

Stitch Labs PHP Engineering Test

1 Introduction

Welcome to the StitchLabs Software Engineer coding test. All work must be completed in a Git repository and uploaded to Github or Bitbucket. You have 24 hours to make the last commit to that repository.

This test is intentionally left a little unstructured to allow you to show us your creative problem solving skills.

You will need:

PHP, MySQL, REST, and a whole lot of cowbell.

Some optional potentially useful things:

Composer, Guzzle, a PHP framework (Laravel, Symfony, Silex, etc)

You may not use:

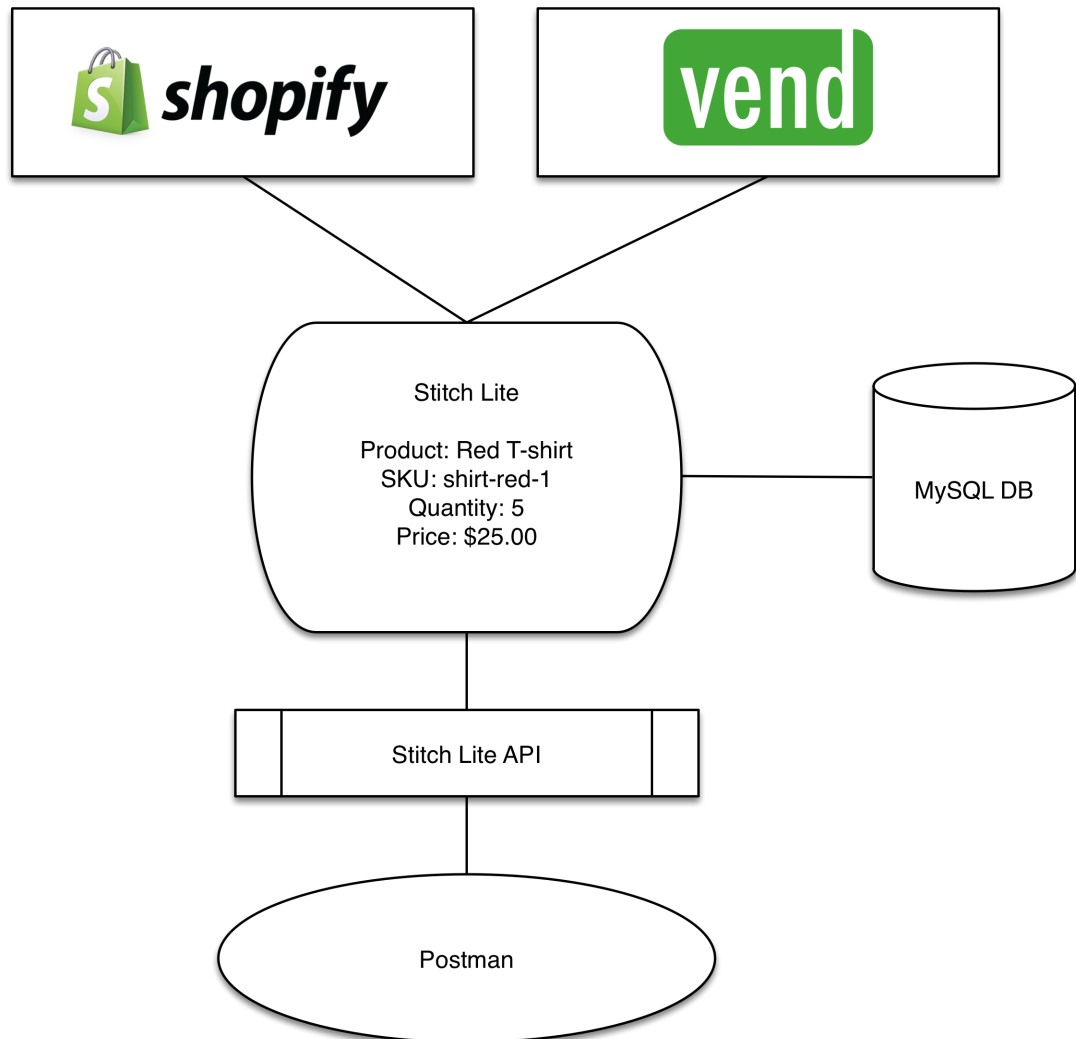
A non-MySQL database such as PostgreSQL, MongoDB, etc

A non-PHP language such as Ruby, Python, Java, etc

2 Stitch Lite

You will be writing an application called StitchLite. It will communicate with Shopify and Vend's respective API's to download product information. You will expose the functionality of StitchLite by creating a public API for

it. To test your API and show us that it works, you can use an application such as **Postman**. Please review the diagram below.



3 Integration Information

3.1 Connecting to Shopify

Create a Shopify account and product catalog:

1. Sign up for a [Shopify](#) free trial.
2. Login and manually create at least 3 products, each of which having 1 to 3 variants
3. Make sure all of the variants have SKU's
4. Connect to your Shopify store via their API utilizing the Private App authentication.

<http://docs.shopify.com/api/tutorials/creating-a-private-app>

3.2 Connecting to Vend

Create a Vend account and product catalog:

1. Sign up for a [Vend](#) free trial
2. Create two of the products that you made in Shopify, and then make at lease one different product. When this process is complete, you should have:
 - (a) One product on Shopify, with 1 to 3+ variants, that is unique to Shopify
 - (b) One product on Vend, with 1 to 3+ variants, that is unique to Vend
 - (c) At lease 2 products, each with 1 to 3+ variants, on both Shopify and Vend, where each have identical variants on both channels.
3. Make sure to add SKU's to your variants, making sure that the SKU's are the same as the ones you added in Shopify.
4. Connect to your Vend store via their API.

<https://developers.vendhq.com/documentation>.

Please note that this will require using OAuth2, as they have deprecated Basic Auth.

4 Stitch Lite Functionality

The purpose of your application is to track the products and their variants across the two sales channels (Vend and Shopify).

- Pull in products from Shopify and store them as Stitch Lite products
- Pull in products from Vend and store them as Stitch Lite products
- Be sure to pull in at least the following fields from the sales channels:
 - Product Name
 - SKU
 - Quantity
 - Price
- Products that have the same SKU on Shopify and Vend should be linked to the same Stitch Lite product
- Create a method of updating Stitch Lite products by syncing with the sales channels
 - This means that if you login to Vend and change a products name, sku, quantity, and/or price, by running a sync the Stitch Lite products name, sku, quantity, and/or price should change.
 - Please be prepared to discuss with us your architecture in regards to easily supporting the addition of more sales channels and integrations.

5 The Stitch Lite API

You will also expose the Stitch Lite application by creating an API. The API should expose the data within Stitch Lite that another client might care about and allow another client to trigger your sync functionality.

The following are our suggested endpoints:

- **POST** */api/sync* - Syncs all sales channels to pull in product updates to Stitch Lite.
- **GET** */api/products* - Returns all Stitch Lite products

- **GET** `/api/products/[productid]` - Returns the specified Stitch Lite product

Unless you feel strongly about XML, the response format should be JSON.

6 Useful Vocabulary

Product vs Variant: A product on most channels represents top level information like a description or name. Example: You have one product with the name: "Game of Thrones T-Shirt". A variant under this parent may have three *attributes* like Size, Color, or House. Some example variants could be: "Game of Thrones T-Shirt / XXL / Green / Targaryen", or "Game of Thrones T-Shirt / L / Gold / Lannister". The SKU, Price, and Quantity are usually tracked at a variant level.

SKU: Also known as Stock Keeping Unit. This is a free form field to identify an item that you sell. If you sell peanut butter, it's SKU could be "1234", "pb-no-jelly", or "i-hope-you-arent-allergic". It really doesn't matter as long as they are unique for each item that you sell. This is the field that you could use to link up listings across sales channels.

7 Wrap Up

Please keep in mind, this programming test is meant to help us see your ability to design and architect an application. If you get hung up on any of the steps listed here, please reach out to us immediately and we will be happy to clarify. Don't forget to ask questions!

If you want an even more difficult challenge, you could consider additionally syncing Stitch Lite product quantities **up** to the sales channels.

IMPORTANT! To be ready for your code review, please be prepared to answer questions regarding schema decisions, the normalization of product data, the linking of sales channel products to Stitch Lite products, and your architecture in regards to easily supporting the addition of more sales channels and integrations.