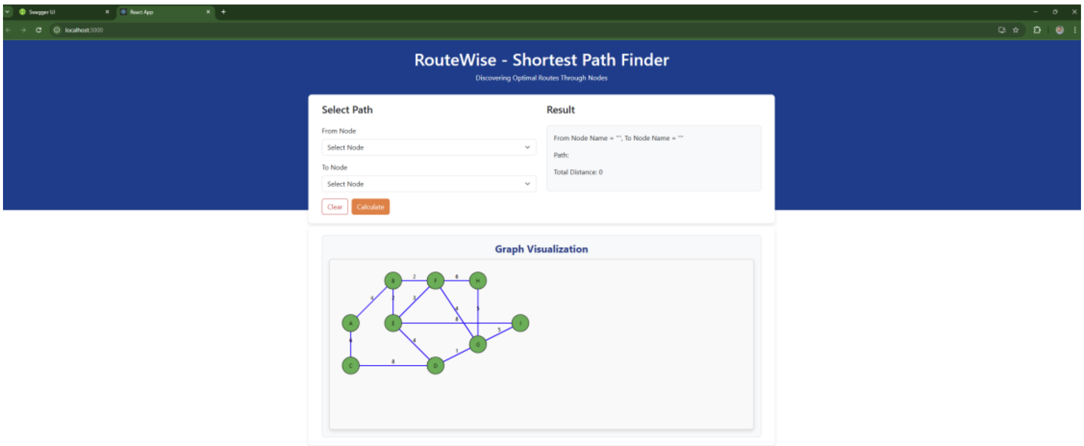
<u>RoutWise App – Developer Note</u>

This app is used to find the shortest path from node graph with two nodes. There are 3 apps,

1. API app – Calculate shortest path from the provided nodes or predefine nodes
2. UI app – To facilitate user to do the finding node paths and see the result
3. Console app – To fid the shortest path using console app without a proper UI

UI APP

App loading screen,



RouteWise - Shortest Path Finder

Discovering Optimal Routes Through Nodes

Select Path

From Node: Select Node

To Node: Select Node

Result

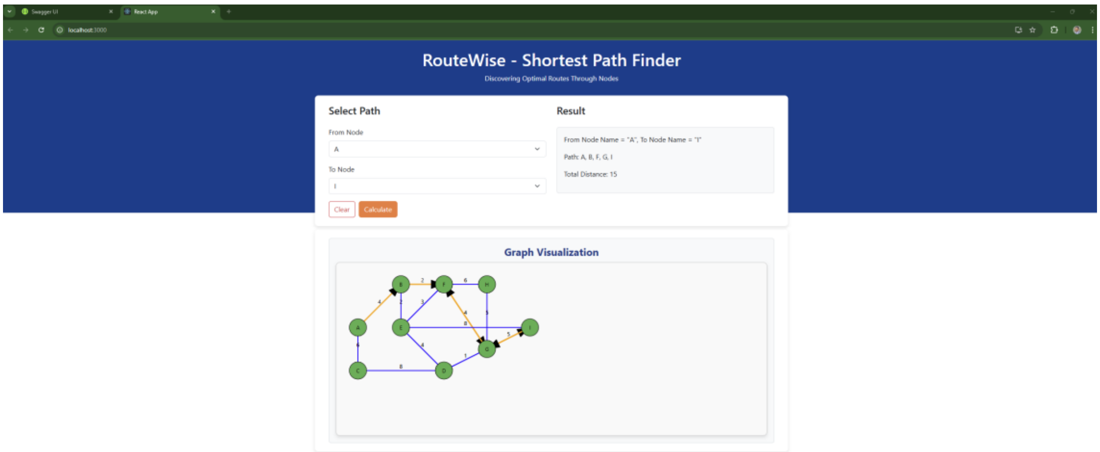
From Node Name = "", To Node Name = ""

Path:

Total Distance: 0

Graph Visualization

Users can select the start node and end node from the dropdowns, then click the calculate button and see the result with the path highlighter as below,



RouteWise - Shortest Path Finder

Discovering Optimal Routes Through Nodes

Select Path

From Node: A

To Node: I

Result

From Node Name = "A", To Node Name = "I"

Path: A, B, F, G, I

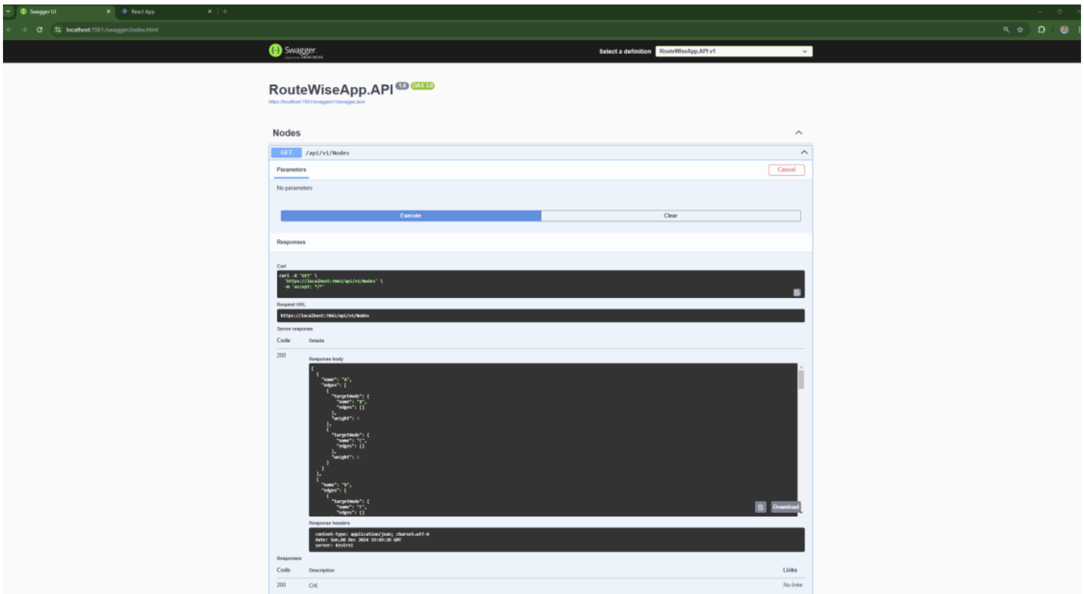
Total Distance: 15

Graph Visualization

API APP

API app has two endpoints to get graph nodes and find the short path, the node graphs store nodes in config JSON. If user does not provide node graph the default one will be loaded from configs – Mocking.

Get graph nodes



No parameters

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7061/api/v1/Nodes' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7061/api/v1/Nodes
```

Server response

Code Details

200

Response body

```
[
  {
    "name": "A",
    "edges": [
      {
        "targetNode": {
          "name": "B",
          "edges": []
        },
        "weight": 4
      },
      {
        "targetNode": {
          "name": "C",
          "edges": []
        },
        "weight": 6
      }
    ]
  },
  {
    "name": "B",
    "edges": [
      {
        "targetNode": {
          "name": "F",
          "edges": []
```

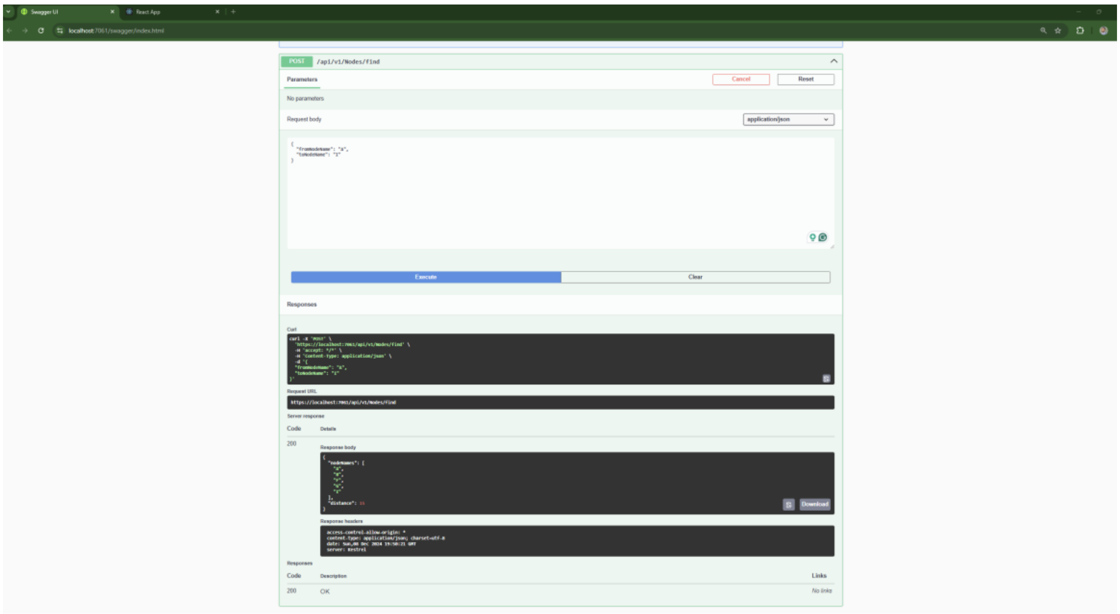
Response headers

```
content-type: application/json; charset=utf-8
date: Sun,08 Dec 2024 19:49:28 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

Find a shortest path – Without providing graph nodes



No parameters

Request body

```
{
  "fromNodeName": "A",
  "toNodeName": "I"
}
```

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7061/api/v1/Nodes/find' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "fromNodeName": "A",
  "toNodeName": "I"
}'
```

Request URL

```
https://localhost:7061/api/v1/Nodes/find
```

Server response

Code Details

200

Response body

```
{
  "nodeNames": [
    "A",
    "B",
    "F",
    "G",
    "I"
  ],
  "distance": 15
}
```

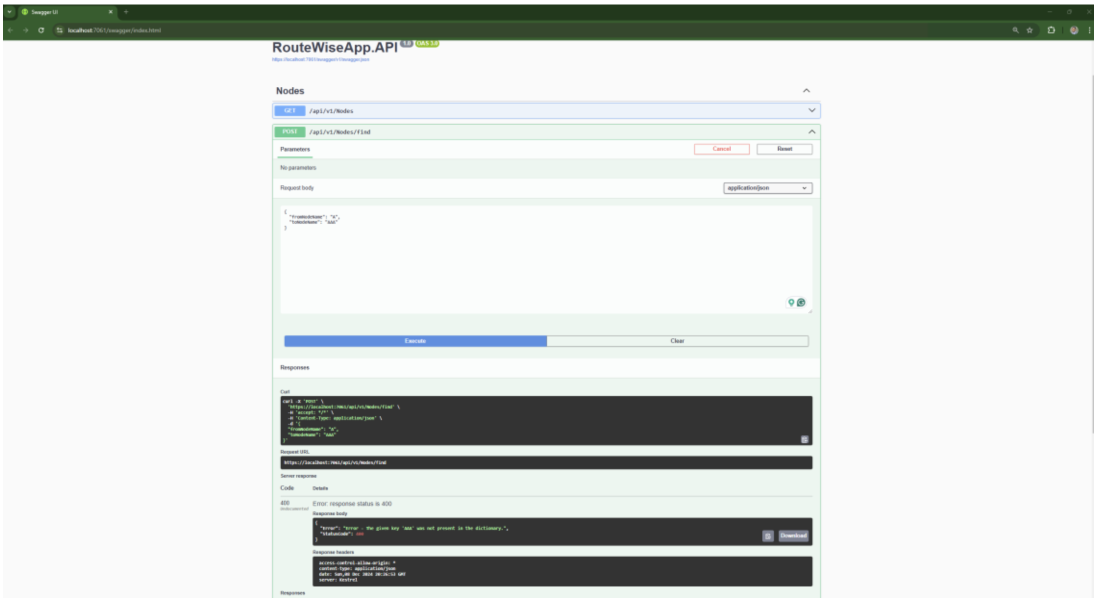
Response headers

```
access-control-allow-origin: *
content-type: application/json; charset=utf-8
date: Sun,08 Dec 2024 19:50:21 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	OK	No links

Find a path with an invalid node,



```
{
  "fromNodeName": "A",
  "toNodeName": "AAA"
}
```

Curl

```
curl -X 'POST' \
  'https://localhost:7061/api/v1/Nodes/find' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "fromNodeName": "A",
  "toNodeName": "AAA"
}'
```

Request URL

```
https://localhost:7061/api/v1/Nodes/find
```

400 Undocumented — Error: response status is 400

Response body

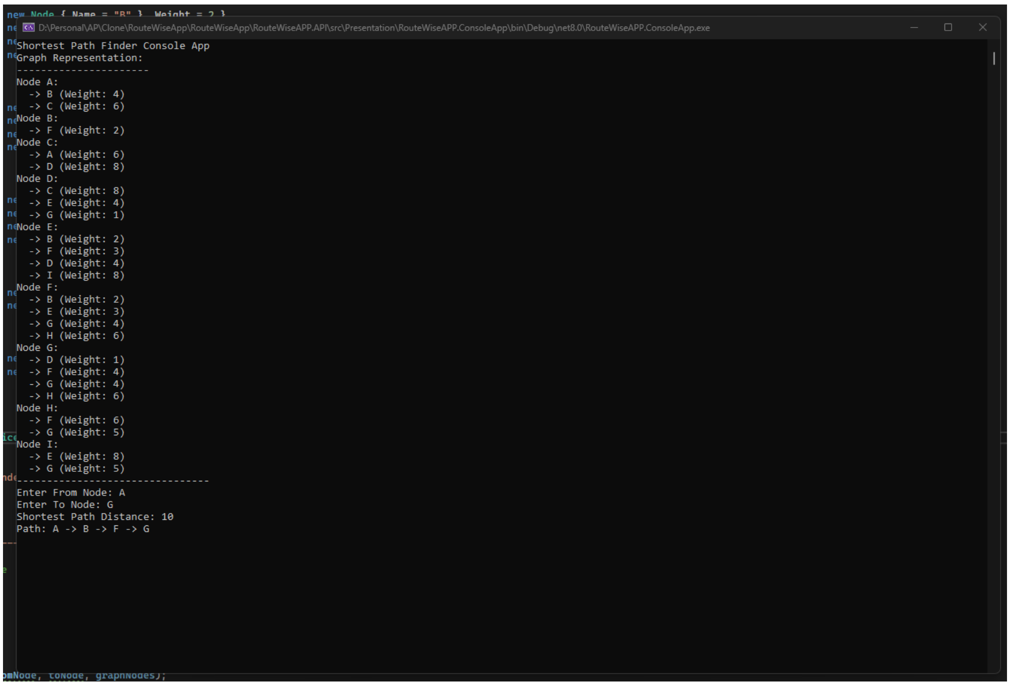
```
{
  "error": "Error : The given key 'AAA' was not present in the dictionary.",
  "statusCode": 400
}
```

Response headers

```
access-control-allow-origin: *
content-type: application/json
date: Sun,08 Dec 2024 20:26:53 GMT
server: Kestrel
```

CONSOLE APP

This app provides an easy way to find the path,



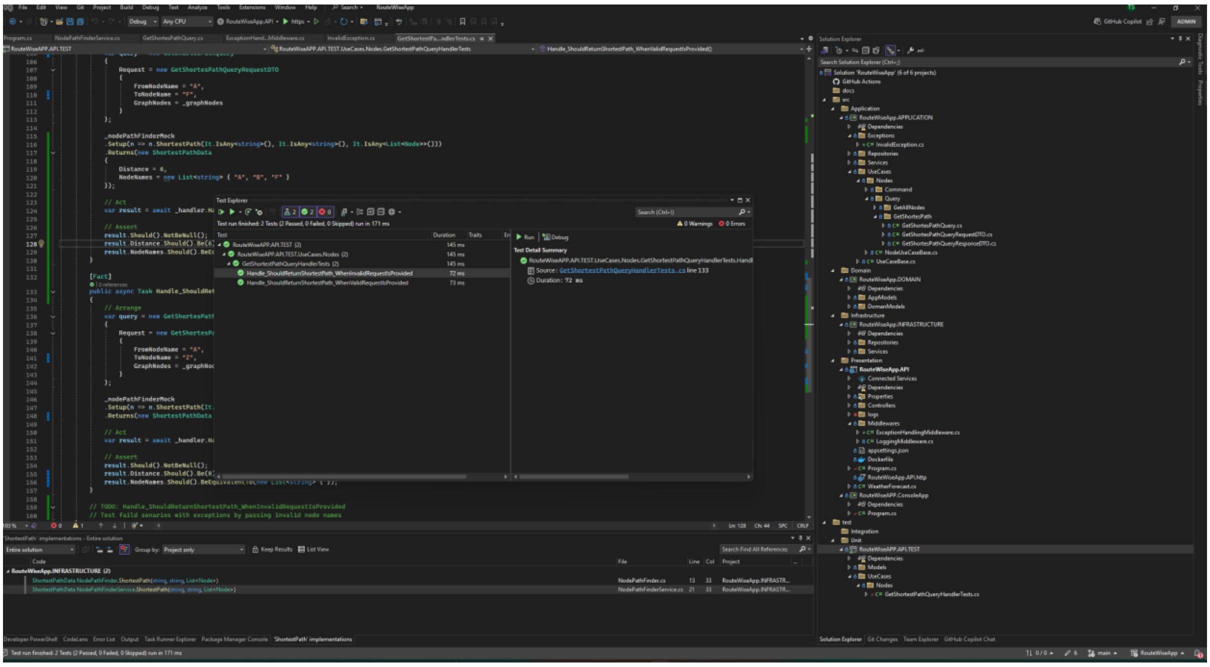
D:\Personal\AP\Clone\RouteWiseApp\RouteWiseApp\RouteWiseAPP.API\src\Presentation\RouteWiseAPP.ConsoleApp\bin\Debug\net8.0\RouteWiseAPP.ConsoleApp.exe

```
Shortest Path Finder Console App
Graph Representation:
----------------------
Node A:
  -> B (Weight: 4)
  -> C (Weight: 6)
Node B:
  -> F (Weight: 2)
Node C:
  -> A (Weight: 6)
  -> D (Weight: 8)
Node D:
  -> C (Weight: 8)
  -> E (Weight: 4)
  -> G (Weight: 1)
Node E:
  -> B (Weight: 2)
  -> F (Weight: 3)
  -> D (Weight: 4)
  -> I (Weight: 8)
Node F:
  -> B (Weight: 2)
  -> E (Weight: 3)
  -> G (Weight: 4)
  -> H (Weight: 6)
Node G:
  -> D (Weight: 1)
  -> F (Weight: 4)
  -> G (Weight: 4)
  -> H (Weight: 6)
Node H:
  -> F (Weight: 6)
  -> G (Weight: 5)
Node I:
  -> E (Weight: 8)
  -> G (Weight: 5)
--------------------------------
Enter From Node: A
Enter To Node: G
Shortest Path Distance: 10
Path: A -> B -> F -> G
```

TESTER APP

This shows the unit testing result for find path logic,

1. Should Return ShortestPath When Valid Request Is Provided
2. Should Return ShortestPath When Invalid Request Is Provided



```
        {
            Request = new GetShortesPathQueryRequestDTO
            {
                FromNodeName = "A",
                ToNodeName = "F",
                GraphNodes = _graphNodes
            }
        };

        _nodePathFinderMock
        .Setup(n => n.ShortestPath(It.IsAny<string>(), It.IsAny<string>(), It.IsAny<List<Node>>()))
        .Returns(new ShortestPathData
        {
            Distance = 6,
            NodeNames = new List<string> { "A", "B", "F" }
        });

        // Act
        var result = await _handler.Ha

        // Assert
        result.Should().NotBeNull();
        result.Distance.Should().Be(6)
        result.NodeNames.Should().BeE
    }

    [Fact]
    | 0 references
    public async Task Handle_ShouldRet
    {
        // Arrange
        var query = new GetShortesPath
        {
            Request = new GetShortesPa
            {
                FromNodeName = "A",
                ToNodeName = "Z",
                GraphNodes = _graphNod
            }
        };

        _nodePathFinderMock
        .Setup(n => n.ShortestPath(It.
        .Returns(new ShortestPathData

        // Act
        var result = await _handler.Ha

        // Assert
        result.Should().NotBeNull();
        result.Distance.Should().Be(0
        result.NodeNames.Should().BeEquivalentTo(new List<string> { });
    }

    // TODO: Handle_ShouldReturnShortestPath_WhenInvalidRequestIsProvided
    // Test faild senarios with exceptions by passing invalid node names
```

Test Explorer

2 ✓ 2 ✕ 0

Search (Ctrl+I)

Test run finished: 2 Tests (2 Passed, 0 Failed, 0 Skipped) run in 171 ms

⚠ 0 Warnings ✕ 0 Errors

Test	Duration	Traits
✓ RouteWiseAPP.API.TEST (2)	145 ms	
✓ RouteWiseAPP.API.TEST.UseCases.Nodes (2)	145 ms	
✓ GetShortestPathQueryHandlerTests (2)	145 ms	
✓ Handle_ShouldReturnShortestPath_WhenInvalidRequestIsProvided	72 ms	
✓ Handle_ShouldReturnShortestPath_WhenValidRequestIsProvided	73 ms	

Run | Debug

Test Detail Summary

✓ RouteWiseAPP.API.TEST.UseCases.Nodes.GetShortestPathQueryHandlerTests.Handl

Source: GetShortestPathQueryHandlerTests.cs line 133

Duration: 72 ms

103 % ✕ 0 ⚠ 1 Ln: 128 Ch: 44 SPC CRLF

'ShortestPath' implementations - Entire solution

Entire solution | Group by: Project only | Keep Results | List View | Search Find All References

Code	File	Line	Col	Project
RouteWiseApp.INFRASTRUCTURE (2)				
ShortestPathData NodePathFinder.ShortestPath(string, string, List<Node>)	NodePathFinder.cs	13	33	RouteWiseApp.INFRASTR...
ShortestPathData NodePathFinderService.ShortestPath(string, string, List<Node>)	NodePathFinderService.cs	21	33	RouteWiseApp.INFRASTR...

Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'RouteWiseApp' (6 of 6 projects)
  GitHub Actions
  docs
  src
    Application
      RouteWiseApp.APPLICATION
        Dependencies
        Exceptions
          InvalidException.cs
        Repositories
        Services
        UseCases
          Nodes
            Command
            Query
              GetAllNodes
              GetShortesPath
                GetShortesPathQuery.cs
                GetShortesPathQueryRequestDTO.cs
                GetShortesPathQueryResponseDTO.cs
            NodeUseCaseBase.cs
          UseCaseBase.cs
    Domain
      RouteWiseApp.DOMAIN
        Dependencies
        AppModels
        DomainModels
    Infrastructure
      RouteWiseApp.INFRASTRUCTURE
        Dependencies
        Repositories
        Services
    Presentation
      RouteWiseApp.API
        Connected Services
        Dependencies
        Properties
        Controllers
        logs
        Middlewares
          ExceptionHandlingMiddleware.cs
          LoggingMiddleware.cs
        appsettings.json
        Dockerfile
        Program.cs
        RouteWiseApp.API.http
        WeatherForecast.cs
      RouteWiseAPP.ConsoleApp
        Dependencies
        Program.cs
  test
    Integration
    Unit
      RouteWiseAPP.API.TEST
        Dependencies
        Models
        UseCases
          Nodes
            GetShortestPathQueryHandlerTests.cs
```

Test run finished: 2 Tests (2 Passed, 0 Failed, 0 Skipped) run in 171 ms