

Please, read the problem of the exercise, choose your technologies of choice (preferably C# or Java) and at the end of the exercise, please commit your solution to a GitHub Repository and send us the repo url so we can see the solution and guarantee that all the necessary description to use the solution is on the read.me file.

You have 1 week, starting on the day that the email was sent to finish the exercise.

In Farfetch we do appreciate that a candidate demonstrate knowledge in:

- Object oriented programming;
- SOLID Principles;
- Clean Code;
- Unit/Automation/Integration Testing;
- Design Patterns;
- REST API;

As extra that we also appreciate:

- Authentication & Authorization;
- SQL and NoSQL database;
- Message Driven Development;
- Service built with the possibility to scale and with high performance.

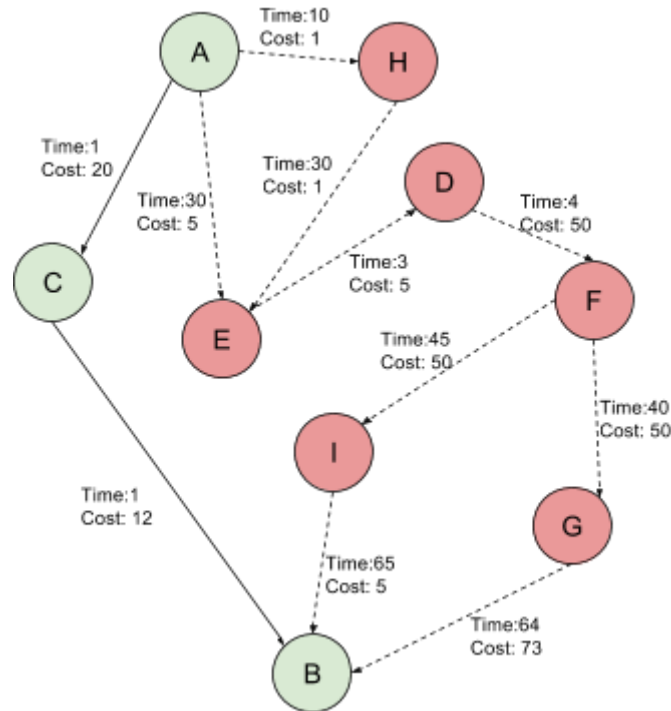
Delivery Service

Company *XPTO* has a ecommerce platform built under SOA and currently with 62 services/applications built within. This platform enables the final users to search by products, buy and ship them. However, the shipping process it's a manual process to find out which route should be used.

This process is painful, and the company wants to have a solution so they can infer the route to be used to ship certain product from point A to B.

Some rules must be accomplished within the solution:

- The route must not perform a direct delivery between the origin and the destination, for example, if a delivery must be made from point A to point B, it must choose the routes that pass through intermediate points, as for example, A - C - B (see figure below)
- Each route have a cost and time associated so it is possible to retrieve the best option depending on the cost our time (in the figure below, it exists multiple options to deliver from A to B, however the route A-C-B has a less time to deliver).



Please, develop a service with an REST API (don't need to implement an UI interface for this) where it will be possible to:

- Retrieve the routes within a defined origin and destination points;
- Manage points and routes (implement the CRUD);
- Guarantee a minimum amount of tests so you feel confident with any developments in the future;
- Please inject some example data to your solution database.

As a Plus

- Only users with admin permission may create, update or delete new points or routes.

Don't forget, you must be proud of your delivery and completely confident to deploy it to production.