server.js (tabs: public.sql, JS server.js ×, JS ProductsMapper.js, JS ProductsService.js, ···, JS app.js ×, JS ProductsRoutes.js, JS ProductsDao)

JS server.js > ...
```js
const app = require("./app");
const db = require("./db/DBConnection");

app.set("port", process.env.PORT || 8080);

app.listen(app.get("port"), () => {
  console.log("Express server started and running!");
});
```

app.js (JS app.js > ...)
```js
const express = require("express");
const app = express();
const ProductsRoutes = require("./routes/ProductsRoutes");

// parse application/json
app.use(express.json());

app.use("/", ProductsRoutes);

module.exports = app;
```

ProductsRoutes.js (tabs: DbConnection.js, JS ProductsRoutes.js ×, database ···)

routes > JS ProductsRoutes.js > ...
```js
const express = require("express");
const router = express.Router();
const ProductsController = require("./../controller/ProductsContr

router.get("/list", ProductsController.list);
// router.post("/add", ProductController.add);
// router.delete("/delete", ProductController.delete);

module.exports = router;
```

ProductsController.js (tabs: JS ProductsController.js ×, JS app.js, JS Pro)

controller > JS ProductsController.js > ...
```js
const Products = require("../services/ProductsServic
function ProductsController() {
  const listProducts = function(req, res) {
    Products.list().then(data => res.json(data));
  };

  //   const addBooks = function(req, res) {
  //     Products.add(req.body).then(data => res.json(
  //   };

  //   const deleteBooks = function(req, res) {
  //     Products.delete(req.param.id).then(data => res
  //   };

  return {
    list: listProducts
    // add: addBooks,
    // delete: deleteBooks
  };
}

module.exports = ProductsController();
```

DbConnection.js (tabs: ProductsService.js, JS DbConnection.js ×)

db > JS DbConnection.js > ...
```js
const Pool = require('pg').Pool
const pool = new Pool({
    user: 'service',
    host: 'localhost',
    database: 'api',
    password: 'password',
    port: 5432,
})

module.exports = pool;
```

ProductsDao.js (tabs: JS ProductsDao.js ×, JS ProductModel.js)

dao > JS ProductsDao.js > ...
```js
const Products = require("../models/ProductModel");
const pool = require("../db/DbConnection");
const ProductsMapper = require('../mapper/ProductsMapper');

function ProductsDao() {
    const find = function () {
        return new Promise((resolve, reject) => {
            pool.query(
                'SELECT * FROM Products_table;',
                (error, results) => {
                    if (error) {
                        throw error;
                    }
                    resolve(ProductsMapper.MapProducts(results.rows))
                }
            );
        })
    };

    return {
        find: find,
        // add: addBooks,
        // delete: deleteBooks
    };
}
// let b = ProductsDao().find()
// console.log(b)
module.exports = ProductsDao();
```

JS ProductsMapper.js ×   JS ProductsService.js   •••   JS ProductModel.js ×

mapper > JS ProductsMapper.js > ƒx ProductsMappers > (x) MapPr   models > JS ProductModel.js > ...

```js
1    const Product = require('../models/ProductModel');
2
3    function ProductsMappers() {
4        const MapProducts = function (rows) {
5            let productsList = []
6            for (i =0;i<rows.length;i++){
7                let product = new Product()
8                product.product_id = rows[i].product_id
9                product.product_name = rows[i].product_name
10                product.price = rows[i].price
11                product.availabilty = rows[i].availabilty
12                product.rating = rows[i].rating
13                productsList.push(product)
14                // console.log(product)
15            }
16            // console.log(productsList)
17            return productsList
18        };
19
20        return {
21            MapProducts: MapProducts,
22        };
23    }
24
25    module.exports = ProductsMappers();
```

```js
1    const Product = class{
2        // init(py) = constructor(js)
3        constructor (product_id, product_name,
4        this.product_id = product_id;
5        this.product_name = product_name;
6        this.price = price ;
7        this.availabilty = availabilty;
8        this.rating = rating
9        }
10    }
11
12    module.exports = Product;
```

```js
1    // const Products = require("../models/ProductsModel")
2    const ProductsDao = require("../dao/ProductsDao");
3
4    function ProductsService() {
5        return {
6            list: () => ProductsDao.find()
7            // add: data => new Book(data).save(),
8            // delete: id => Book.findByIdAndRemove(id)
9        };
10    }
11
12    module.exports = ProductsService();
13
14
```

```sql
1    -- Active: 1657821992135@@127.0.0.1@5432@api@public PostgreSQL
     ▶ Execute
2    CREATE TABLE Products_table(
3        product_id int NOT NULL PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
4        product_name VARCHAR(30), /* data type text not prefered because to optimize memory, */
5        price INTEGER,
6        availabilty SMALLINT,
7        rating SMALLINT
8    );
9
     ▶ Execute
10   INSERT INTO products_table(product_name,price,availabilty,rating) VALUES('Iphone 12',150000,10,10);
11
     ▶ Execute
12   INSERT INTO products_table(product_name,price,availabilty,rating) VALUES('Samsung',100000,9,9);
13
     ▶ Execute
14   INSERT INTO products_table(product_name,price,availabilty,rating) VALUES('Redmo',10000,4,4);
15
```