

Full Stack Software Engineer

HR: Please send this questionnaire to the candidate 2/3 days before the interviewer and interviewee can discuss the solutions in a conference call. If working candidates request it to be sent on Friday to discuss solutions on immediately following Monday/Tuesday, so that they can work on weekend. Any exceptions requested by the candidate with sufficient reasoning is acceptable (for e.g. if he/she requests more than 3 days)

General: The solutions to the below questions will be discussed on a conference call through a desktop sharing session. The interviewee will display the execution of code on his/her machine during the conference call.

Interviewer: The interviewer may choose to ask other similar questions during the interview process

Interviewee: The following exercises are intended to help us get a feel for your coding habits and strategies. You may search online and find solutions to the below. For each of the questions below, check-in your code to Github (where relevant) and share the link with us before the interview. You will have to share your desktop while explaining the solution

The exercise will be given to the candidate couple of days before we discuss solutions. At a high level, below is the break-up of the exercise:

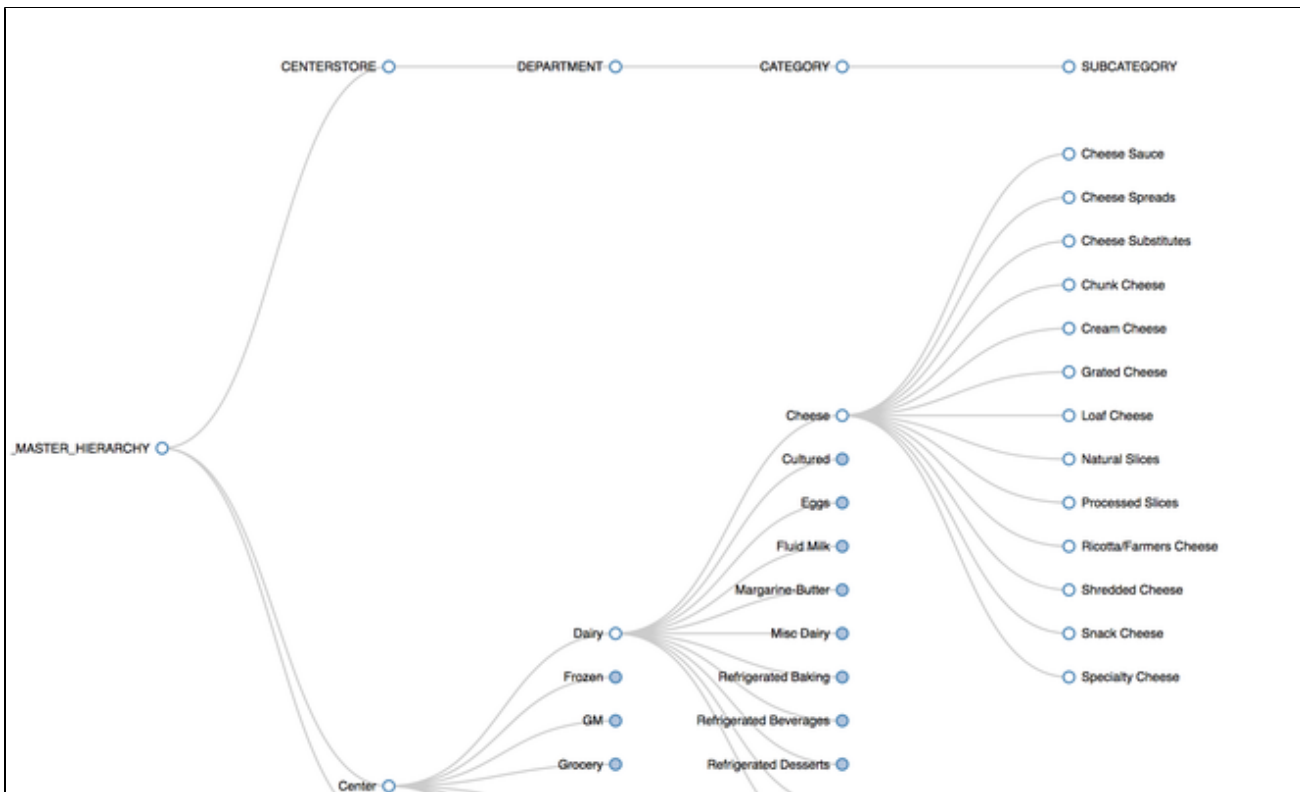
1. Demonstrate your hands-on full stack experience with Exercise 1 (BUILD)
 2. Demonstrate your expertise with Test-driven development practices and CI/CD concepts with Exercise 2 (SHIP)
 3. Demonstrate your skills to maintain, support and address production issues with Exercise 3 (RUN)
-

Exercise1 (BUILD):

The hierarchical meta-data below helps uniquely identifying a SKU in a retail store.

Hierarchical data because Department is contained within location, Category is contained within Department and so on.

1. Develop a REST API for representing Meta-Data with end points (GET, POST, PUT, DELETE etc.). For example GET on `/api/v1/location` returns all location objects
 - a. `/api/v1/location`,
 - b. `/api/v1/location/{location_id}/department`
 - c. `/api/v1/location/{location_id}/department/{department_id}/category`
 - d. `/api/v1/location/{location_id}/department/{department_id}/category/{category_id}/subcategory`
 - e. `/api/v1/location/{location_id}/department/{department_id}/category/{category_id}/subcategory/{subcategory_id}`
2. Persist the data in your favorite db - relational or non-relational (You are expected to install, configure, populate the db). You may feel free to qualify object representations with additional attributes to enhance modeling (For e.g. Location object attributes = locationid, location description)
3. In your chosen web front-end framework, develop a web-based UI, that the user can interact with and perform CRUD (create, read, update, delete) operations on the data (The calls should go through API layer built in step 1). Bonus points, if you can develop an interactive tree showing the Meta-Data as depicted in this [infographic](#).



Meta-Data

Hierarchical MetaData

Location,Department,Category,SubCategory
 Perimeter,Bakery,Bakery Bread,Bagels
 Perimeter,Bakery,Bakery Bread,Baking or Breeding Products
 Perimeter,Bakery,Bakery Bread,English Muffins or Biscuits
 Perimeter,Bakery,Bakery Bread,Flatbreads
 Perimeter,Bakery,In Store Bakery,Breakfast Cake or Sweet Roll
 Perimeter,Bakery,In Store Bakery,Cakes
 Perimeter,Bakery,In Store Bakery,Pies
 Perimeter,Bakery,In Store Bakery,Seasonal
 Center,Dairy,Cheese,Cheese Sauce
 Center,Dairy,Cheese,Specialty Cheese
 Center,Dairy,Cream or Creamer,Dairy Alternative Creamer
 Center,Dairy,Cream or Creamer,Whipping Creams
 Center,Dairy,Cultured,Cottage Cheese
 Center,Dairy,Refrigerated Baking,Refrigerated Breads
 Center,Dairy,Refrigerated Baking,Refrigerated English Muffins and Biscuits
 Center,Dairy,Refrigerated Baking,Refrigerated Hand Held Sweets
 Center,Dairy,Refrigerated Baking,Refrigerated Pie Crust
 Center,Dairy,Refrigerated Baking,Refrigerated Sweet Breakfast Baked Goods
 Perimeter,Deli and Foodservice,Self Service Deli Cold,Beverages
 Perimeter,Deli and Foodservice,Service Deli,Cheese All Other

Perimeter,Deli and Foodservice,Service Deli,Cheese American
Perimeter,Floral,Bouquets and Cut Flowers,Bouquets and Cut Flowers
Perimeter,Floral,Gifts,Gifts
Perimeter,Floral,Plants,Plants
Center,Frozen,Frozen Bake,Bread or Dough Products Frozen
Center,Frozen,Frozen Bake,Breakfast Cake or Sweet Roll Frozen
Center,Frozen,Frozen Breakfast,Frozen Breakfast Entrees
Center,Frozen,Frozen Breakfast,Frozen Breakfast Sandwich
Center,Frozen,Frozen Breakfast,Frozen Egg Substitutes
Center,Frozen,Frozen Breakfast,Frozen Syrup Carriers
Center,Frozen,Frozen Desserts or Fruit and Toppings,Pies Frozen
Center,Frozen,Frozen Juice,Frozen Apple Juice
Center,Frozen,Frozen Juice,Frozen Fruit Drink Mixers
Center,Frozen,Frozen Juice,Frozen Fruit Juice All Other
Center,GM,Audio Video,Audio
Center,GM,Audio Video,Video DVD
Center,GM,Audio Video,Video VHS
Center,GM,Housewares,Bedding
Center,GM,Housewares,Candles
Center,GM,Housewares,Collectibles and Gifts
Center,GM,Housewares,Flashlights
Center,GM,Housewares,Frames
Center,GM,Insect and Rodent,Indoor Repellants or Traps
Center,GM,Insect and Rodent,Outdoor Repellants or Traps
Center,GM,Kitchen Accessories,Kitchen Accessories
Center,GM,Laundry,Bleach Liquid
Center,GM,Laundry,Bleach Powder
Center,GM,Laundry,Fabric Softener Liquid
Center,GM,Laundry,Fabric Softener Sheets
Center,Grocery,Baking Ingredients,Dry or Canned Milk
Center,Grocery,Baking Ingredients,Food Coloring
Center,Grocery,Spices,Salt Cooking or Edible or Seasoned
Center,Grocery,Spices,Salt Substitute
Center,Grocery,Spices,Seasoning Dry
Center,Grocery,Stuffing Products,Stuffing Products
Perimeter,Seafood,Frozen Shellfish,Frozen Shellfish
Perimeter,Seafood,Other Seafood,All Other Seafood
Perimeter,Seafood,Other Seafood,Prepared Seafood Entrees

```
Perimeter,Seafood,Other Seafood,Seafood Salads
Perimeter,Seafood,Other Seafood,Smoked Fish
Perimeter,Seafood,Other Seafood,Seafood Breeding Sauces Dips
```

Data

```
SKU,NAME,LOCATION,DEPARTMENT,CATEGORY,SUBCATEGORY
1,SKUDESC1,Permitter,Bakery,Bakery Bread,Bagels
2,SKUDESC2,Permitter,Deli and Foodservice,Self Service Deli
Cold,Beverages
3,SKUDESC3,Permitter,Floral,Bouquets and Cut Flowers,Bouquets and
Cut Flowers
4,SKUDESC4,Permitter,Deli and Foodservice,Service Deli,All Other
5,SKUDESC5,Center,Frozen,Frozen Bake,Bread or Dough Products Frozen
6,SKUDESC6,Center,Grocery,Crackers,Rice Cakes
7,SKUDESC7,Center,GM,Audio Video,Audio
8,SKUDESC8,Center,GM,Audio Video,Video DVD
9,SKUDESC9,Permitter,GM,Housewares,Beeding
10,SKUDESC10,Permitter,Seafood,Frozen Shellfish,Frozen Shellfish
11,SKUDESC11,Permitter,Seafood,Other Seafood,All Other Seafood
12,SKUDESC12,Permitter,Seafood,Other Seafood,Prepared Seafood
Entrees
13,SKUDESC13,Permitter,Seafood,Other Seafood,Salads
14,SKUDESC14,Permitter,Bakery,Bakery Bread,Bagels
15,SKUDESC15,Permitter,Deli and Foodservice,Self Service Deli
Cold,Beverages
16,SKUDESC16,Permitter,Floral,Bouquets and Cut Flowers,Bouquets and
Cut Flowers
17,SKUDESC17,Permitter,Deli and Foodservice,Service Deli,All Other
18,SKUDESC18,Center,Frozen,Frozen Bake,Bread or Dough Products
Frozen
.....
FULL DATA will be provided as attachment when this coding test is
given to the candidate
```

4. Let's say you get the above "Data" that contains SKU's and its meta-data. Write a program/api end point that takes input meta-data and returns all the SKU rows in the "Data" that matches with the input meta-data.

For example, for input meta-data (Perimeter, Bakery, Bakery Bread, Bagels) , return the rows with SKUs 1 & 14 (The actual data given to you might contain more rows)

Exercise2 (SHIP):

1. Demonstrate test-driven development practices through code written in Exercise1 i.e. unit/component/integration/acceptance tests that can be run for a change in source code
2. Build and Deploy your application outside of your IDE with automated scripts (Imagine someone who has access to source code and has to build and deploy your application without talking to you - and doesn't have all the customized settings of your IDE). If you cannot write automated script, demonstrate the steps you would take , show us manually how you would deploy

Exercise3 (RUN):

1. Once the app (front-end, API/middle tier, back-end) that you built in Exercise1 and Exercise2 is running, implement a logging framework and make changes to the source code above.
2. Implement an authentication mechanism for the UI (username/password) and API (Basic Auth) - Bonus points if you can demonstrate OpenIDConnect, JWT, OAuth2
3. How would you check the HEALTH of the application by being proactive than reactive ? Feel free to use monitoring tools/frameworks available out-of-box (you are expected to set them up yourself)