**TRAINING Excersice**

Table of Contents

[1 Introduction 2](#_Toc452637153)

[2 getQuote 2](#_Toc452637154)

[2.1 Rate data 2](#_Toc452637155)

[2.2 Request 3](#_Toc452637156)

[2.3 Response 4](#_Toc452637157)

[2.4 Process 4](#_Toc452637158)

[3 createShipment 4](#_Toc452637159)

[3.1 Request 4](#_Toc452637160)

[3.2 Response 5](#_Toc452637161)

[3.3 Process 5](#_Toc452637162)

[4 getShipment 5](#_Toc452637163)

[4.1 Request 5](#_Toc452637164)

[4.2 Response 6](#_Toc452637165)

[4.3 Process 7](#_Toc452637166)

[5 deleteShipment 7](#_Toc452637167)

[5.1 Request 7](#_Toc452637168)

[5.2 Response 7](#_Toc452637169)

[5.3 Process 7](#_Toc452637170)

[6 trackShipment 8](#_Toc452637171)

[6.1 Request 8](#_Toc452637172)

[6.2 Response 8](#_Toc452637173)

# Introduction

With the purpose of verifying the output after studying and training time, new member need to complete this exercise to show how he is ready to join Temando project.

We need to use NodeJS to complete this exercise.

The idea is to simulate a simple shipping work follow with below **REST APIs**,

* getQuote
* createShipment
* getShipment
* deleteShipment
* trackShipment



# getQuote

## Rate data

|  |  |  |  |
| --- | --- | --- | --- |
| **Weight** | **Price** | **From Country** | **To Country** |
| <= 250 g | 12,43 USD | FR | FR |
| 500 g | 12,43 USD | FR | FR |
| 750 g | 15,42 USD | FR | FR |
| 1 000 g | 15,42 USD | FR | FR |
| 2 000 g | 20,77 USD | FR | FR |
| 3 000 g | 26,07 USD | FR | FR |
| 4 000 g | 31,43 USD | FR | FR |
| 5 000 g | 36,77 USD | FR | FR |
| 6 000 g | 42,13 USD | FR | FR |
| 7 000 g | 47,49 USD | FR | FR |
| 8 000 g | 52,83 USD | FR | FR |
| 9 000 g | 58,83 USD | FR | FR |
| 10 000 g | 63,54 USD | FR | FR |
| 11 000 g | 88,19 USD | FR | FR |
| 12 000 g | 88,19 USD | FR | FR |
| 13 000 g | 88,19 USD | FR | FR |
| 14 000 g | 88,19 USD | FR | FR |
| 15 000 g | 88,19 USD | FR | FR |
| > 15 000 g | 100 USD | FR | FR |

Load rate data into “rate” table in SQL Lite, this data will be used to query shipment price.

## Request

POST /client/getquote HTTP/1.1

{

"data": {

"origin": {

"contact": {

"name": "La Redoute Contact",

"email": "laredoute@example.com",

"phone": "07 1234 5678",

},

"address": {

"country\_code": "FR",

"locality": "Anzin",

"postal\_code": "59410",

"address\_line1": "Rue Jean Jaures",

}

},

"destination": {

"contact": {

"name": "Marquise de Pompadour",

"email": "marquise-de-pompadour@example.com",

"phone": "07 9876 5432",

},

"address": {

"country\_code": "FR",

"locality": "Marseille",

"postal\_code": "13006",

"address\_line1": "175 Rue de Rome"

}

},

"package": {

"dimensions": {

"height": 10,

"width": 10,

"length": 10,

"unit": "cm"

},

"grossWeight": {

"amount": 100,

"unit":kg,

}

}

}

}

}

## Response

{

"data": [

{

"id": "\_\_QUOTE\_ID\_\_",

"amount": 100, // USD

}]

}

## Process

* Query rate base on “rate” table, shipment weight, origin country and destination country.
* Origin country, destination country, weight fields are required

# createShipment

## Request

POST /client/creatshipment HTTP/1.1

{

"data": {

"quote": {

"id": "\_\_QUOTE\_ID\_\_"

},

"origin": {

"contact": {

"name": "La Redoute Contact",

"email": "laredoute@example.com",

"phone": "07 1234 5678",

},

"address": {

"country\_code": "FR",

"locality": "Anzin",

"postal\_code": "59410",

"address\_line1": "Rue Jean Jaures",

"organisation": true

}

},

"destination": {

"contact": {

"name": "Marquise de Pompadour",

"email": "marquise-de-pompadour@example.com",

"phone": "07 9876 5432",

},

"address": {

"country\_code": "FR",

"locality": "Marseille",

"postal\_code": "13006",

"address\_line1": "175 Rue de Rome",

"organisation": false

}

},

"package": {

"dimensions": {

"height": 10,

"width": 10,

"length": 10,

"unit": "cm",

},

"grossWeight": {

"amount": 100,

"unit":kg,

}

}

}

}

}

## Response

{

"data": [

{

"ref": "\_\_REFERENCE\_NUMBER\_\_", // Random number in 10 character

"created\_at": "2015-05-13T07:00:08+0000",

"cost": 10.00 // USD

}]

}

## Process

* Request data need to be stored into “shipment” table it will be used to lookup for other request.
* Based on origin and destination address from the request look up in the rate table in SQL Lite to calculate the price.
* All the request fields are required
* For the response
  + “ref” field will be generated randomly and store with the shipment
  + “cost” is the price of the shipment, this is queried by origin country, destination country and weight of the shipment.

# getShipment

## Request

GET /client/getshipment HTTP/1.1

{

"data": {

"ref": "\_\_REFERENCE\_NUMBER\_\_", // Use to lookup the shipment

}

}

## Response

**Shipment found:**

{

"data": {

"ref" : ""\_\_REFERENCE\_NUMBER\_\_",

"origin": {

"contact": {

"name": "La Redoute Contact",

"email": "laredoute@example.com",

"phone": "07 1234 5678",

},

"address": {

"country\_code": "FR",

"locality": "Anzin",

"postal\_code": "59410",

"address\_line1": "Rue Jean Jaures",

"organisation": true

}

},

"destination": {

"contact": {

"name": "Marquise de Pompadour",

"email": "marquise-de-pompadour@example.com",

"phone": "07 9876 5432",

},

"address": {

"country\_code": "FR",

"locality": "Marseille",

"postal\_code": "13006",

"address\_line1": "175 Rue de Rome",

"organisation": false

}

},

"package": {

"dimensions": {

"height": 10,

"width": 10,

"length": 10,

"unit": "cm",

},

"grossWeight": {

"amount": 100,

"unit":kg,

}

}

}

}

}

**Shipment not found:**

{

"data": {

"ref" : ""

}

}

## Process

* Using “ref” number to query the shipment, if the shipment exists, return format as **4.2**
* “ref” field is required

# deleteShipment

## Request

POST /client/deleteshipment HTTP/1.1

{

"data": {

"ref": "\_\_REFERENCE\_NUMBER", // Use to lookup the shipment

}

}

## Response

{

"data": [

{

"status": "OK”,

"message": "shipment has been deleted",

}]

}

## Process

* Using “ref” number to query the shipment, if the shipment exists, return format as 4.2
* “ref” field is required
* Status:
  + “OK” if the shipment is deleted successfully
  + “NOK” if the shipment is not found or if there is any error
* Message:
  + Delete shipment successfully: “Shipment has been deleted”
  + Cannot found the shipment: “Shipment not found”

# trackShipment

## Request

**OUT OF SCOPE**

## Response

**OUT OF SCOPE**