

# The Mail Service

---

Our mail service needs a new scheduling system for its facility coming online... next month!

The mail/parcels are received at our service/distribution center. There are number of rail lines (train tracks) extending out where eventually items will reach their destinations.

Each parcel of mail has a

- Weight (kg)
- Volume in  $\text{cm}^3$

**Train Operators** bid to carry these parcels in trains which have a maximum weight and a maximum volume.

**Parcel Owners** submit parcels to be shipped via our APIs.

## Hiring Trains & Charging Parcel Owners

---

Our mail service hires trains from the Train Operators and assigns parcels to them. It then charges each Parcel Owner a proportionate amount of the cost of shipping their parcel.

If we charge Parcel Owners too much, they will shift to another operator, so it is in our best interest to charge the owners as little as possible (turns out we're a government entity).

## REST Interface

---

For this exercise, you should design a small program which features a REST API that allows:

### Train Operators

- Tell the system
  - Trains that are currently available
  - Current capacity and cost
  - Lines they can operate on
  - Withdraw bids
- Ask the system
  - Which trains our system has booked
  - Line that it is assigned to
  - Time the train left

### Parcel Owners

- Tell the system
  - A parcel for shipping
  - To withdraw a package
- Ask the system
  - has the package shipped
  - the cost of shipping

### Post Master

- Ask the system
  - What parcels were loaded into which trains and what line the train went out on

## Constraints

---

- Train Operator posts an offer for a train, which you can then book it unless it was previously withdrawn.
- When you book a train, it is instantly loaded and leaves.
- Trains are a one-time thing when you book a train for a line, it is used and cannot be rebooked. Trains will be re-submitted by the Train Operator as new trains.
- Booked trains take 3h to run. This makes the line unavailable during that time.
  - If you send a train on an occupied line, they will crash!

## Notes

---

Finesse is not required, and neither is a frontend, though you should give enough documentation that we can verify your solution is correct.

Your solver need not be complex, but must make an attempt to solve the optimisation problem and you should document how far you decided to go in doing so.

You will note that there are some ambiguities in the above, and some points where you will have to make design decisions. You should clearly note which ones you spotted and what restrictions you placed on behaviour in order to make the problem tractable.

Your REST API design need not be clever, or optimise load, but must be functionally correct.

You need not worry about authenticating owners, Train Operators, or postmasters.

This task can easily take way longer than 6 hours if you allow it so don't let that happen.

## Example

---

An example configuration might be:

3 Lines - A, B, C

Trains:

- Thomas - cost 200, total weight 500kg, total volume 8 m3, lines A B
- Percy - cost 150, total weight 150kg, total volume 1 m3, lines B C
- Evelyn - cost 300, total weight 300kg, total volume 4 m3, lines A C
- Toby - cost 250, total weight 100kg, total volume 2 m3, lines B C

Packages:

- 0001 - weight 1kg , volume 20 cm3
- 0002 - weight 0.5kg, volume 5 cm3
- ...