:30" will query for all active deals at 10.30am.

The API will return an array of objects containing all matching deals, that looks like this:

```
{
    "deals": [
        {
            "restaurantObjectId": "...",
            "restaurantName": "...",
            "restaurantAddress1": "...",
            "restarantSuburb": "...",
            "restaurantOpen": "...",
            "restaurantClose": "...",
            "dealObjectId": "...",
            "discount": "...",
            "dineIn": "...",
            "lightning": "...",
            "qtyLeft": "..."
        }
    ]
}
```

You can query this API for the necessary data: https://eccdn.com.au/misc/challengedata.json

Testing your code

What will the API return for each of these cases:
- **timeOfDay** = 3:00pm
- **timeOfDay** = 6:00pm
- **timeOfDay** = 9:00pm

EatClub

Task 2

Write an API that calculates the 'peak' time window, during which most deals are available.

This API does not take any parameters.

The API will return an object that looks like this:

```
{
    "peakTimeStart": "...",
    "peakTimeEnd": "..."
}
```

Task 3 - Bonus Points

For bonus points you can choose one (or both!) of the following tasks:

- Design a DB schema
- Deploy your solution to AWS

Design a DB Schema

Select a database technology, and design a schema that you would use to store the data that is returned by this API:

https://eccdn.com.au/misc/challengedata.json

- You do not need to build or deploy this database, just provide a design for the schema

- Tell us which database you will be using to host this schema and a brief description of why you chose this database technology

- Provide your design in the form of a diagram, accompanied by brief notes/comments so that we can understand it.

    Provide the diagram in a file that can be easily viewed, eg a jpg/pdf printout or screen capture, mermaid diagram, draw.io diagram.

Deploy your solution to AWS

Deploy your solution to AWS. Grant us access to an AWS account including all AWS services that comprise your solution, and tell us how to test your solution.