**Network Service**

**Given User story**

As an application developer, I want a Web API which returns the number of “downstream” customers from a selected node in a network of nodes and branches, so that I can build features related to electrical current.

**Assumptions :**

Request data is in order,

Start and End node must not be same

Once root is assigned, no node must have an end node same as root node in the same network,

If selected node not exists simply return TotalNumberOfCustomers = 0.

**Solution**

Looking at the user story, I adopted an idea of network tree structure.

Network is made of Interconnected nodes

Node object contains data and one or more end nodes.

In adjacent figure, Node 10 connected to Node 20 and Node 30

We represent Node 10 as start node and Node 20 and Node 30 are end nodes

Here is my C# Node object

public class Node

{

public int Value { get; set; }

public List<Node> EndNodes { get; } = new List<Node>();

public Node(int value)

{

Value = value;

}

}

When a Node contains “EndNode”, that represents Node is connected to another node (end node); Similarly we connect other nodes to form a network.

Above figure could be written as below

var node = new Node(10);

To connect to another end nodes we add end nodes.

node.EndNodes.Add(20);

node.EndNodes.Add(30);

Connected nodes will form a Network.

public class Network

{

public Node? Root { get; set; }

List<int> nodes;

public Network()

{

nodes = new List<int>();

}

public void AddBranch(int StartNodeValue, int EndNodeValue)

{

…

}

public Node? FindNode(int value)

{

…

}

public Node? FindNode(Node node, int value)

{

…

}

public List<int> TraverseDownStream(int selectedNodeValue)

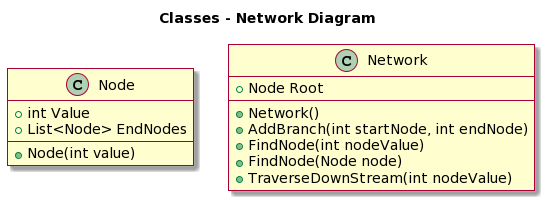
{

…

}

}

Network has a Root node and contains few methods to add branch, Find node in a network, traverse through the network from a given node.



Here we have a Json request.

Json Request

{

"network": {

"branches": [

{"startNode": 10,"endNode": 20},

{"startNode": 20,"endNode": 30},

{"startNode": 30,"endNode": 50},

{"startNode": 50,"endNode": 60},

{"startNode": 50,"endNode": 90},

{"startNode": 60,"endNode": 40},

{"startNode": 70,"endNode": 80}

],

"customers": [

{"node": 30,"numberOfCustomers": 8},

{"node": 40,"numberOfCustomers": 3},

{"node": 60,"numberOfCustomers": 2},

{"node": 70,"numberOfCustomers": 1},

{"node": 80,"numberOfCustomers": 3},

{"node": 90,"numberOfCustomers": 5}

]

},

"selectedNode": 50

}

We have “Branches” - an array of start and end nodes, “Customers” - an array of nodes with number of customers data and “selected node”.

**Choice of approach**

1. Validate and Create network using “branches“ request data.
2. Based on data we could get one or more networks, Here we got two networks

Once the network is created, based on the selected node, traverse through the network and return all nodes.

For example selected node is 50

Traverse from 50 to end and return 50, 60, 90, 40.

1. Once downstream nodes are received, select and sum customers data that contains downstream nodes.
2. For 50, 60,90,40 , we have

"customers": [

{"node": 30,"numberOfCustomers": 8},

{"node": 40,"numberOfCustomers": 3},

{"node": 60,"numberOfCustomers": 2},

{"node": 70,"numberOfCustomers": 1},

{"node": 80,"numberOfCustomers": 3},

{"node": 90,"numberOfCustomers": 5}

]

3+2+5 = 10

1. Since it is small tree, we can afford to get nodes and then compare with “customers” data. **Note**: We can also store number of customers in the node itself

Please find small solution attached.

Solution contains 3 projects

1. API
2. Core
3. Test

API is created using .NET8. Minimal API.

Core contains “NetworkProvider” which provides features to create network and nodes.

Test application contains few unit tests.

**URL**

**Post http://localhost:5207/api/NumberOfCustomers**

**Execute application**

Run and Execute API,

We can test using **Swagger** and **http** file

**Using Swagger**

When application runs, a new browser window opens with swagger open api.

Click “Try it out”, copy paste json string and execute.

A screenshot of a computer

Description automatically generated

**Using Http file**

Run the application and use http file in the project, by clicking “send request”.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Json response**

Json response returns TotalNumberOfCustomers downstream from the selected node.

{

"selectedNode": 50,

"totalNumberOfCustomers": 10

}