

Tech Challenge: Design an API for a Coupon Book Service

Overview

Your task is to design a high-level architecture and API for a service that allows businesses to create, distribute, and manage coupons. This service should support operations such as creating coupon books, assigning coupons to users, locking coupons temporarily during redemption attempts, and redeeming coupons.

System should also include the capability to upload a list of codes to a given coupon book or have the API generate codes following a pattern up to a total amount provided.

Take the following into consideration:

1. Coupon book codes can optionally be redeemed more than once per user (coupon book parameter level)
2. Max number of codes per Coupon Book assigned per member can also be optionally specified (coupon book parameter level)

You are welcome to introduce any necessary constraints or assumptions to streamline the process and ensure the delivery of a more complete solution.

Expected System Design Problems to address:

1. Correct Database (SQL or NoSQL) locking and state management
2. Code redeeming and generation logic
3. Randomness logic when assigning coupon codes

Objectives:

- **Design the System Architecture**: Outline a scalable system architecture, choosing appropriate technologies, databases, and cloud services. Focus on backend technologies and cloud platforms like AWS or GCP.

- **API Design**: Design RESTful API endpoints to support the main functionalities of the service: creating coupons, assigning coupons to users, locking and unlocking coupons, and redeeming coupons.
- **Security and Performance Considerations**: Ensure the API is secure and can handle a high volume of requests efficiently.
- **Concurrency Handling**: Propose a method to manage concurrency, especially for coupon redemption, to prevent race conditions and ensure data integrity.

Deliverables:

- **High Level System Architecture**: Provide an outline for system architecture, including databases, servers, and any external services or APIs.
- **High Level Database Design**
- **API endpoints**: including request and response formats. Highlight how each endpoint interacts with the system's components.
- **Pseudocode for Key Operations**: Include pseudocode for at least three critical operations: assigning a coupon to a user, locking a coupon, and redeeming a coupon. This should demonstrate your approach to handling concurrency and ensuring security.
- **High-Level Deployment Strategy**: Briefly describe how you would deploy this system on a cloud platform like AWS or GCP, considering scalability and availability.

Evaluation Criteria:

- **System Design**: Clarity and scalability of the proposed architecture.
- **API Design**: Completeness, usability, and security of the API.
- **Problem-Solving Skills**: Effectiveness of solutions for handling concurrency and ensuring performance.
- **Technical Knowledge**: Understanding of backend technologies, databases, and cloud deployment strategies.
- **Example API Endpoints and Pseudocode**

Expected API Endpoints with Pseudocode:

- POST /coupons: Create a new coupon book.
- POST /coupons/codes: Upload a code list to an existing Coupon Book (optionally if not generated)
- POST /coupons/assign: Assign a new random coupon code to a user.
- POST /coupons/assign/{code}: Assign a given coupon code to a user.

- POST /coupons/lock/{code}: Lock a coupon for redemption. Code should have been previously assigned to a user. This is a temporary lock operation, but not a definitive redeem.
- POST /coupons/redeem/{code}: Redeem a coupon. Code should have been previously assigned to a user. This is a permanent lock operation.

Expected use cases to be resolved:

- Creation
 - Create a new coupon book
 - Upload a code list to an existing Coupon Book (optionally if not generated)
- Usage
 - Assign a new random coupon code to a user.
 - Assign a specific coupon code to a user.
 - Redeem a coupon. Code should have been previously assigned to a user. This is a permanent lock operation.
 - Get the user's assigned coupon codes