REST API Design Assignment

Overview

- Designing a good API for any given business process requires having a strong understanding of the entire process. This helps ensure you have thought of the different use cases, error scenarios, and other design considerations.
- Writing high-quality API documentation is one of the most important steps in designing, building, and having other developers use your API. Luckily, there are pre-defined guidelines/specifications that make it easy for REST APIs to be understood by other software engineers.

Resources

- REST API Design
- OpenAPI specification
- Basic structure of an OpenAPI spec
- Swagger vs OpenAPI
- API overview & API Documentation Best Practices (same source)

Assumptions

- This assignment assumes you have foundational knowledge of what a REST API is. If you don't yet, there are great introduction articles and youtube explanations available. Start with those first!

Instructions

- *Review all links in the "Resources" section before attempting this assignment!*
- In this assignment you will be drawing out the end-to-end business process of someone who is buying movie tickets online, using a client/server UML Sequence Diagram (example).
    - Specifically, you should highlight process steps that would require a transfer of information between the customer's device and the movie theater's servers (ex: getting movie times, selecting a seat, making a payment, etc.).
    - You can use any online diagram tool (like Lucidchart) for this portion, as long as you do it in the format of a sequence diagram. You can also draw it out by hand and submit a photo as long as it is legible.
    - Note: You should *not* include low-level details like the 3-way handshake between the client and the server, just the high level steps in the business process.
    - **Submit this in the form of an image, pdf, or a link to your diagram.**
- Then, select *one* service in the process you have mapped out (ex: making a payment) and create an OpenAPI specification for that service.
    - Note: You do *not* need to code/implement your API!
    - **Submit your OpenAPI spec in the form of a yaml file attachment or link to your yaml file** (ex: if it's on your github account).
    - *Note: If submitting anything as a link, it must be accessible to anyone with the link. You will not receive credit if your link is private, broken, or inaccessible.*

Grading (6 points total)

- (2 points) Business process sequence diagram with at least 4 potential services/APIs (i.e. 4 round trip request/responses)
    - For each service:
        - 0.25 points for listing (in the diagram) what information would need to be included in the request from the client (you/the customer).
        - 0.25 points for listing (in the diagram) what information would need to be included in the response from the server (movie theater).
- (4 points) OpenAPI specification:
    - Basic information (1 point):
        - Includes the version, “info” section (with non-trivial description), and “paths” section with relevant endpoints *(note: the endpoints do <u>not</u> need to be real/functional)*
    - GET (1 point):
        - Includes summary, parameters (if applicable), and at least *two* responses (ex: 200 OK & 404 Not Found)
    - POST (1 point):
        - Includes summary, requestBody, and at least *two* responses
    - DELETE (1 point):
        - Includes summary and at least *three* responses