# Exercise Notes

Looking at this exercise, the steps I will take are as follows:

1. Determine the formulas for the deliveries.
2. Plan the modules (ScheduleDelivery(), ListDrones() etc)
3. Check if the planned simulator works using the formulas.
4. Build.

Package example:

Delivery 1:

Weight = 20kg

Distance = 2.82km

Speed = 40 km/h

Time = 4 min 14 seconds.

Delivery 2:

Weight = 5kg

Distance = 1.4km

Speed= 70 km/h

Time = 51 seconds

Trip home:

Weight: 0

Distance = 3.16km

Speed = 80 km/h

Time = 2 min 22 secs

Total time = 7 min 27 seconds

Time left before refuelling = 52 minutes 33 seconds

Each coordinate = 1 km from home base therefore (0,1) would be 1 km.

I will work on the assumption that each coordinate increases 1 km until I can do this dynamically.

| Important information to note. | |
| --- | --- |
| Drone Default Speed | 80 km/h |
| Drone speed at max weight | 40 km/h |
| Coordinates represent an increase in distance of 1 km (**in the example**) | Example, distance to (2,2) equals 2.82 because the shortest distance (the hypotenuse) is 2.82 at that coordinate. |
| **Speed is inversely proportional to weight.** | |

Time Log 1: I started at 3 pm and it is 3:42 pm right now.

* I have created a front end page first in order to visualise the CRUD operations. I will first start with getting all drones and entering a new drone.
* Drone details will include:
  + DroneID (or DroneName, because I’m not creating a database)
  + Status (Travelling or Not)
  + NumPackages
  + Deliveries (array with the ETA for each package and ID)
  + CurrentWeight
  + TimeToRefuel
  + TotalTravelTime

Time Log 2: 16:06

* I realised that I may be doing work outside the requirement.
* I need to build an api, not a system.

What do I need to do?

Create an API with the following functionalities:

Create Drone : POST request (Using the above mentioned data)

Schedule Delivery : POST request (Enter coordinates, and package weight)

(We must get the information about each delivery after the post request)

View All Drones : GET request (Table with all the drones and their data)

View Packages : GET request

Data for packages with include

* PackageID
* ETA
* DroneID
* DistanceToTravel (This must account for if the package is one of many trips)

19:00 - The routes I am going to do are as follows

| Route | Request |
| --- | --- |
| /api/drones/ | GET |
| /api/drone/:id | GET, POST, PATCH |
| /api/packages | GET |
| /api/package/:id | GET, POST |
| /api/makeDelivery | POST |
|  |  |

Now I will set up all the routes.

19:13 - I am starting to work with the routes.

19:40 - I have put all my routes into a “api/routes/” folder so they are easier to manage. I am doing this so that when I get to the part where I have to implement the endpoints, I can guarantee they are working and can debug efficiently if they are not.

20:00 - I have completed the route set up and confirmed that all the routes are working correctly (confirmed using Postman). Tomorrow I will set up the functionality of each other the endpoints

Friday - 14:30

I am working on the (Register Drone) functionality. I will get user input via request, and then save it on the node server.

**\*A lot of this stuff I have not done in a while so I have been using a number of resources on google to catch up.**

14:44 - Moved the post request for registering a drone to “/api/drones” because the id is not yet known and would cause errors later on.

When a used registers a drone, most of the drone data will be default and the ID needs to be auto-added and unique (could also leave it as unique and figure out the auto-increment later on)

The idea is that the user would select “register drone” and a new drone would be created, since most of the information should not be inputted by the user during creation.

15:38 - Right now, my main issue is I can add a new drone, but I don’t really know how to retrieve that information in the GET request. Looking at options online now.

15:41 - I think the best thing to do is create a MongoDB database. Just so the data doesn’t disappear everytime I refresh the server.

16:28 - I am creating models for my data. For the variables that have to do with time, I decided that they will be Number variables, in seconds, which will be easier to convert to minutes and hours when needed. The ‘current weight’ variable will be in grams too.

16:47 - I am getting an error trying to POST data into my db. When I do the POST request in Postman for /api/drones, postman loads continuously and there's no output. Checking to see if my db is connected correctly.

16:50 - Data is not being sent to my mongodb atlas, the db connection is working however. Debugging.

17:06 - I am still having the error, I have decided to use only a couple inputs in my model and post request to see if that works, if it does, that means my previous structure was incorrect, if not, then I need to keep searching.

17:09 - It did not work.

17:29 - I am still having the error, and I have tried many different fixes I’ve seen on the internet but still no solution.

17:50 - I did some investigating and it seems like all my routes are not sending back information, that may be the same issue.

17:59 - Now that I figures out that there’s something wrong with the server, I decided to comment out everything and do a simple hello world server and listen on port 5000,that works, so now I will uncomment out the other code until I get the same issue so I can triangulate the problem.

18:03 – The issue had to do with the Cors middleware I am using. I need to figure out what I am doing incorrectly.

18:11 - Commented out the previous cors middleware and added “app.use(cors())” and now it works. I do not know why.

18:36 - Refactored some of my files because they were unclear, I had redundant files for the routes.

18:51 - API is now able to add a new drone, view one or multiple drones successfully. I will call it a day. The package and delivery’s functionality are what's left.

Saturday 13:00 - Starting work for the day. I am going to set up the (createPackage()) functionality. What I will need to do is create a query where:

* Get the package weight
* Get the package destination
* Get the first drone that can fulfil this package delivery.
* If there is none, display a message to the user
* If there is one, create package and update drone
* If the package is over 20kg (20000 grams) or the delivery and return would take over 1 hour (3600 seconds), display a message to the user.

First I will create a model for packages

I read over the requirements again and it says (array of packages) which means that each delivery would have one or many packages and I do not have to add that dynamically. That means we can remove (package) and have delivery do all the functionality.

14:32 - I’ve created the packages model and I found a way to map over an array of objects in order to enter them into the db, however I do not know how to use each destination to calculate the most efficient route between packages.

14:32 - The endpoint works. However I still do not know how I can calc route between packages. **The api currently meet the the criteria for (2a, 2b, 2c) of the requirements**

14:57 I am refactoring some code and formatting in order for it to be readable, however I am stopping right now. I am pretty sure I’ve worked on the project for over 10 hours and I feel now I am over engineering and confusing myself.

13:15 - Get request to get all the packages works.

Ending notes:

* I first made a mistake at the beginning by jumping into creating a frontend site for this test which was not requested and derailed my progress.
* Highlighting important points in the slides to focus on key areas was really helpful.
* I needed to create a database because I wanted to make the data easier to visualise and work with continuously.
* The api doesn’t meet all the criteria for 2c, the drones are listable but the data is just the same is still default. The idea was that drone data would change when a package is added.