Laravel and Filament Test

Submition GIT repo

https://github.com/nmsan/webco-test

Demo URL: https://webco-test-main-08jreg.laravel.cloud/admin

Email: admin@example.com

Password: password

Description

The below specification provides a sample project that aims to test both your ability to work with the expected technologies and your resourcefulness in solving problems.

Note: While AI can assist, you will be asked to explain your use of anything within the code. Relying strictly on AI-based assistance is not recommended.

You should still submit your project even if you have not achieved all the goals. Some of the tasks are designed to be a little tricky.

Note: At any point, you are permitted to ask questions. They will be responded to as quickly as possible.

Topics

1. Setting up Laravel (using SQLite to keep things simple).
2. Installing and Configuring Filament.
3. Building complete, valid models using an ERD as reference.
4. Importing seeded data.
5. Installing a custom theme.
6. Changing the Theme CSS to override the sidebar background color using Tailwind CSS types.
7. Creating a basic CRUD to manage specific models.
8. Creating a complex CRUD for the main model (specification in the project).

9. Creating an 'infolist' for a read-only view.
10. Using a 'suffix Action' to fetch or validate a field using an external API.
11. Building a simple custom field (Create a status bar on the 'Product' model that says 'Hello' and the background color of the bar is mapped against the product).
12. Creating a simple job.
13. Creating an action on a model list that will use the 'simple job' and process the request.
14. Creating a 'loading' symbol when the state is changed in the required text field that appears in the suffix field. Perform an external integration within the action.
15. Finding a neat way to show the amount of models created (that are custom to the project) on the Dashboard.

Note: Additional fields may be added if required to support the outcome.

Technical Areas Covered

- PHP
- Filament
- Laravel Queues
- Laravel Artisan
- Eloquent (query, scopes, relationships)
- REST
- Token Authorization
- Livewire
- Tailwind CSS
- ERD / UML
- SQL

Sample

A sample of what is expected has been provided: Sample Project

User: project-test@projecttest.com.au
Pass: oxhyV9NzkZ^02MEB

Objective

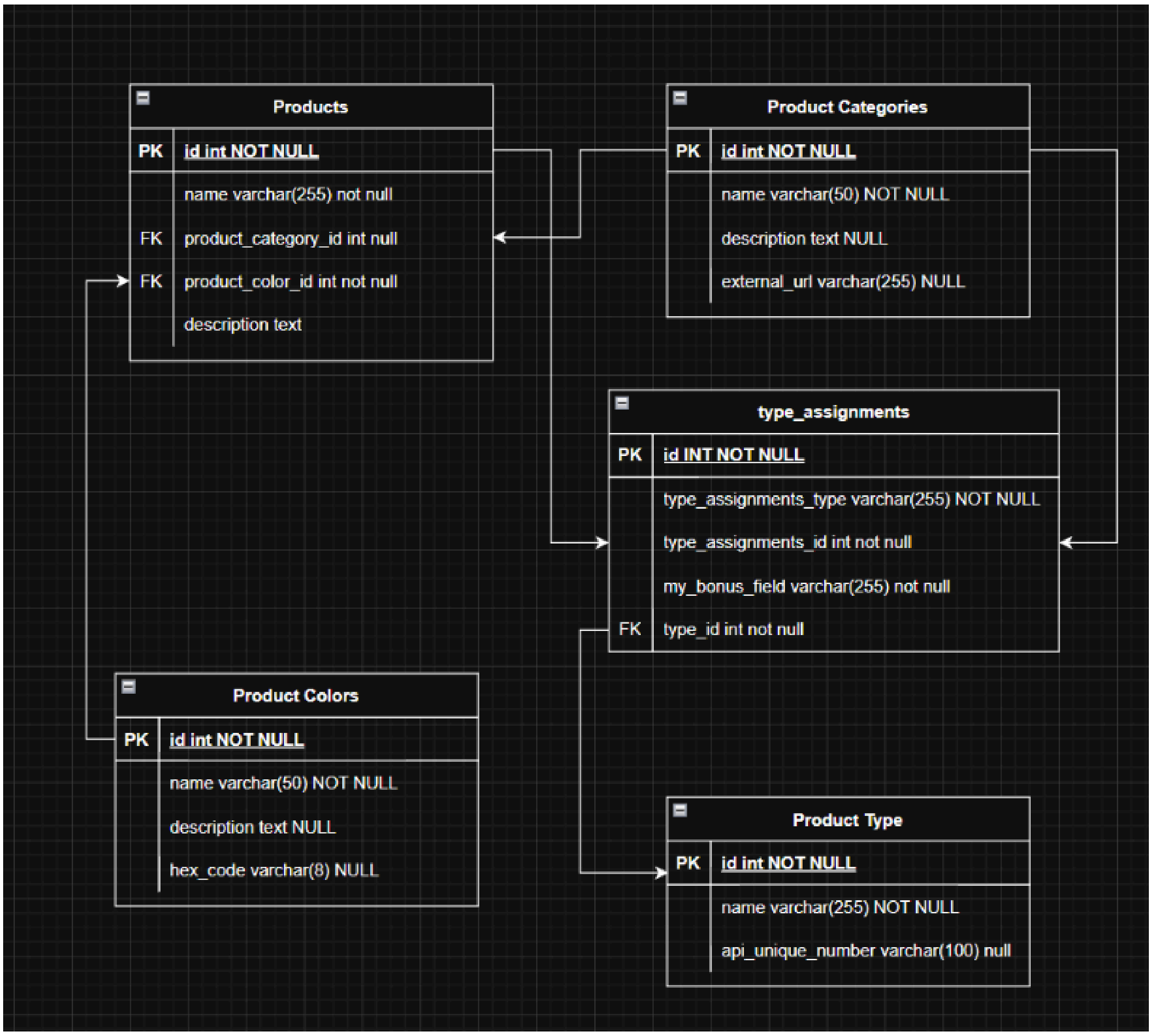
Attempt to achieve all goals as best as possible. If you choose to add more to your project to showcase additional skills, feel free to do so.

Review Metrics

Goal	Metric
Clean Code	Your code should be organised and written cleanly. Like all programming, it should be written in a way that is not for you, but rather the next person that will see the code to work on it.
Self-documenting	While your code should use PHP Doc standards, it should be clear on what it does. This will be a combination of naming conventions and adherence to best practices. The project is based on PHP 8.2+, so return types and defined parameters are part of the game.
Business functionality addressed	The business requirements set above and described in detail below should be completed as accurately as you can.
Adherence to supplied toolsets	Try to avoid going ‘low-level’ for solving problems. The test has been designed to be achievable using the software described.
Complete processes	Creating a model has multiple components. They should all be used

---

Database Model


Products
PK
id int NOT NULL
name varchar(255) not null
FK
product_category_id int null
FK
product_color_id int not null
description text
Product Categories
PK
id int NOT NULL
name varchar(50) NOT NULL
description text NULL
external_url varchar(255) NULL
type_assignments
PK
id INT NOT NULL
type_assignments_type varchar(255) NOT NULL
type_assignments_id int not null
my_bonus_field varchar(255) not null
FK
type_id int not null
Product Colors
PK
id int NOT NULL
name varchar(50) NOT NULL
description text NULL
hex_code varchar(8) NULL
Product Type
PK
id int NOT NULL
name varchar(255) NOT NULL
api_unique_number varchar(100) null

Tables & Descriptions

Table	Description

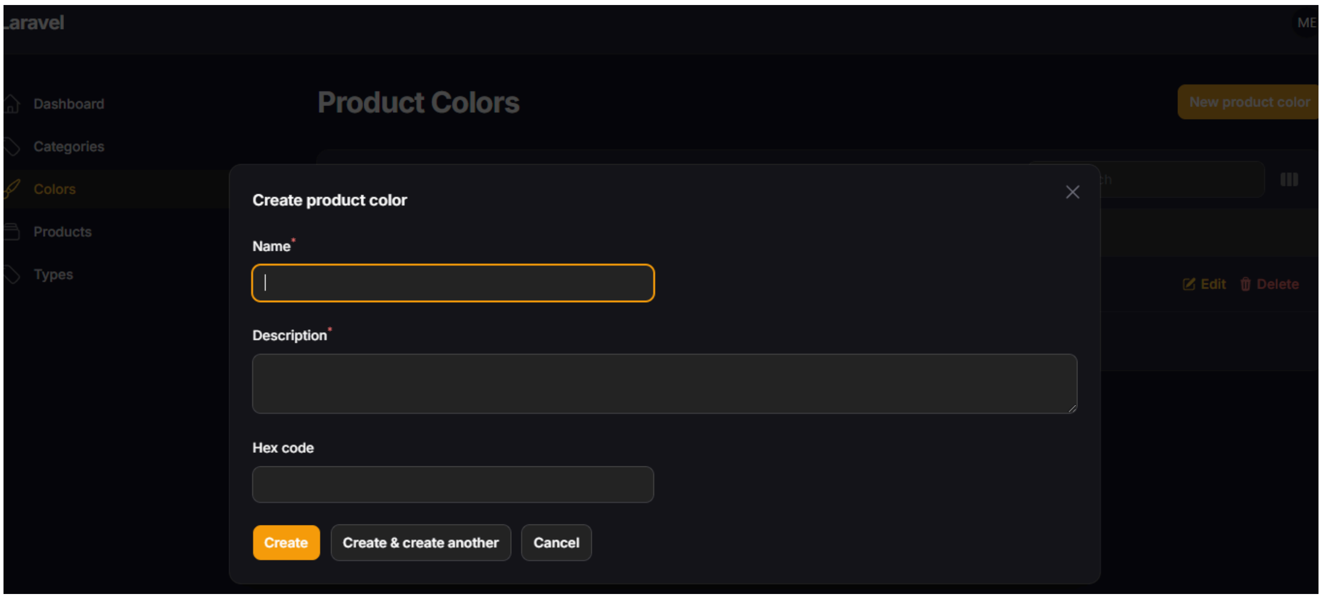
Product Colors	The Product Colors table is self-explanatory. It is a table full of HTML colours. Populate this table with web-safe colors, including their HEX code. The contents of this table should also be exported as a seeded file to enable it to be quickly rebuilt elsewhere.
Product Categories	A name, description and URL for a category. The product category is also bound to multiple product types using a 'morph Many to Many' relationship.
Products	The main products table has links to each of the reference tables. Similar to categories, it also has types bound to it. Business Rule A product can only have categories selected against it where the list of product types matches. You may need to consider how this will work.
Product Types	The types of products and categories. They can be assigned to multiple of each. The API Unique number is a number that should be fetched when editing or adding the product type. Use a suffix action.

User Interaction and Design

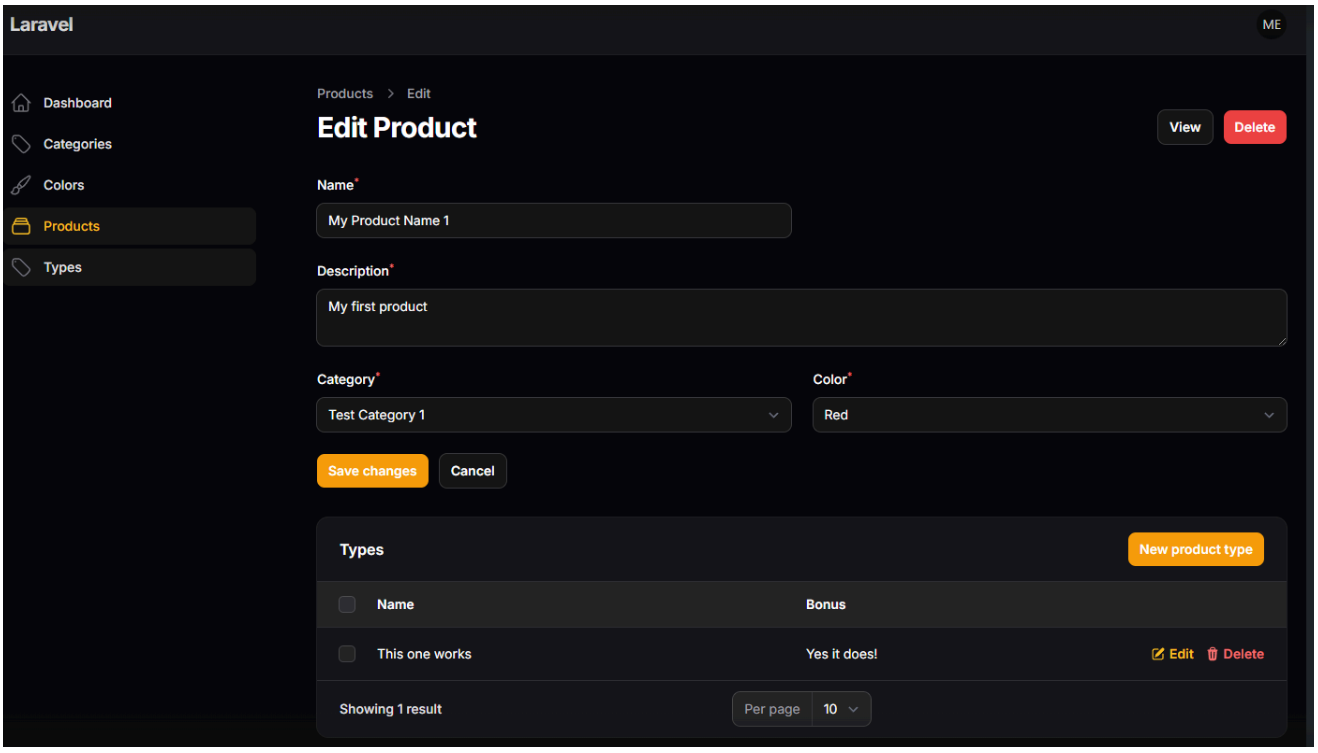
Further examples and information on the requirements

- Setting up Laravel. (I'm using SQLite to keep things simple.).
    - Tip: Read the documentation for Laravel 12 (note, it's a 'Livewire' project.
    - Tip: Use Laravel Cloud Sandbox. Add your code to Git. It will make it easier for me to review also.
- Install Filament
    - Tip: Read the guide
- Create seeded data for your sample of 'colors. '
- Install a custom theme.
    - Tip: Read the Filament website.
- Change the Theme CSS to override the sidebar background color using Tailwind CSS types.
    - Tip: Read the filament website and other guides

- Creating a basic CRUD to manage specific models.



- Creating a complex CRUD for the main model. (Specification in the project)



Edit Product

Name*: My Product Name 1

Description*: My first product

Category*: Test Category 1

Color*: Red

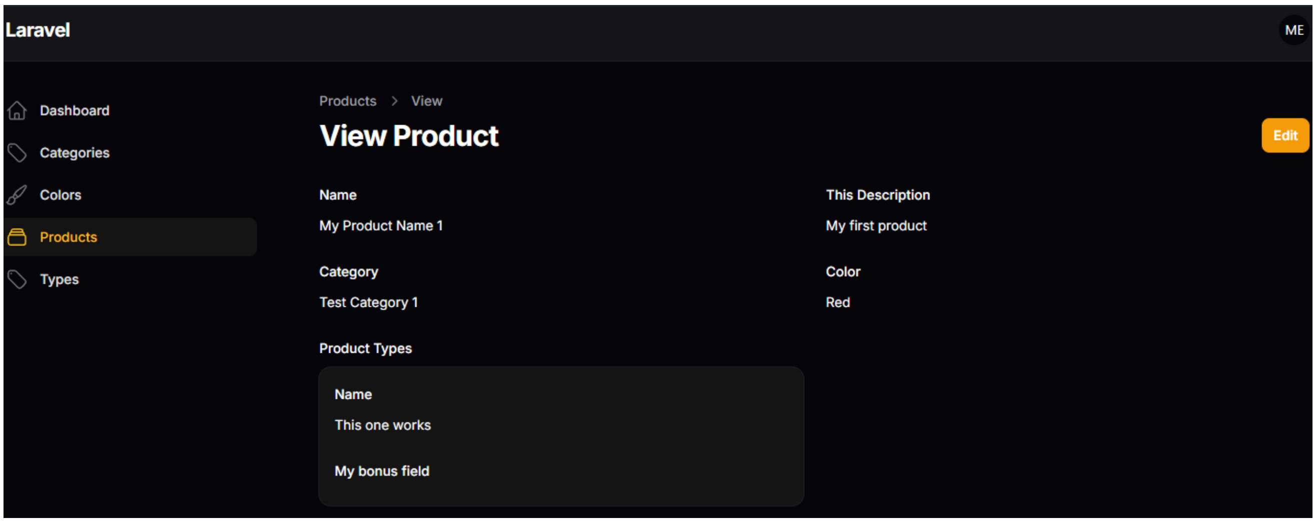
Types

Name	Bonus
This one works	Yes it does!

Showing 1 result

Per page: 10

- Creating an ‘infolist’ for a read-only view.



View Product

Name
My Product Name 1

This Description
My first product

Category
Test Category 1

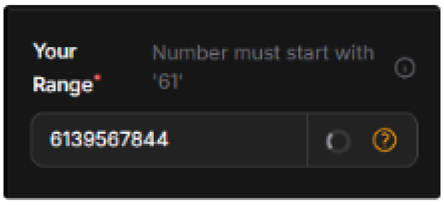
Color
Red

Product Types

Name
This one works

My bonus field

- Using a ‘suffix Action’ to fetch or validate a field using an external API - There is an endpoint described below you can use.
  - Tip: Try an address field and the API example below.
  - Tip: Use the portal link as an example
  - Tip: Search ‘getSearchResultsUsing as a 'Closure’ for inspiration.
- Building a simple custom field (Create a status bar on the ‘Product’ model that says ‘Hello’ and the bar's background colour is the color mapped against the product).
- Creating a simple job - Read the docs. The job should be executed from the user interface and modify the product it was executed from. This change should be visible in the UI - extra points for sending a database notification (read the docs).
  - This means an action on a model list that will use the ‘simple job’ and process the request.
  - Tip: Look into actions
- Create a ‘loading’ symbol when the state is changed in the required text field that appears in the suffix field. Perform an external integration within the action. Tip: Livewire hack



Loading Marker

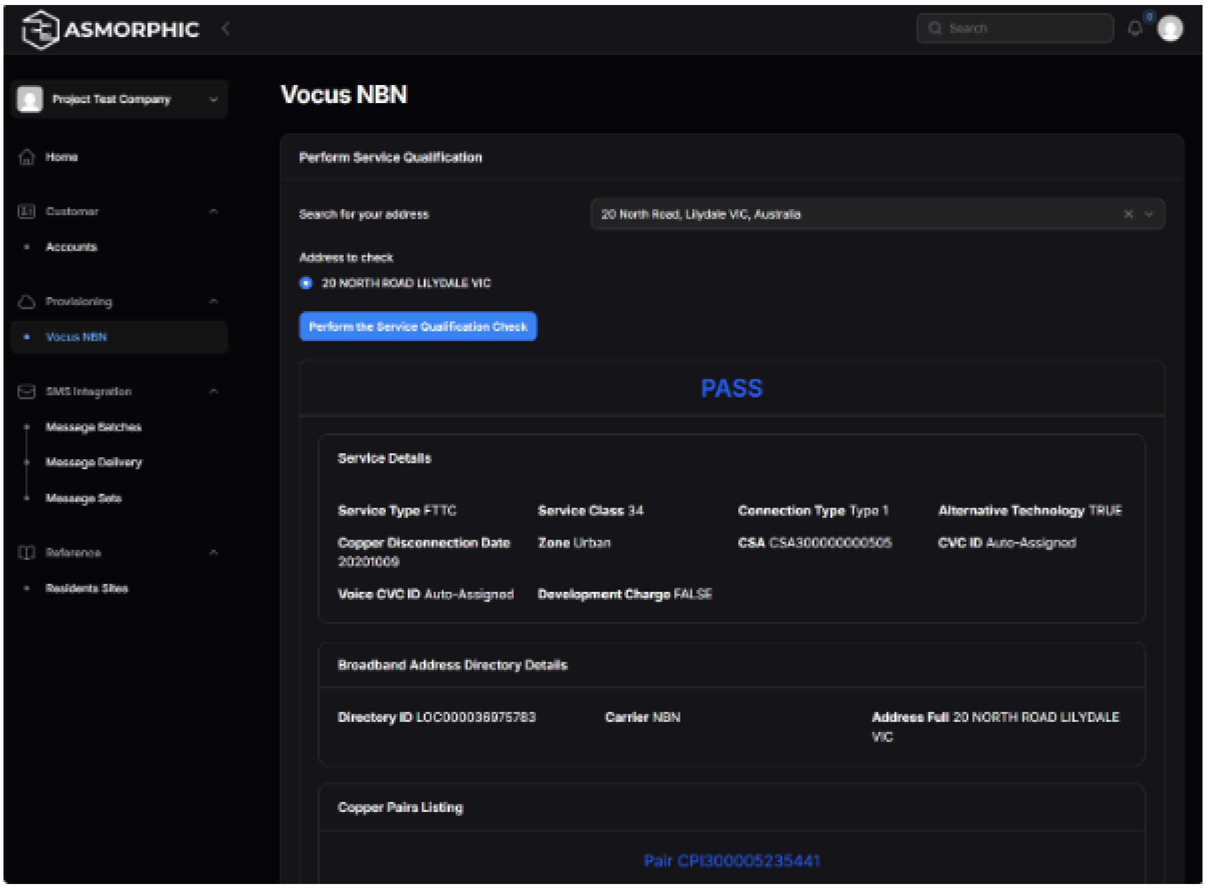
- Find a neat way to show the number of models created (custom to the project) on the Dashboard.

- Tip: There might be a plugin available somewhere.

Additional References

Address SQ Check

This is the API that can be used to qualify an address. You can log into https://extranet.asmorphic.com/portal/ using the below credentials and find an address using the 'Vocus SQ' section.



Vocus NBN

Perform Service Qualification

Search for your address: 20 North Road, Lilydale VIC, Australia

Address to check
● 20 NORTH ROAD LILYDALE VIC

PASS

Service Details

Service Type FTTC | **Service Class** 34 | **Connection Type** Type 1 | **Alternative Technology** TRUE
Copper Disconnection Date 20201009 | **Zone** Urban | **CSA** CSA300000000505 | **CVC ID** Auto-Assigned
Voice CVC ID Auto-Assigned | **Development Charge** FALSE

Broadband Address Directory Details

Directory ID LOC000038975783 | **Carrier** NBN | **Address Full** 20 NORTH ROAD LILYDALE VIC

Copper Pairs Listing

Pair CPI300005235441

SQ Helper Page

Note: A bug in the SQ page means you may have to hit 'process' again after it loads.

Username: project-test@projecttest.com.au

Password: oxhyV9NzkZ^02MEB

Login Via: https://extranet.asmorphic.com/api/login (post with username and password to retrieve a token)

    ["email" => $username, "password" => $password]

API End Point: https://extranet.asmorphic.com/ (use Bearer token in the header)

    ["Authorization: Bearer $this->bearerToken", "Content-Type: application/json", "Accept: application/json"]

FIND ADDRESS

Find Address: api/orders/findaddress

Method: POST

Parameters:

company_id = 17

street_number = nullable

street_name = not nullable

street_type = full street type like ‘Street’ – Matching Vocus

suburb = not nullable

postcode = not nullable

state = VIC/NSW/QLD/WA etc

Output

```
Array = [

[

 DirectoryIdentifier => LOCXXXX,

 Address => ‘ADDRESS STRING’

]
```

SERVICE QUALIFICATION

Qualify: api/orders/qualify

Method: POST

```
Parameters:

company_id = 17

qualification_identifier = LOCDIRECTORYID

service_type_id = 3

array (

 'result' => 'Success',

 'message' => 'Qualification Processed for LOC000092333312',

 'data' =>

 array (

   0 =>

   (object) array(

      'Result' => 'PASS',

      'ServiceType' => 'FTTC',

      'ServiceClass' => '34',

      'ConnectionType' => 'Type 1',

      'AlternativeTechnology' => 'TRUE',

      'CopperDisconnectionDate' => '20211112',

      'Zone' => 'Urban',

      'DevelopmentCharge' => 'FALSE',

      'CSA' => 'CSA300111111154',

      'CVCID' => 'Auto-Assigned',

      'CopperPairRecords' =>

     array (

       0 =>
'<CopperPairRecord><CopperPairID>CPI300012170843</CopperPairID><CopperPairStatus>N/A</Coppe
rPairStatus><NBNServiceStatus>Line In
Use</NBNServiceStatus><ServiceClass>34</ServiceClass><POTSInterconnect>N/A</POTSInterconnect><
POTSMatch>FALSE</POTSMatch><TC2Speed>5Mbps,10Mbps,20Mbps</TC2Speed><UploadSpeed>38-40
</UploadSpeed><DownloadSpeed>95-100</DownloadSpeed><NetworkCoExist>TRUE</NetworkCoExist>
<ExtraCharge>FALSE</ExtraCharge></CopperPairRecord>',

       1 =>
'<CopperPairRecord><CopperPairID>CPI300009457769</CopperPairID><CopperPairStatus>Inactive</Co
pperPairStatus><NBNServiceStatus>N/A</NBNServiceStatus><ServiceClass>32</ServiceClass><POTSInte
rconnect>N/A</POTSInterconnect><POTSMatch>FALSE</POTSMatch><TC2Speed>5Mbps,10Mbps,20Mb
ps</TC2Speed><UploadSpeed>20-40</UploadSpeed><DownloadSpeed>50-100</DownloadSpeed><Net
workCoExist>TRUE</NetworkCoExist><ExtraCharge>TRUE</ExtraCharge></CopperPairRecord>',

     ),

      'BroadbandAddressRecord' =>
'<BroadbandAddressRecord><DirectoryID>LOC000092206412</DirectoryID><Carrier>NBN</Carrier><Ad
dressLong>YOUR ADDRESS STREET VIC</AddressLong></BroadbandAddressRecord>',

      'NBNCOATRecord' =>
'<NBNCOATRecord><ForecastAccessTechnology>Fibre</ForecastAccessTechnology><RTCDate>2022333
3330000</RTCDate><Reason>on-demand</Reason></NBNCOATRecord>',

   ),

 ),

)
```