Zid-Task documentation

Safwan Saigh safwan9f@gmail.com, 0534501056

I. Table of Contents

II. Introduction

This document is used to demonstrate for you what are the technologies and recourses used to develop a Shipment API System. In addition, it illustrates the steps followed to develop the system. Further, the document shows how to use the system. However, this document is used as a task requirement requested by Zid. All the information here is written by the developer (Safwan Saigh). If there is anything not clear, please contact me through my email.

III. Technologies Used

1) Databases

a) MySQL

I have decided to go with SQL database because it is the first thing came to my mind.

b) MongoDB

After a while from implementing the SQL database, I decided to follow the convention of the MEAN stack. In addition, working with NoSQL is much easier that working with SQL database.

2) Backend language

a) JavaScript (Node js)

I have chosen Node js because I can work with it quickly. Also, it is more fun than Python(Flask) and PHP.

3) Backend framework

a) Express

4) Backend Libraries
   a) Bcryptjs: Encrypting passwords saved in the database
   b) Jsonwebtoken: Authenticate API’s users
   c) Mongoose: Work with MongoDB
   d) Joi: Validate body requests coming from the frontend
   e) Pdfkit: Create labels as pdf documents
   f) Aws-sdk: Work with S3 service for storing attachments
   g) Multer: File uploads
   h) Multer-s3: File uploads to S3

5) Other services
   a) Postman

6) Hosting services
   a) Heroku for the backend server
   b) FreeDB for the SQL database server
   c) MongoDB Atlas for the NoSQL database server
   d) Amazon S3 for the files’ storage

IV. Resources

- Stack overflow
- Heroku documentation
- Amazon S3 documentation
- Multer library documentation
- Pdfkit library documentation
- Jwt library documentation
- Youtube
- ShipEngine

V. Implementation Summery

1- Setup the node js project

2- Implement simple endpoints for listing and creating shipments with proper validation

3- Implement an endpoint for tracking shipments

4- Implement an endpoint to download a label

5- Implement generating shipments labels and save them as pdf docs

6- Update shipment’s schema

7- Setup Heroku server and deploy the app

8- Developing SQL schema for shipments data

9- Setup MySQL database

10- Use the database to save and retrieve shipments

11- Setup the S3 service, this includes creating the buckets that will hold the files

12- Implement uploading files and attachments to the S3

13- Refactoring and bugs fixes

14- Deciding to replace MySQL with MongoDB, this decision was clarified in the Database section

15- Change the database schemas to schemeless, which means tables to collections and tuples to documents

16- Setup MongoDB database in the MongoDB Atlas

17- Create the required models and implement the new database

18- Re-implement the previous functionalities with the new database

19- Refactoring and bug fixes

20- Setup authentication schemas and models

21- Implement register and login a user with decrypting the sensitive information

22- Implement token authentication

23- Refactoring

24- Change the shipment’s model with adding more attributes and enhancing the population from another collections

25- Change the user input validation when creating a new shipment

26- Trying to secure the API more

27- Refactoring and bug fixes

VI. Assumptions and Constraints

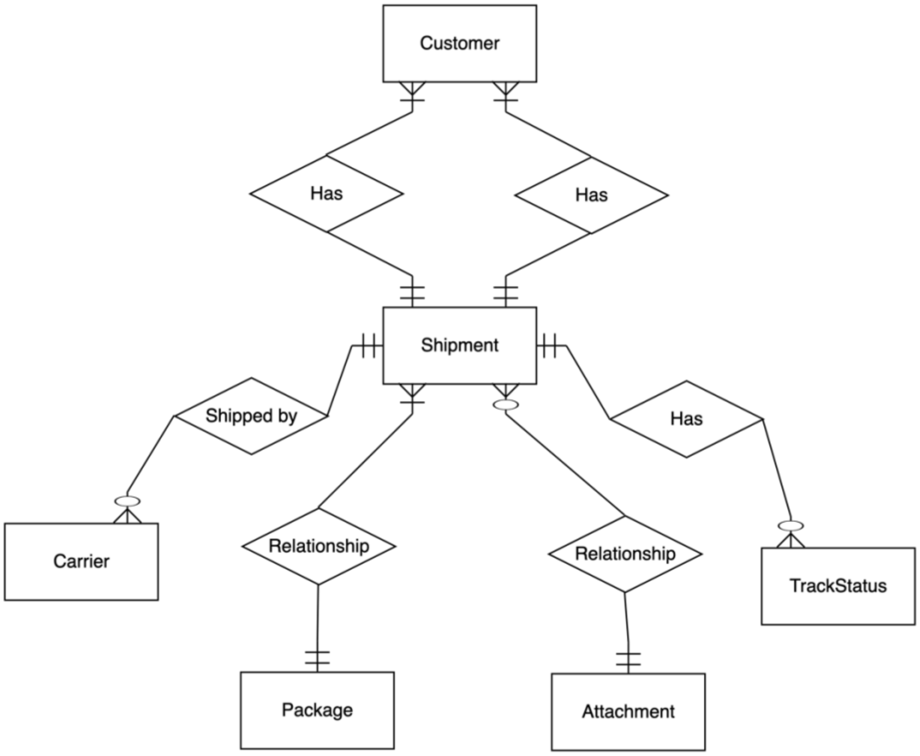
1- Assuming there are two actors for this system which are a regular user and an admin

2- Assuming in the frontend side there should be two main views, one for tracking the shipments and the second is an Admin panel. The admin can create and list shipments. Also, print shipments labels.

3- Assuming an admin is a generalized actor for the seller, the warehouse worker, carrier, or the system owner. All of them can enter the system and update a shipment status.

I. Schemas and Models

1- MySQL Schema



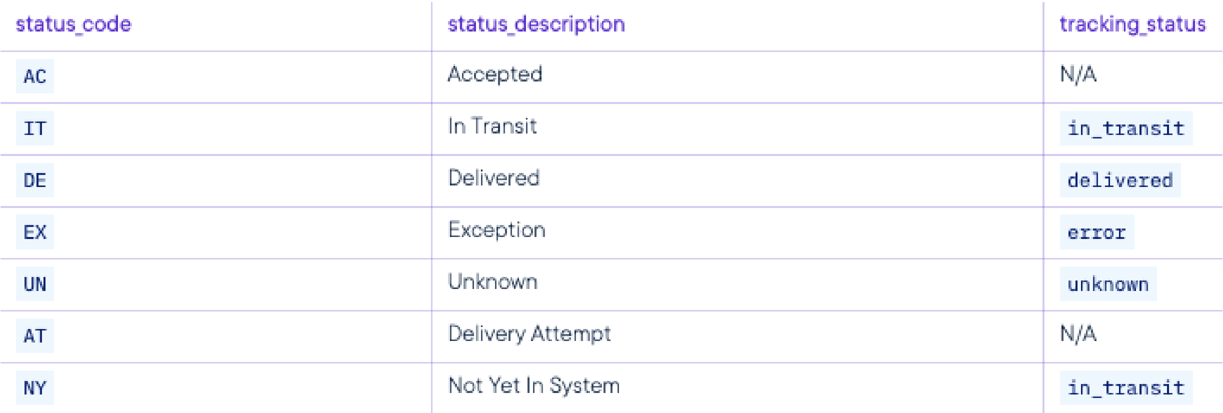
2- MongoDB collections and Models

a) Collections: Carriers, Shipments, Users, and TrackingStatuses

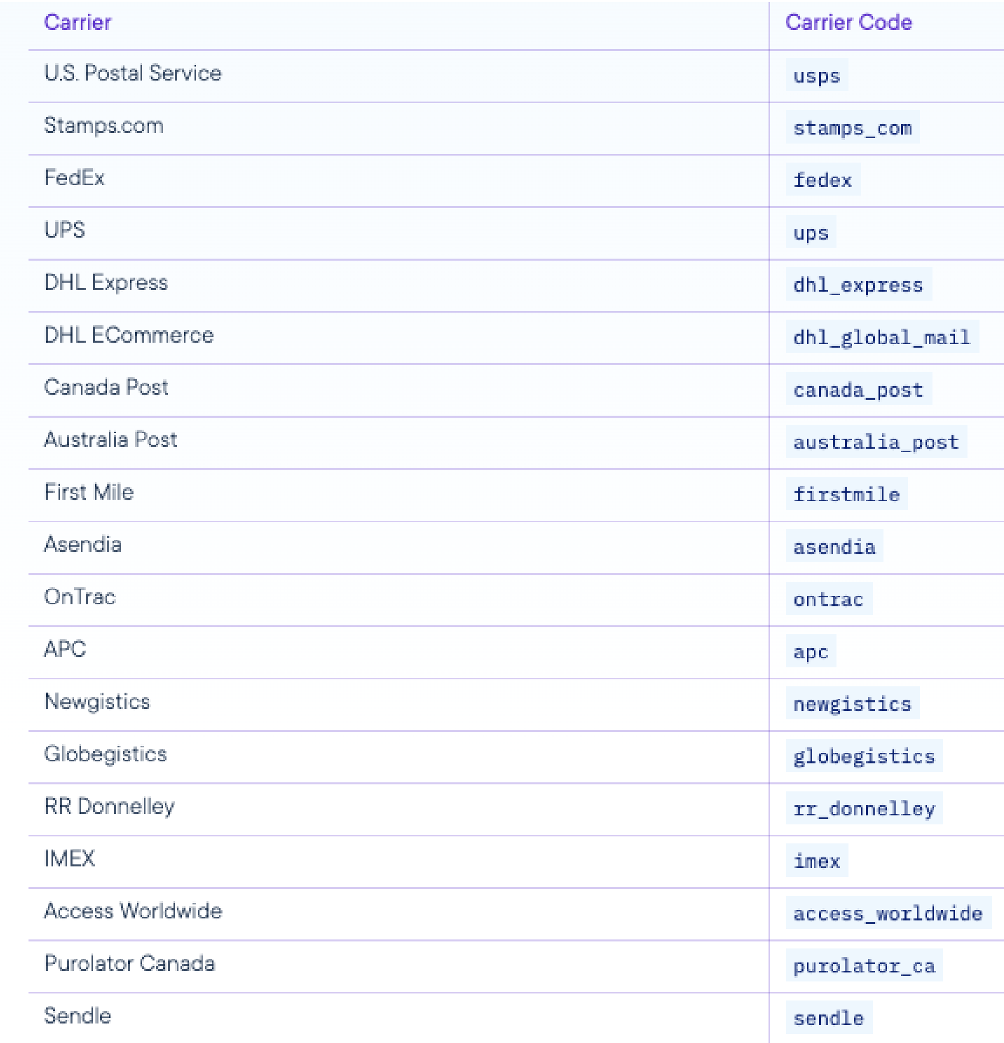
b) Models: Carrier, Shipment, User, and TrakingStatus

VII. Data used

I have used the tracking statuses and carrier data used in *ShipEngine* API with little adjustments.



status_code	status_description	tracking_status
AC	Accepted	N/A
IT	In Transit	in_transit
DE	Delivered	delivered
EX	Exception	error
UN	Unknown	unknown
AT	Delivery Attempt	N/A
NY	Not Yet In System	in_transit



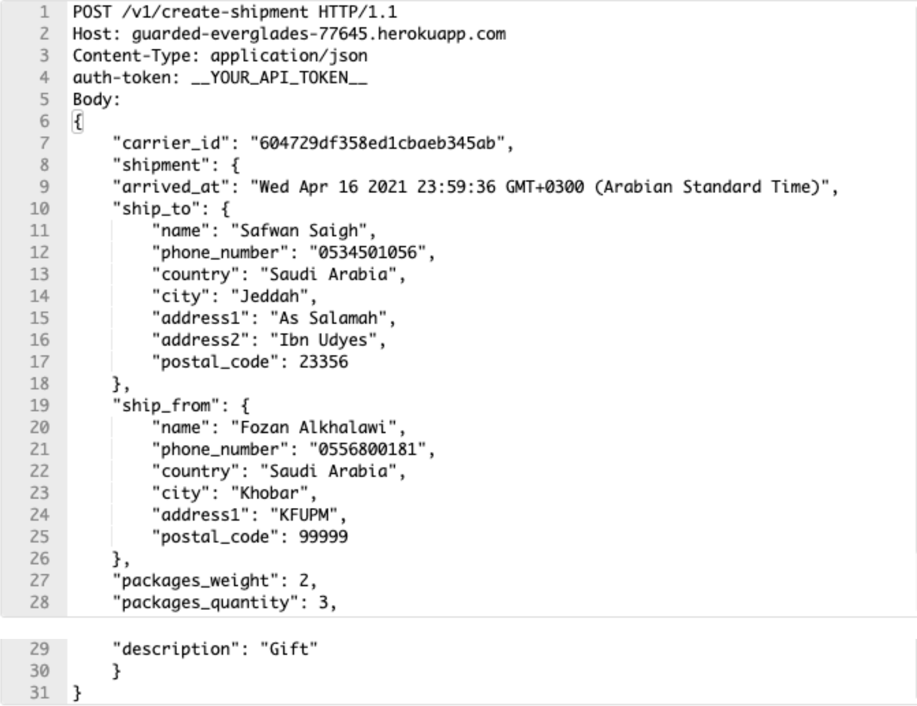
Carrier	Carrier Code
U.S. Postal Service	usps
Stamps.com	stamps_com
FedEx	fedex
UPS	ups
DHL Express	dhl_express
DHL ECommerce	dhl_global_mail
Canada Post	canada_post
Australia Post	australia_post
First Mile	firstmile
Asendia	asendia
OnTrac	ontrac
APC	apc
Newgistics	newgistics
Globegistics	globegistics
RR Donnelley	rr_donnelley
IMEX	imex
Access Worldwide	access_worldwide
Purolator Canada	purolator_ca
Sendle	sendle

VIII. API Usage

This section will show you how to use the API based on the functionality.

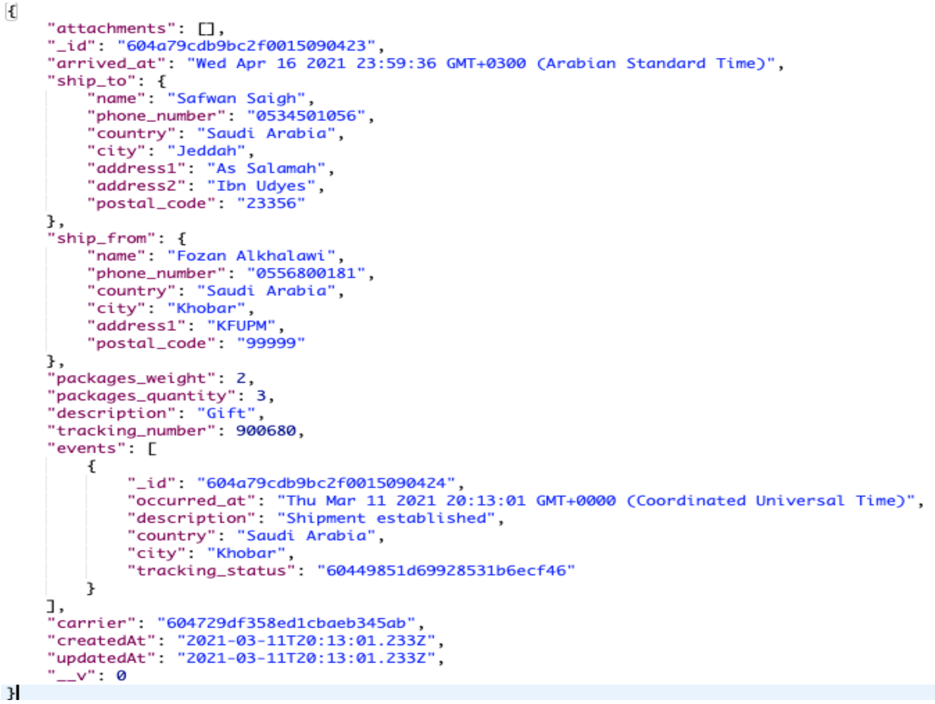
1- Create a shipment

Example Request



```
POST /v1/create-shipment HTTP/1.1
Host: guarded-everglades-77645.herokuapp.com
Content-Type: application/json
auth-token: __YOUR_API_TOKEN__
Body:
{
    "carrier_id": "604729df358ed1cbaeb345ab",
    "shipment": {
    "arrived_at": "Wed Apr 16 2021 23:59:36 GMT+0300 (Arabian Standard Time)",
    "ship_to": {
        "name": "Safwan Saigh",
        "phone_number": "0534501056",
        "country": "Saudi Arabia",
        "city": "Jeddah",
        "address1": "As Salamah",
        "address2": "Ibn Udyes",
        "postal_code": 23356
    },
    "ship_from": {
        "name": "Fozan Alkhalawi",
        "phone_number": "0556800181",
        "country": "Saudi Arabia",
        "city": "Khobar",
        "address1": "KFUPM",
        "postal_code": 99999
    },
    "packages_weight": 2,
    "packages_quantity": 3,
    "description": "Gift"
    }
}
```

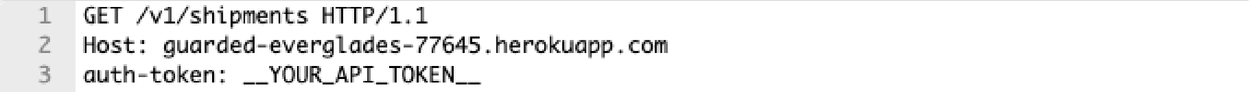
Example Response



```
{
    "attachments": [],
    "_id": "604a79cdb9bc2f0015090423",
    "arrived_at": "Wed Apr 16 2021 23:59:36 GMT+0300 (Arabian Standard Time)",
    "ship_to": {
        "name": "Safwan Saigh",
        "phone_number": "0534501056",
        "country": "Saudi Arabia",
        "city": "Jeddah",
        "address1": "As Salamah",
        "address2": "Ibn Udyes",
        "postal_code": "23356"
    },
    "ship_from": {
        "name": "Fozan Alkhalawi",
        "phone_number": "0556800181",
        "country": "Saudi Arabia",
        "city": "Khobar",
        "address1": "KFUPM",
        "postal_code": "99999"
    },
    "packages_weight": 2,
    "packages_quantity": 3,
    "description": "Gift",
    "tracking_number": 900680,
    "events": [
        {
            "_id": "604a79cdb9bc2f0015090424",
            "occurred_at": "Thu Mar 11 2021 20:13:01 GMT+0000 (Coordinated Universal Time)",
            "description": "Shipment established",
            "country": "Saudi Arabia",
            "city": "Khobar",
            "tracking_status": "60449851d69928531b6ecf46"
        }
    ],
    "carrier": "604729df358ed1cbaeb345ab",
    "createdAt": "2021-03-11T20:13:01.233Z",
    "updatedAt": "2021-03-11T20:13:01.233Z",
    "__v": 0
}
```

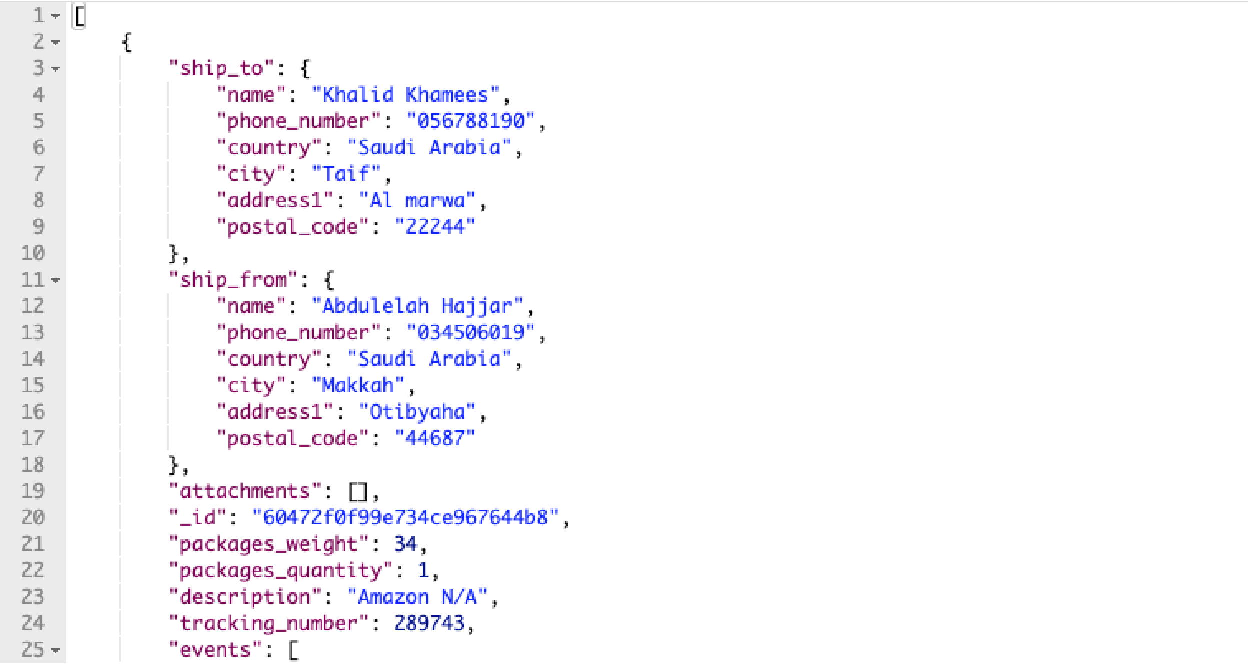
2- Listing previous shipments

Example Request

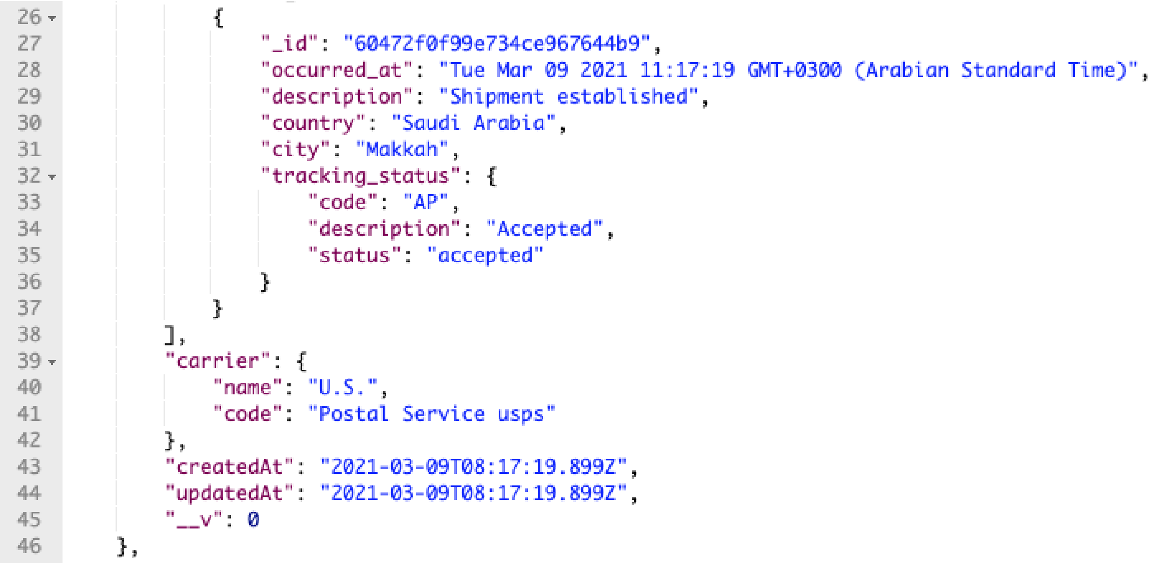


```
GET /v1/shipments HTTP/1.1
Host: guarded-everglades-77645.herokuapp.com
auth-token: __YOUR_API_TOKEN__
```

Example Response



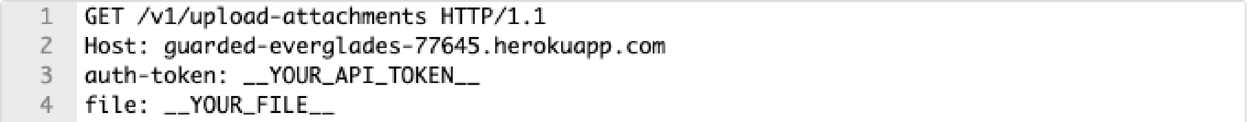
```
[
    {
        "ship_to": {
            "name": "Khalid Khamees",
            "phone_number": "056788190",
            "country": "Saudi Arabia",
            "city": "Taif",
            "address1": "Al marwa",
            "postal_code": "22244"
        },
        "ship_from": {
            "name": "Abdulelah Hajjar",
            "phone_number": "034506019",
            "country": "Saudi Arabia",
            "city": "Makkah",
            "address1": "Otibyaha",
            "postal_code": "44687"
        },
        "attachments": [],
        "_id": "60472f0f99e734ce967644b8",
        "packages_weight": 34,
        "packages_quantity": 1,
        "description": "Amazon N/A",
        "tracking_number": 289743,
        "events": [
```



```
            {
                "_id": "60472f0f99e734ce967644b9",
                "occurred_at": "Tue Mar 09 2021 11:17:19 GMT+0300 (Arabian Standard Time)",
                "description": "Shipment established",
                "country": "Saudi Arabia",
                "city": "Makkah",
                "tracking_status": {
                    "code": "AP",
                    "description": "Accepted",
                    "status": "accepted"
                }
            }
        ],
        "carrier": {
            "name": "U.S.",
            "code": "Postal Service usps"
        },
        "createdAt": "2021-03-09T08:17:19.899Z",
        "updatedAt": "2021-03-09T08:17:19.899Z",
        "__v": 0
    },
```

3- Attach documents to a shipment

Example Request



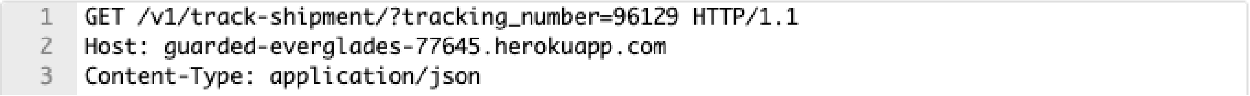
```
GET /v1/upload-attachments HTTP/1.1
Host: guarded-everglades-77645.herokuapp.com
auth-token: __YOUR_API_TOKEN__
file: __YOUR_FILE__
```

Example Response

The attachment will be added to the specified shipment

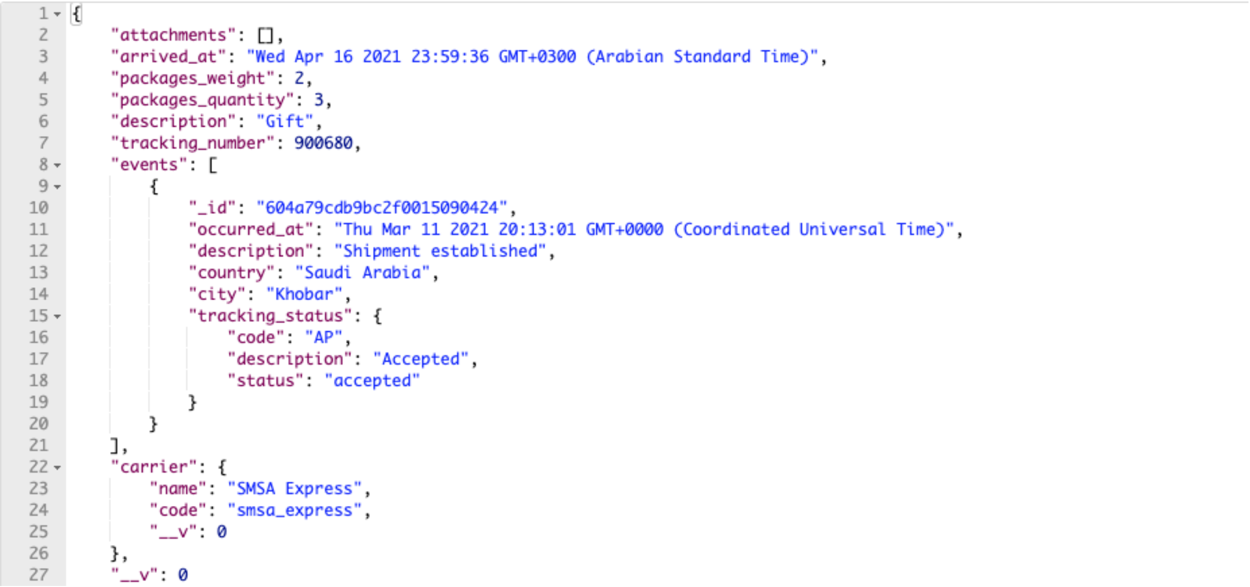
4- Tracking a shipment

Example Request



```
GET /v1/track-shipment/?tracking_number=96129 HTTP/1.1
Host: guarded-everglades-77645.herokuapp.com
Content-Type: application/json
```

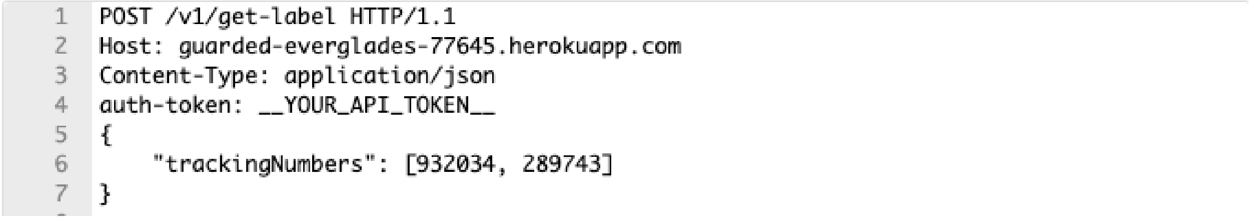
Example Response



```
{
    "attachments": [],
    "arrived_at": "Wed Apr 16 2021 23:59:36 GMT+0300 (Arabian Standard Time)",
    "packages_weight": 2,
    "packages_quantity": 3,
    "description": "Gift",
    "tracking_number": 900680,
    "events": [
        {
            "_id": "604a79cdb9bc2f0015090424",
            "occurred_at": "Thu Mar 11 2021 20:13:01 GMT+0000 (Coordinated Universal Time)",
            "description": "Shipment established",
            "country": "Saudi Arabia",
            "city": "Khobar",
            "tracking_status": {
                "code": "AP",
                "description": "Accepted",
                "status": "accepted"
            }
        }
    ],
    "carrier": {
        "name": "SMSA Express",
        "code": "smsa_express",
        "__v": 0
    },
    "__v": 0
```

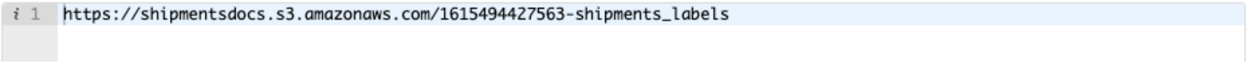
5- Get shipments' labels

Example Request

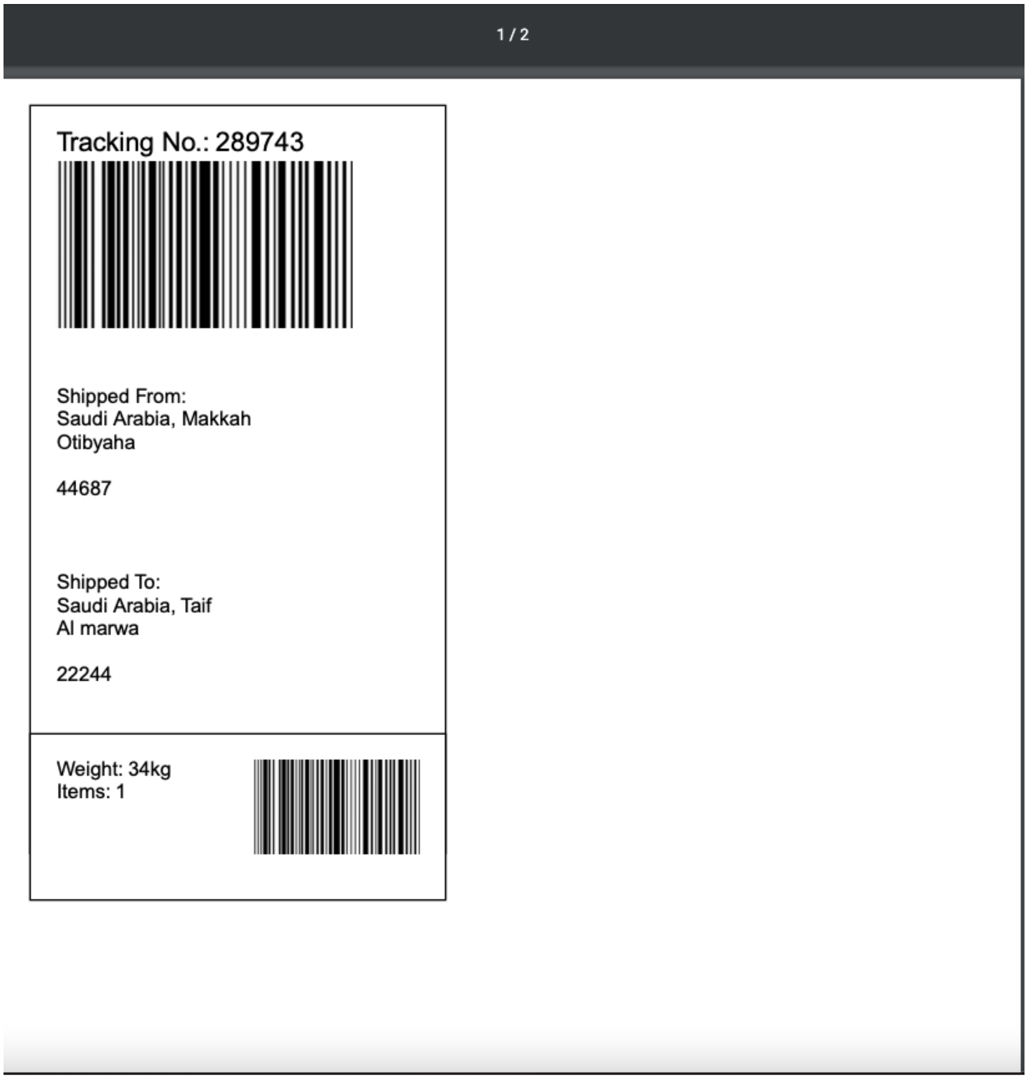


```
POST /v1/get-label HTTP/1.1
Host: guarded-everglades-77645.herokuapp.com
Content-Type: application/json
auth-token: __YOUR_API_TOKEN__
{
    "trackingNumbers": [932034, 289743]
}
```

Example Response



```
https://shipmentsdocs.s3.amazonaws.com/1615494427563-shipments_labels
```



Tracking No.: 289743

Shipped From:
Saudi Arabia, Makkah
Otibyaha

44687

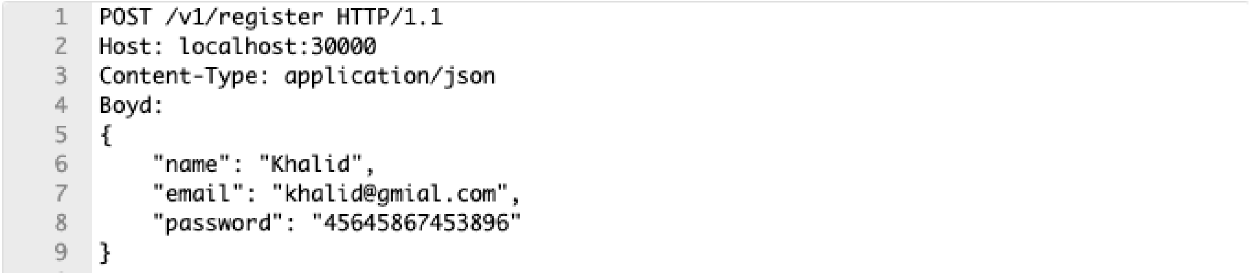
Shipped To:
Saudi Arabia, Taif
Al marwa

22244

Weight: 34kg
Items: 1

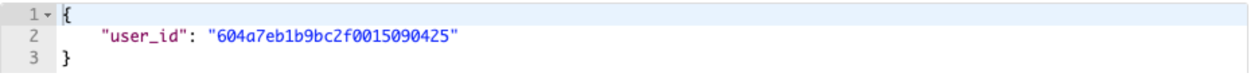
6- Simple Register

Example Request



```
POST /v1/register HTTP/1.1
Host: localhost:30000
Content-Type: application/json
Boyd:
{
    "name": "Khalid",
    "email": "khalid@gmial.com",
    "password": "45645867453896"
}
```

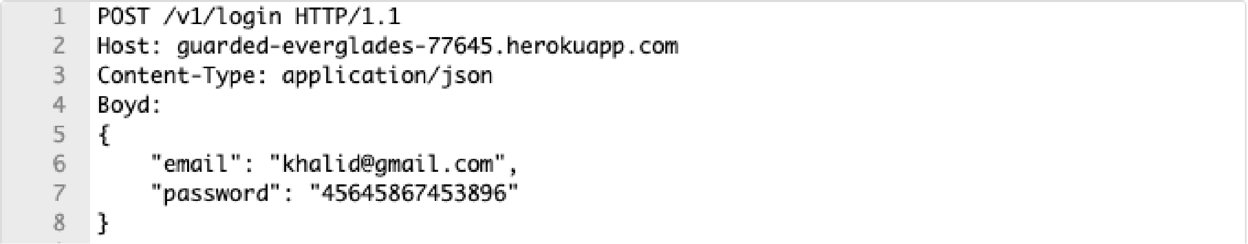
Example Response



```
{
    "user_id": "604a7eb1b9bc2f0015090425"
}
```

7- Simple Login

Example Request



```
POST /v1/login HTTP/1.1
Host: guarded-everglades-77645.herokuapp.com
Content-Type: application/json
Boyd:
{
    "email": "khalid@gmail.com",
    "password": "45645867453896"
}
```

Example Response Header

auth-token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2MDQ2NWM0YTEyYzk0ZGE2MDIwY2U5YjEiLCJpYXQiOjE2MTU0OTMzNTN9.usVBcNkgs8omdMLOtzaOWEFYEO0VF8MUQRRS-5ZnAJA

IX. Conclusion

In conclusion, I really enjoyed working on such a system and I have learned many things. Unfortunately, it was a very busy 2 weeks due to the midterm exams at KFUPM, so I could not implement the webhooks and the optional task. However, I have more ideas and I believe I can enhance the code as well as some functionality. I hope I did what it requires to have an internship opportunity within Zid. If there is anything not clear, do not hesitate to contact me on safwan9f@gmail.com.