Cust.js

**const** fs = require(**'fs'**);  
**const** \_ = require(**'lodash'**);

**const notes1** = require(**'./notes1.js'**);  
  
  
**const** yargs = require(**'yargs'**);  
  
  
*// ------------ Begin - command configuration -----------------***const** customer\_idOptions = {  
 **describe**: **'Customer\_ID'**,  
 **demand** : **true**,  
 **alias** : **'a'**}  
  
**const** customer\_nameOptions = {  
 **describe**: **'Customer\_Name'**,  
 **demand** : **true**,  
 **alias** : **'b'**}  
  
**const** customer\_emailOptions = {  
 **describe**: **'Customer\_Email'**,  
 **demand** : **true**,  
 **alias** : **'c'**}  
**const** argv = yargs  
  
 .command(**'add'**,**'Add a new note'**,{  
 **customer\_id**: customer\_idOptions,  
 **customer\_name**: customer\_nameOptions,  
 **customer\_email**: customer\_emailOptions  
 })

.command(**'read'**,**'Read a note'**,{  
 **customer\_id**: customer\_idOptions  
 })  
 .command(**'list'**,**'List all notes'**)

.command(**'update'**, **'Update a Note'**,{  
 **customer\_id**: customer\_idOptions,  
 **customer\_name**: customer\_nameOptions,  
 **customer\_email**: customer\_emailOptions  
 })  
  
 .command(**'remove'**,**'Remove a Note'**,{  
 **customer\_id**: customer\_idOptions  
 })  
 .**help**()  
 .**argv**;  
  
  
*// ------------ End - command configuration -----------------***var** command = argv.**\_**[0];  
  
  
**if** (command === **'add'**){  
 **var** note = **notes1**.*addNote*(argv.**customer\_id**,argv.**customer\_name**,argv.**customer\_email**);  
 **if** (note){  
 **notes1**.*logNote*(note); *//add a new note* } **else**{  
 ***console***.log(**"Note already exists"**);  
 }  
}  
  
**else if** (command === **'list'**) {  
 **var** AllNotes1 = **notes1**.*getAll*();  
 ***console***.log(**`Printing** ${AllNotes1.**length**} **note(s).`**);  
 AllNotes1.forEach((note)=>{ *//list all note(s)* **notes1**.*logNote*(note);  
 });  
}  
  
**else if** (command === **'read'**) {  
 **var** note = **notes1**.*getNote*(argv.**customer\_id**);  
 **if**(note){  
 **notes1**.*logNote*(note); *//read a note* }  
 **else**{  
 ***console***.log(**"Note not found"**);  
 }  
}

**if** (command === **'update'**){  
 **var** note = **notes1**.*remove*(argv.**customer\_id**);  
  
 **var** note = **notes1**.*addNote*(argv.**customer\_id**,argv.**customer\_name**,argv.**customer\_email**);  
 **if** (note){  
 **notes1**.*logNote*(note); } **else**{  
 ***console***.log(**"Note already exists"**);  
 }  
}  
  
**else**{  
 ***console***.log(**'command note recognized'**);  
}

**else if** (command === **'remove'**) {  
 **var** note = **notes1**.*remove*(argv.**customer\_id**);  
 **if**(note){  
 **notes1**.*logNote*(note); }  
 **else**{  
 ***console***.log(**"Note not found"**);  
 }  
}

notes1.js

**const** fs = require(**'fs'**);  
  
*// ------------------Begin of Reusable functions ---------------------***var** *fetchNotes1* = () => {  
 **try** { *//if file won't exist* **var** notes1String = fs.readFileSync(**'notes1-data.json'**)  
 **return *JSON***.parse(notes1String);  
 } **catch**(e){  
 **return** [];  
 }  
};  
  
**var** *saveNotes1* = (notes1) => {  
 fs.writeFileSync(**'notes1-data.json'**,***JSON***.stringify(notes1));  
};  
  
  
*// ------------------End of Reusable functions ---------------------  
  
  
// to add a new note***var** *addNote* = (customer\_id,customer\_name,customer\_email) => {  
 **var** notes1 = *fetchNotes1*();  
 **var** note = {customer\_id,customer\_name,customer\_email}  
  
 **var** duplicateNotes = notes1.filter((note) => { *// to check if note already exists* **return** note.**customer\_id** === customer\_id;  
 });  
  
 **if** (duplicateNotes.**length** === 0){  
 notes1.push(note);  
 *saveNotes1*(notes1);  
 **return** note  
 }  
  
};  
  
  
*//to list all the notes***var** *getAll* = () => {  
 **return** *fetchNotes1*();  
};  
  
  
*// to read a note***var** *getNote* = (customer\_id) => {  
  
 **var** notes1 = *fetchNotes1*();  
  
 **var** getNotes = notes1.filter((note) => { *// to check if note exists and return note* **return** note.**customer\_id** === customer\_id;  
 });  
  
 **return** getNotes[0]  
  
};  
  
  
*// to delete a note***var** *remove* = (customer\_id) => {  
  
 **var** notes1 = *fetchNotes1*(); *// reusable func* **var** filteredNotes = notes1.filter((note) => { *// will return all other notes other than "note to be removed"* **return** note.**customer\_id** !== customer\_id;  
 });  
  
 *saveNotes1*(filteredNotes); *//save new notes array* **return** notes1.**length** !== filteredNotes.**length**};  
  
**var** *updateNote* = (customer\_id,customer\_name,customer\_email) => {  
 **var** notes1 = *fetchNotes1*();  
 **var** note = {customer\_id,customer\_name,customer\_email}  
  
 **var** duplicateNotes = notes1.filter((note) => { *// to check if note already exists* **return** note.**customer\_id** === customer\_id;  
 });  
  
 **if** (duplicateNotes.**length** !== 0){  
 notes1.push(note);  
 *saveNotes1*(notes1);  
 **return** note  
 }  
  
};

notes1-data.json

[{**"customer\_id"**:1,**"customer\_name"**:**"gopi"**,**"customer\_email"**:**"gopi@gmail.com"**},{**"customer\_id"**:2,**"customer\_name"**:**"chand"**,**"customer\_email"**:**"chand@gmail.com"**},{**"customer\_id"**:**"aaa"**,**"customer\_name"**:**"abc"**,**"customer\_email"**:**"abc"**}]