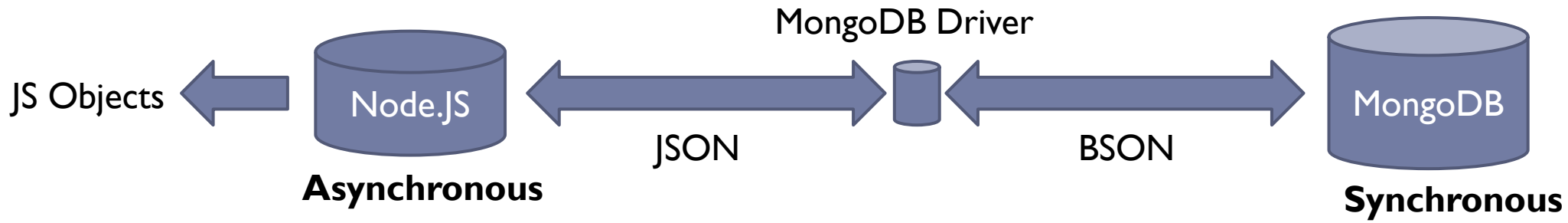
MongoDB Client – Intro & CRUD

MongoDB Driver

- A library written in JS to handle the communication, open sockets, handle errors and talk with MongoDB Server.

```
npm install mongodb
```

- Note that Mongo Shell is **Synchronous** while Node.JS is **Asynchronous**.



Connect to MongoDB

```
const { MongoClient } = require('mongodb');

// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);

// Database Name
const dbName = 'onlineshopping';

async function main() {
    // Use connect method to connect to the server
    await client.connect();
    console.log('Connected successfully to server');
    const db = client.db(dbName);
    const collection = db.collection('products');

    const result = await collection.find({}).toArray();
    return result;
}
```

```
main()
    .then(console.log)
    .catch(console.error)
    .finally(() => client.close());
```

Example - Using findOne()

```
const { MongoClient } = require('mongodb');

// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);

// Database Name
const dbName = 'onlineshopping';

async function main() {
    // Use connect method to connect to the server
    await client.connect();
    console.log('Connected successfully to server');
    const db = client.db(dbName);
    const collection = db.collection('products');

    const result = await collection.findOne({"description": "Good"});
    return result;
}
```

Example - Using insertOne()

```
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'onlineshopping';

async function main() {
    // Use connect method to connect to the server
    await client.connect();
    console.log('Connected successfully to server');
    const db = client.db(dbName);
    const collection = db.collection('products');

    return await collection.insertOne({"title": "Acer", "price": 400, "description":
"Awesome"});
}
```

Example - Using insert() multiple docs

```
//other code is omitted above

async function main() {
    // Use connect method to connect to the server
    await client.connect();
    console.log('Connected successfully to server');
    const db = client.db(dbName);
    const collection = db.collection('products');

    const insertResult = await collection.insertMany([
        {"title": "Acer 2", "price": 402, "description": "Awesome 2"},
        {"title": "Acer 3", "price": 403, "description": "Awesome 3"},
        {"title": "Acer 4", "price": 404, "description": "Awesome 4"}
    ]);
    return insertResult;
}
```

Example - Using update()

```
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'onlineshopping';

async function main() {
    // Use connect method to connect to the server
    await client.connect();
    console.log('Connected successfully to server');
    const db = client.db(dbName);
    const collection = db.collection('products');

    const result = await collection.updateOne({ title: "HP" }, { $set: { price: 111,
description: "Good 111" } });
    return result;
}
```

Example - Using deleteOne()

```
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'onlineshopping';

async function main() {
    // Use connect method to connect to the server
    await client.connect();
    console.log('Connected successfully to server');
    const db = client.db(dbName);
    const collection = db.collection('products');

    const result = await collection.deleteOne({title: "Acer 2"});
    return result;
}
```

In Real Application… Like this

models/product.js

```
const { MongoClient } = require('mongodb');
const client = new MongoClient('mongodb://127.0.0.1:27017');

module.exports = class Product {

    static async fetchAll() {
        await client.connect();
        const db = client.db('onlineshopping');
        const collection = db.collection('products');
        return await collection.find({}).toArray();
    }

}
```

Controllers/productController.js

```
exports.getProducts = async (req, res, next) => {
    const products = await Product.fetchAll();
    res.status(200).json(products);
}
```

In Real Application… (First version – refactoring)

util/database.js

```
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'onlineshopping';

module.exports = async function main() {
    // Use connect method to connect to the server
    await client.connect();
    return client.db(dbName);
}
```

models/product.js

```
const mongoConnect = require('../util/database');

static async fetchAll() {
        const db = await getDb();
        const collection = db.collection('products');
        const result = await
collection.find({}).toArray();
        return result;
}
```

In Real Application…

util/database.js

```
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'onlineshopping';
let _db;

exports.mongodbConnect = async function (callback) {
    await client.connect();
    _db = client.db(dbName);
    callback();
}

exports.getDb = function(){
    if(_db){
        return _db;
    } else{
        throw new Error('No Database Found');
    }
}
```

app.js

```
const {mongoConnect} = require('./util/database');
mongoConnect(() => {
    app.listen(3000);
});
```

models/product.js

```
const {getDb} = require('../util/database');

class Product {
…
   static async fetchAll() {
        const db = await getDb();
        const collection =
db.collection('products');
        const result = await
collection.find({}).toArray();
        return result;
    }
}
```

Resources

- SQL vs NoSQL: https://academind.com/learn/web-dev/sql-vs-nosql/
- Mongo Shell: https://docs.mongodb.com/manual/mongo/
- MongoDB CRUD Operations: https://docs.mongodb.com/manual/crud/
- Node.js MongoDB Driver API: https://mongodb.github.io/node-mongodb-native/3.5/api/

Homework

- Update online shopping application, change CRUD operations on Product Model to use MongoDB.
  - Admin: save/edit/delete product, view all products
  - Shop: view detail of product, view all products