

# Backend Programming Challenge - 2019

Create a basic automatic dispatch system for some over-the-road shipments! Your app will be able to read from a list of shipments that need to be picked up and a list of available drivers and their locations, and hit an existing REST endpoint to dispatch these shipments to nearby drivers.

You are given the following:

- JSON file representing a list of drivers with their current locations
- driverId is the key of each JSON object (1, 2, 3...)
- JSON file representing a list of 5 active shipments and their pickup locations
- shipmentId is the key of each JSON object (65289023243, 3823958290...)

Create an app that runs a periodic task to dispatch each of the 5 shipments to the three closest drivers every 10 seconds. Dispatch a shipment to a driver by hitting the dispatch REST endpoint, which is detailed below.

If none of the three drivers accept the dispatch in that run, expand your radius to dispatch to the next 3 closest drivers. Allow your periodic task to continue to run and repeat this process until a dispatch has been Accepted, or you've dispatched to all the available drivers.

Do not dispatch a shipment to the same driver more than once. You may dispatch multiple shipments to the same driver.

**Bonus:** Write unit tests to test the functionality of your app.  
When completed, zip your project and send it to [jeremy@shipwithbolt.com](mailto:jeremy@shipwithbolt.com).

## Dispatch REST Endpoint

Create a dispatch for a shipment by hitting the following endpoint:

**POST** `http://challenge.shipwithbolt.com/driver/:driverId/dispatch`

Replace `:driverId` in the URL with the driverId you are dispatching to

Pass in as the body in JSON format:

```
{
  shipmentId: x (number)
}
```

If the driverId is invalid (does not exist in the driver JSON file), a 404 error will be returned.

If the shipmentId is missing or invalid (does not exist in the shipment JSON file), a 400 error will be returned.

If the driverId is valid, you will receive the following payload:

```
{
  response: 'Accepted' or 'Denied' (string), driverId: x (number),
  shipmentId: x (number)
}
```

The response will be random - **Accepted** 15% of the time, and **Denied** 85% of the time.

Notes:

- Feel free to use an existing formula or solution for determining proximity from a shipment to a driver.

- No time limit with this programming challenge - take your time and showcase your coding

style and skills. We're looking for submissions that emulate what you would write in a production codebase.

- Use any programming language that you are comfortable with.

