

Winter 2019 Developer Intern Challenge Question

Please complete the following questions, and provide your thought process/work. You can attach your work in a text file, link, etc. on the application page. Please ensure answers are easily visible for reviewers!

Question 1: Please design a server-side web API that models the following simple relationship:

Shops have many Products
Shops have many Orders
Products have many Line Items
Orders have many Line Items

Line items refer to any service or product added to an order, along with the quantity and price that pertain to them.

For example, if you buy a carton of milk and a loaf of bread at the grocery store, your bill (the representation of your order) will have two line items on it. One for the carton of milk, and the other for the loaf of bread.

Requirements for each object type

Products, Line Items and Orders all need a dollar value
The value of a Line Item should map to the value of the Product that the Line Item represents
The total value of an Order should equal the sum of the values of all of its Line Items

Demo requirements

All of the functionality of your API should be documented so we know what it does, and how to interact with it.

When using your API, there should be at least one shop, one product, one line item, and one order to query. Feel free to commit your data file, include a seed file to populate the db or find some other way to make sure that the app we'll be testing has data in it.

Extra credit (not required)

Bonus points for supporting full CRUD operations, extending the base functionality in interesting ways that make sense for an e-commerce platform, making your API (at least partly) [secure](#), writing documentation that doesn't suck, writing tests for your functionality, and/or building your API using [GraphQL](#).

Tips

While you can tackle this challenge using any language and platform that you want, you'll almost certainly find the work easier and more manageable with the aid of a web development framework.

Extra extra credit (not required)

At Shopify, our core infrastructure is running in a Kubernetes environment. As an extra bonus step, deploy the web API you have created to a Kubernetes environment that is publicly accessible.

Here are some resources that can help you achieve this goal:

- Sign up for [Google Cloud Platform Free Tier](#). This would allow you to deploy to Google Kubernetes Engine (GKE).
- Learn about [Docker](#). Docker images are required to deploy to GKE.
- Learn about [Kubernetes](#).
- Learn [how to use GKE](#). GKE is Google's hosted Kubernetes product.

There are other Kubernetes environments available such as [Azure Kubernetes Service](#) or [Amazon EKS](#). You are not limited to GKE.

This can seem like a daunting task if you're not familiar with Kubernetes. Don't fret, any attempts demonstrates your curiosity and wanting to learn. Include any partial solutions should the task be not complete.