Backend ruby developer test

The Problem

Our application has movies, seasons and episodes. Let's simplify the metadata, so movies and episodes only have a title and a plot, and seasons have title, plot and a number. An episode belongs to a season and a season can have N episodes, the episode also has the number of episode in the season.

Movies and seasons can be purchased (but not episodes), each one can have several purchase options, a purchase option has a price (2.99) and a video quality (HD or SD).

We also have users, to simplify, a user will only have an email.

A user can make a purchase of a content (movie/season) through a purchase option. When a user makes a purchase, we store it in his library, the user has up to 2 days to see the content. In his library, we only show the titles that the user has "alive" and he can see and not the expired purchases. Also, while the user has a content in his library, he can't purchase the same content again.

For instance, if the user purchases the movie '300: Rise of an Empire' and goes to his library, he will find the movie. If he tries to purchase it again, he will receive an error. 3 days after, if he goes to the library, the movie will not appear in his library and he can purchase it again.

Define and implement the following JSON REST API in Ruby:

1. An endpoint to return the movies, ordered by creation.
2. An endpoint to return the seasons ordered by creation, including the list of episodes ordered by its number.
3. An endpoint to return a single list of movies and seasons, ordered by creation.
4. An endpoint for a user to perform a purchase of a content.
5. An endpoint to get the library of a user ordered by the remaining time to watch the content.

Notes:

A. Implement the API following REST principles

B. Use any gem or library that you need.

C. Implement the tests that you consider appropriate

D. Implement a caching mechanism when appropriate

E. There is no need to implement authentication nor authorization. Suppose that the user is already registered and authenticated, you can identify the user in each request by a parameter like user_id.

F. Use a SQL database to persist the data

Deliverable

We expect a git repository link with the project. Or a zip with all the repo (including commits).

Please do not use Rakuten.tv or any other related name for the project. Don't put this problem definition in the repo either. Thanks! 🙂