

Objective

Create a new CRUD API for Customers data with

some fields like `customer_id`, `customer_name`, `customer_email` etc.

Create methods for adding a new customer, listing all the customers, updating a customer and deleting a customer.

Features Steps

It was new thing for me making my own command. it's kinda different using json array after using database for a while.

Source Code

Commands

```

83 //-----
84 // node app.js add -i <id> -f <first name> -l <last name> -e <email>
85 //-----
86 if (command === 'add'){
87     var customer = customers.addCustomer(argv.id,argv.firstName,argv.lastName,argv.email);
88     if (customer){
89         customers.logCustomer(customer);
90     }
91 }
92 //-----
93 //node app.js list
94 //-----
95 else if (command === 'list') {
96     var AllCustomers = customers.getAll();
97     console.log(`Printing ${AllCustomers.length} customers.`);
98     AllCustomers.forEach( callbackfn: (customer)=>{
99         customers.logCustomer(customer);
100     });
101 }
102 //-----
103 //node app.js remove -d <id, first, last, email> -a <attribute to delete by>
104 //-----
105 else if (command === 'remove') {
106     var deleted = customers.removeCustomer(argv.detail,argv.value);
107     if (deleted)
108         console.log('Unable to remove customer!');
109     else
110         console.log('Customer has been removed.');
```

```

112 //-----
113 // node app.js update --d1 <[id]> --v1 <[attribute of the id]> -d <[id or first or last or email]> -v <[attribute of
114 //the id or first or last or email]>
115 //-----
116 else if (command === 'update') {
117     var update = customers.modifyCustomer(argv.detail1,argv.value1,argv.detail,argv.value);
118     if (update)
119         console.log('Updating Customers placeholder.');
```

Get all customers using fetchCustomers

```
1  const fs = require('fs');
2  const _ = require('lodash');
3  //-----
4  //Parse through customer-data.json file
5  //-----
6  var fetchCustomers = () => {
7    try { //if file won't exist
8      var customerString = fs.readFileSync('customer-data.json');
9      return JSON.parse(customerString);
10   } catch(e){
11     return [];
12   }
13 };
14 //-----
15 // Update customer-data.json file
16 //-----
17 var saveCustomers = (customer) => {
18   fs.writeFileSync('customer-data.json', JSON.stringify(customer));
19 };
```

```
41 //-----  
42 // Log customer details to console  
43 //-----  
44 var logCustomer = (customer) => {  
45   console.log('--');  
46   console.log(`ID: ${customer.id}`);  
47   console.log(`First Name: ${customer.firstName}`);  
48   console.log(`Last Name: ${customer.lastName}`);  
49   console.log(`Email: ${customer.email}`);  
50 };  
51 //-----  
52 // Get all customers using fetchCustomers  
53 //-----  
54 var getAll = () => {  
55   return fetchCustomers();  
56 };
```

update a customer

```

88 // update a customer by detail [id] and attribute
89 //-----
90 var modifyCustomer = (idd,val1,det,val) => {
91     var customers = fetchCustomers();
92
93     if (idd === "id") {
94
95         var index = customers.findIndex(function (item, i) {
96             return item.id === val1
97         });
98         console.log(index);
99
100         if(index > -1){
101
102             if (det === "id") {
103                 customers[index].id = val;
104             }
105             else if (det === "first") {
106                 customers[index].firstName = val;
107             }
108             else if (det === "last") {
109                 customers[index].lastName = val;
110             }
111             else if (det === "email") {
112                 customers[index].email = val;
113             }else {
114                 console.log('Detail must be id, first, last, or email');
115                 return false;
116             }
117
118             console.log(index);
119             saveCustomers(customers);
120         }else{
121
122             console.log('No Customer available for Id' + index);

```

Adding a new customer

```
20 //-----
21 // Adding a new customer
22 //-----
23 var addCustomer = (id,firstName,lastName,email) => {
24   var customers = fetchCustomers();
25   var customer = {id,firstName,lastName,email};
26
27   var duplicateCustomers = customers.filter((customer) => {
28     return customer.id === id;
29     return customer.firstName === firstName;
30     return customer.lastName === lastName;
31     return customer.email === email;
32   });
33   if (duplicateCustomers.length === 0){
34     customers.push(customer);
35     saveCustomers(customers);
36     return customer
37   } else {
38     console.log(`Customer ID:${customer.id} already exists!`);
39   }
40 };
```

Remove a customer

```

57 //-----
58 // Remove a customer by detail [id, first name, last name, email] and attribute
59 //-----
60 var removeCustomer = (det,val) => {
61     var customers = fetchCustomers();
62     var updtCustomers = '';
63
64     if (det === "id") {
65         updtCustomers = customers.filter((customer) => customer.id !== val);
66         saveCustomers(updtCustomers);
67     }
68     else if (det === "first") {
69         updtCustomers = customers.filter((customer) => customer.firstName !== val);
70         saveCustomers(updtCustomers);
71     }
72     else if (det === "last") {
73         updtCustomers = customers.filter((customer) => customer.lastName !== val);
74         saveCustomers(updtCustomers);
75     }
76     else if (det === "email") {
77         updtCustomers = customers.filter((customer) => customer.email !== val);
78         saveCustomers(updtCustomers);
79     }
80     else
81         console.log('Detail must be id, first, last, or e');
82
83     return customers.length === updtCustomers.length;
84
85 };
86

```

Limitations

it is Json array not a database no primary key