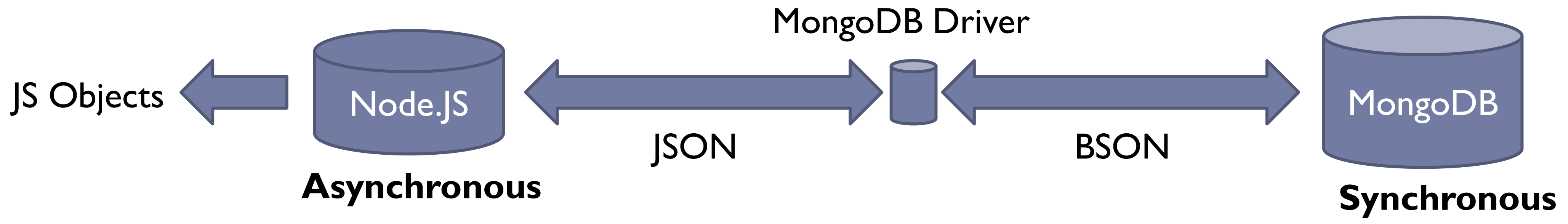
# MongoDB – Intro & CRUD

## 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 – 3.0+

```
const MongoClient = require('mongodb').MongoClient;
MongoClient.connect('mongodb://localhost:27017')
    .then(client => {
        console.log('Connected......');
        const db = client.db('onlineshopping');
        db.collection('products').find().forEach(function(doc) {

            console.log(doc);
            // Close the DB
            client.close();
        })
    })
    .catch(err => console.log('Error: ', err));
```

## Example - Using findOne()

```
const mongodb = require('mongodb');
const MongoClient = mongodb.MongoClient;
MongoClient.connect('mongodb://localhost:27017')
    .then(client => {
        console.log('Connected......');
        const db = client.db('onlineshopping');
        db.collection('products').findOne({ 'title': 'Angular' }, function(err, doc) {
            if (err) throw err;
            // Print the result.
            // Will print a null if there are no documents in the db.
            console.log(doc);
            // Close the DB
            client.close();
        });
    })
    .catch(err => console.log(err));
```

**`console.dir` vs `console.log`**

`console.log()` only prints out a string, whereas `console.dir()` prints out a navigable object tree

## Example - Using insertOne()

```
const MongoClient = require('mongodb').MongoClient;
MongoClient.connect('mongodb://localhost:27017', function(err, client) {
    if (err) throw err;
    const db = client.db('onlineshopping');
    let doc = { title: 'React', price: 29, description: 'This is a React course'
};
    db.collection('products').insertOne(doc, (err, docInserted) => {
        if (err) throw err;
        console.log(docInserted);
        return client.close();
    });
});
```

## Example - Using insert() multiple docs

```
const MongoClient = require('mongodb').MongoClient;
MongoClient.connect('mongodb://localhost:27017', function(err, client) {
    if (err) throw err;
    const db = client.db('onlineshopping');
    const docs = [
        { title: 'SSP', price: 2000, description: 'Server Side Programming' },
        { title: 'AP', price: 1000, description: 'Asynchronous Programming' }
    ];
    db.collection('products').insertMany(docs, (err, docInserted) => {
        if (err) throw err;
        console.log(`Success: ${JSON.stringify(docInserted)}!`);
        return client.close();
    });
});
```

## Example - Using update()

```
const MongoClient = require('mongodb').MongoClient;
MongoClient.connect('mongodb://localhost:27017', function(err, client) {
    if (err) throw err;
    const db = client.db('onlineshopping');

    db.collection('products')
            .updateOne({ title: 'Angular' }, { $set: { updateTime: new Date() } },
                     function(err, data) {
        console.log(data);
        client.close();
    })
});
```

## Example - Using deleteOne()

```
const MongoClient = require('mongodb').MongoClient;
MongoClient.connect('mongodb://localhost:27017', function(err, client) {
    if (err) throw err;
    const db = client.db('onlineshopping');
    var query = { title: 'Angular' };
    // remove all documents that have 'student' value is 'Susie'
    db.collection('products').deleteOne(query, function(err, result) {
        console.log("Result:" + JSON.stringify(result));
        return client.close();
    });
});
```

## In Real Application… Like this?

util/database.js

```
const mongodb = require('mongodb');
const MongoClient = mongodb.MongoClient;

const mongoConnect = (callback) => {
    MongoClient.connect('mongodb://localhost:27017')
        .then(client => {
            console.log('Connected......');
            callback(client);
        })
        .catch(err => console.log(err));
}

module.exports = mongoConnect;
```

models/product.js

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

class Product {
…
 save() {
   mongoConnect((client) => {
       client.db('onlineshopping').collection('products')
                .insertOne(this)
                .then(result => console.log(result))
                .catch(err => console.log(err));
         });
     }}
```

## In Real Application…

util/database.js

```
const mongodb = require('mongodb');
const MongoClient = mongodb.MongoClient;
let _db; //indicate private variable
const mongoConnect = (callback) => {
    MongoClient.connect('mongodb://localhost:27017’,
        { useUnifiedTopology: true })
        .then(client => {
            console.log('Connected......');
            _db = client.db('testCol');
            callback();
        })
        .catch(err => console.log(err));
}
const getDb = () => {
    if (_db) {
        return _db;
    }
    throw new Error('No Database Found!');
}
exports.mongoConnect = mongoConnect;
exports.getDb = getDb;
```

app.js

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

models/product.js

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

class Product {
…
  save() {
    const db = getDb();
    db.collection('products')
      .insertOne(this)
      .then(result => console.log(result))
      .catch(err => console.log(err));
  }
}
```

## 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