Exercise (Instructions): Callback Hell and Promises

Objectives and Outcomes

In this exercise you will learn to use promise support to update the Node application to avoid callback hell. At the end of this exercise you will be able to:

- Avoid callback hell by using promises

Using Promises

- Update *operations.js* as follows:

```
const assert = require('assert');

exports.insertDocument = (db, document, collection, callback) => {
    const coll = db.collection(collection);
    return coll.insert(document);
};

exports.findDocuments = (db, collection, callback) => {
    const coll = db.collection(collection);
    return coll.find({}).toArray();
};

exports.removeDocument = (db, document, collection, callback) => {
    const coll = db.collection(collection);
    return coll.deleteOne(document);
};

exports.updateDocument = (db, document, update, collection, callback) => {
    const coll = db.collection(collection);
    return coll.updateOne(document, { $set: update }, null);
};

```

- Next open index.js and update it as follows:

```
. . .

MongoClient.connect(url).then((client) => {

    console.log('Connected correctly to server');
    const db = client.db(dbname);

    dboper.insertDocument(db, { name: "Vadonut", description: "Test"},
        "dishes")
        .then((result) => {
            console.log("Insert Document:\n", result.ops);

            return dboper.findDocuments(db, "dishes");
        })
        .then((docs) => {
            console.log("Found Documents:\n", docs);

            return dboper.updateDocument(db, { name: "Vadonut" },
                    { description: "Updated Test" }, "dishes");

        })
        .then((result) => {
            console.log("Updated Document:\n", result.result);

            return dboper.findDocuments(db, "dishes");
        })
        .then((docs) => {
            console.log("Found Updated Documents:\n", docs);

            return db.dropCollection("dishes");
        })
        .then((result) => {
            console.log("Dropped Collection: ", result);

            return client.close();
        })
        .catch((err) => console.log(err));

})
.catch((err) => console.log(err));
```

- Run the node application and see the result
- Do a Git commit with the message "Node Callback Hell and Promises".

Conclusions

In this exercise you learnt to use promises in a Node application and avoid callback hell.

Mark as completed