



Ascend Take-Home Exercise

Overview

At Ascend, one of our product's core functions is providing an online checkout experience, and as part of this exercise, we will build partial functionality to support this.

The goal of this exercise is to **design a REST API and a database schema** to fulfill the use cases outlined below to power a checkout experience. Your API will be the product, and it's meant to be consumed by a third party so consider this when considering tradeoffs.

Concepts

- **Insurance policy** - Will represent the items to be purchased by the customer. For the purpose of the exercise, an insurance policy is composed of the following attributes:
 - Premium - Amount to pay for the policy, net of any fees.
 - Tax Fee - A fee to pay for the policy.
 - Insured Name - The name of the customer purchasing the policy.
- **Finance terms** - The terms that will define the price to pay today. For the purpose of the exercise, finance terms are composed of the following attributes:
 - Downpayment - The amount to be paid initially by the customer. The downpayment will be calculated per insurance policy with the following formula: $(premium * 0.20) + tax\ fee$

- Due date - A date that indicates when the last day the terms will be honored and the customer has to pay by. This will be set by the user.
- Amount financed - The amount of money that will be financed. For the purpose of the exercise, this will be the result of $\text{total amount to pay for each insurance policy} - \text{total downpayment}$.

Example:

Insurance Policy A

Premium \$200

Tax fee: \$50

Insurance Policy B

Premium \$300

Tax fee: \$50

Finance terms:

Downpayment: $\$200 = ((200 * .20) + 50) + ((300 * .20) + 50)$

Amount financed: $\$400 = (200 + 50 + 300 + 50) - 200$

Due Date: 12/12/2023

Requirements

The exercise should implement a solution for the following user stories.

User story 1

A user should be able to create a finance terms agreement and include multiple Insurance policies.

User story 2

A user should be able to agree to the finance terms. The system should keep track of which finance terms have and haven't been accepted.

User story 3

A user should be able to list all the finance terms created. They should be able to filter finance terms by the following two attributes:

- Downpayment amount: greater than, less than, or equal to the amount.
- Status: agreed vs. non-agreed terms

They should also be able to sort the results ascending and descending by downpayment amount and due date in ascending and descending order.

What are we looking for?

- API Design: We're interested in seeing your interpretation of the user stories and how you translated this into an API. Take into consideration:
 - How do you validate payload? What do you consider required vs. optional and any other types of validations?
 - How you design the request schema.
 - How do you deal with errors and surface those to the service user?
 - Think of the API as a Product and assume this will be an API consumed by third parties. Consider the tradeoffs as a Product when modeling the API.
- Data design: We're interested in learning how you model the data based on the requirements and how the data design designs influence your API design and vice versa.
- Product thinking: How you think of these user stories from a product perspective and communicate your understanding of the problem.
- Testing: Add 1 to 2 tests that showcase your approach to testing. We are interested in seeing how you test your work and are not interested in exhaustive tests for the purpose of the exercise.

Guidelines

Timing

We expect this exercise to take around 2 - 3 hours. Please complete as much as you can during that time and be honest about not spending more than this time.

Submitting solution

1. Use any of the following options to submit the solution:
 - a. Set up a private repository and provide us access (username: `eddie-ruva`). Once it is ready, please send an email to eddie@useascend.com.
 - b. Send us a compressed file with the solution to eddie@useascend.com
2. Include a readme describing how to run the project and at least one sample request for each one of the use cases.

Questions

Feel free to send any questions about the exercise to Eddie (eddie@useascend.com).