Advanced Software Engineering

Tutorial 6

Team 1\_2

**Student 1**

Name: Srichakradhar Reddy Nagireddy

Class ID: 32

**Student 2**

Name: Sathwik Banginimatam

Class ID: 5

**CRUD API for Customers data**

Yargs & express

This is a CRUD API for Customers data with some fields like customer\_id, customer\_name, customer\_email etc., with methods for adding a new customer, listing all the customers, updating a customer and deleting a customer.

Source Code

# Command Configuration for Yargs

const idOptions = {

describe: 'ID of customer',

demand: true,

alias: 'i'

}

const nameOptions = {

describe: 'Name of the customer',

demand: true,

alias: 'n'

}

const emailOptions = {

describe: 'e-Mail of the customer',

demand: true,

alias: 'e'

}

const argv = yargs

.command('add', 'Add a new customer', {

id: idOptions,

name: nameOptions,

email: emailOptions

})

.command('update', 'Update an existing customer', {

id: idOptions,

name: nameOptions,

email: emailOptions

})

.command('list', 'List all customers')

.command('read', 'Read a customer', {

id: idOptions

})

.command('remove', 'Remove a Customer', {

id: idOptions

})

.help()

.argv;

// ------------ End - command configuration -----------------

var command = yargs.argv.\_[0];

# Functions to manage customer data

function add(c){

var data = command ? argv : c;

var customer = customers.addCustomer(data.id, data.name, data.email);

if (customer) {

customers.logCustomer(customer); //add a new customer

return "Customer " + data.id + " added.";

} else {

var message = "Customer with id " + data.id + " already exists";

console.log(message);

return message;

}

}

function update(c){

var data = command ? argv : c;

var updateCount = customers.updateCustomer(parseInt(data.id), data.name, data.email);

var message;

if (updateCount > 0) {

message = updateCount + " customer(s) updated.";

} else {

message = "Customer doesn't exist.";

}

console.log(message);

return message;

}

function list(){

var AllCustomers = customers.getAll();

console.log(`Printing ${AllCustomers.length} customer(s).`);

AllCustomers.forEach((customer) => { //list all customer(s)

customers.logCustomer(customer);

});

return AllCustomers;

}

function read(id){

var customerId = command ? argv.id : id;

var customer = customers.getCustomer(parseInt(customerId));

if (customer) {

customers.logCustomer(customer); //read a customer

return customer;

}

else {

var message = "Customer not found";

console.log(message);

return message;

}

}

function remove(customer){

var noteRemoved = customers.remove(command ? argv.id : parseInt(customer.id));

var message = noteRemoved ? 'Customer was removed' : 'Customer not found';

console.log(message);

return message;

}

switch(command){

case 'add': add(); break;

case 'update': update(); break;

case 'list': list(); break;

case 'read': read(); break;

case 'remove': remove(); break;

default: app.listen(port, () => console.log(`Example app listening on port ${port}!`))

}

# Express methods for CRUD API

// ----------------------- Express App -----------------

// List

app.get('/', (req, res) => {

res.status(200);

res.send(list());

});

// Create

app.post('/', (req, res) => {

var customer = req.body;

console.log(customer);

res.status(200);

res.send(add(customer));

});

// Read

app.get('/:id', (req, res) => {

res.status(200);

res.send(read(req.params.id));

});

// Update

app.put('/:id', (req, res) => {

var customer = req.body;

console.log(customer);

customer.id = req.params.id;

res.status(200);

res.send(update(customer));

});

// Delete

app.delete('/:id', (req, res) => {

var customer = req.body;

console.log(customer);

customer.id = req.params.id;

res.status(200);

res.send(remove(customer));

});