Getting started with MongoDB and Typescript

This exercise is backed up by a video with three sections. See links below for what to watch in parallel with individual parts of the exercise

Prerequisites

An account on Atlas setup to allow remote access from all IP's and with a database-user and password you can use for all future databases.

1 Setup for this exercise (should take less than 3-4 min)

Watch the first 7-8 minutes of this video as an introduction to what you have to do

1) Create a folder **`mongotypescript1`**

2) In this folder open a terminal af execute the following statements

```
npm init -y
tsc -init
npm install dotenv
npm install mongodb
npm install @types/mongodb --save-dev
```

Open vs-code in this folder and

3) Add a file **`.env`** with this content: `CONNECTION="Add your connection string from Atlas"`

4) Open **`tsconfig.ts`** and change target to **ES2017**

5) Add a file **`connect.ts`** and copy this code into the file:

```
import { MongoClient } from 'mongodb';
require("dotenv").config();

export default async function connect() {
  const uri = process.env.CONNECTION
  if (!uri) {
    throw new Error("No Connections string found")
  }
  const client = new MongoClient(uri, { useNewUrlParser: true, useUnifiedTopology: true });
  await client.connect()
  return client;
}
```

6) Add **three** files, **`create.ts`**, **`query.ts`** and **`update.ts`** and paste this code into **EACH file**.

```
import { MongoClient, Db, Collection } from "mongodb"
import connect from "./connect";
import setupTestData from "./setupTestData"

(async function Tester() {
  const client = await connect();
  const db = client.db("day1ex1")
  const collection = db.collection("inventory")
  const status = await setupTestData(collection)

  //Add your play-around code here
  client.close()
})()
```

7) Finally, create a file **`setupTestData.ts`** and paste this code into the file:

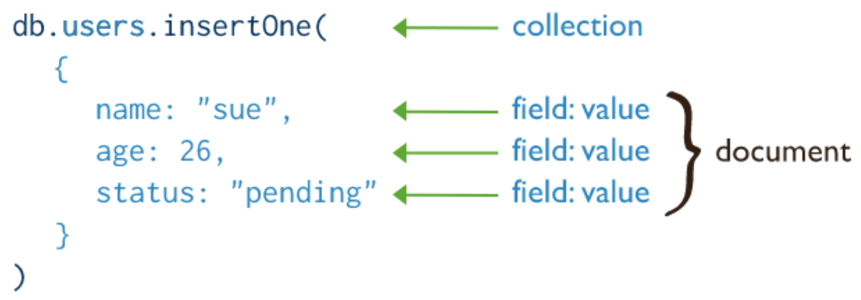
```
import {Collection} from "mongodb"

export default async function setupData(inventoryCollection: Collection) {
  await inventoryCollection.deleteMany({})
  return inventoryCollection.insertMany([
    { item: 'journal',
      qty: 25,
      size: { h: 14, w: 21, uom: 'cm' },
      status: 'A',
      tags: ['blank', 'red'],
      dim_cm: [14, 21],
      instock: [{ warehouse: 'A', qty: 5 }]  },
    { item: 'notebook',
      qty: 50,
      size: { h: 8.5, w: 11, uom: 'in' },
      status: 'A',
      tags: ['red', 'blank'],
      dim_cm: [14, 21],
      instock: [{ warehouse: 'B', qty: 5 }] },
    { item: 'paper',
      qty: 100,
      size: { h: 8.5, w: 11, uom: 'in' },
      status: 'D',
      tags: ['red', 'blank', 'plain'],
      dim_cm: [14, 21],
      instock: [{ warehouse: 'A', qty: 60 }] },
    { item: 'planner',
      qty: 75,
      size: { h: 22.85, w: 30, uom: 'cm' },
      status: 'D',
      tags: ['blank', 'red'],
      dim_cm: [22.85, 30],
      instock: [{ warehouse: 'A', qty: 40 }] },
    { item: 'postcard',
      qty: 45,
      size: { h: 10, w: 15.25, uom: 'cm' },
      status: 'A',
      tags: ['blue'],
      dim_cm: [10, 15.25],
      instock: [{warehouse: 'B', qty: 15 },{ warehouse: 'C', qty: 35 }] },
    { item: 'sketchbook',
      tags: ['black'],
      qty: 80,
      size: { h: 14, w: 21, uom: 'cm' },
      dim_cm: [14, 21],
      status: 'A',
      instock: [{ warehouse: 'A', qty: 23 }] },
    { item: 'sketch pad',
      qty: 95,
      size: { h: 22.85, w: 30.5, uom: 'cm' },
      dim_cm: [22.85, 30],
      status: 'A',
      instock: [{ warehouse: 'D', qty: 7 }]
    }
  ]);
}
```

For all the code samples below, executed via ts-node, which you should have installed with the global option Remember to select NODEJS as your language/platform in the articles below.

2 Create Operations

Watch [this part of the video]() (until the end of the create part)



```
db.users.insertOne(
  {
    name: "sue",
    age: 26,
    status: "pending"
  }
)
```

Read the section *Create Operations* in [MongoDB CRUD Operations](), and code along in the document **create.ts**

Make sure to try the examples referred to in the link above:
[https://docs.mongodb.com/manual/tutorial/insert-documents/]() provided by the document

3 Read Operations

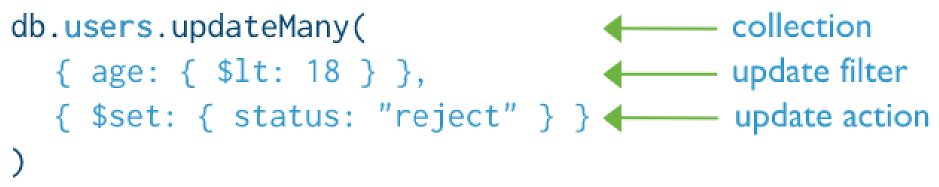
Watch [the last part of the video]() as supplement to this part

Read the section *Read Operations* in [MongoDB CRUD Operations](), and code along in the document **read.ts**

Try out most of the examples given in these sub sections

- [https://docs.mongodb.com/manual/tutorial/query-documents/]()
- [https://docs.mongodb.com/manual/tutorial/query-arrays/]()
- [https://docs.mongodb.com/manual/tutorial/project-fields-from-query-results/]()

4 Update Operations



```
db.users.updateMany(
  { age: { $lt: 18 } },
  { $set: { status: "reject" } }
)
```

Read the section *Update Operations* in [MongoDB CRUD Operations](), and code along in the document **update.ts**

Try out most of the examples given in this subsection:
[https://docs.mongodb.com/manual/tutorial/update-documents/]()