Javascript Technologies

1. Javascript
2. Require JS
3. Knockout JS
4. OJET (Oracle Javascript Extension Toolkit)
5. Node.js

Software requirement

1. VS Code
2. Node.js

Javascript Fundamentals: let, const, arrow functions, callback functions, event handling, template strings, promises, async/await, fetch, validations

Things to cover

* Template Strings
* Callbacks / Arrow functions
* Event Handling
* Promises
* Async/Await
* Validations
* Fetch

Javascript is mainly used to add effects to the front end applications, it can be used at the backend also to access file-systems, DB, or OS resources.

Front end JS technologies

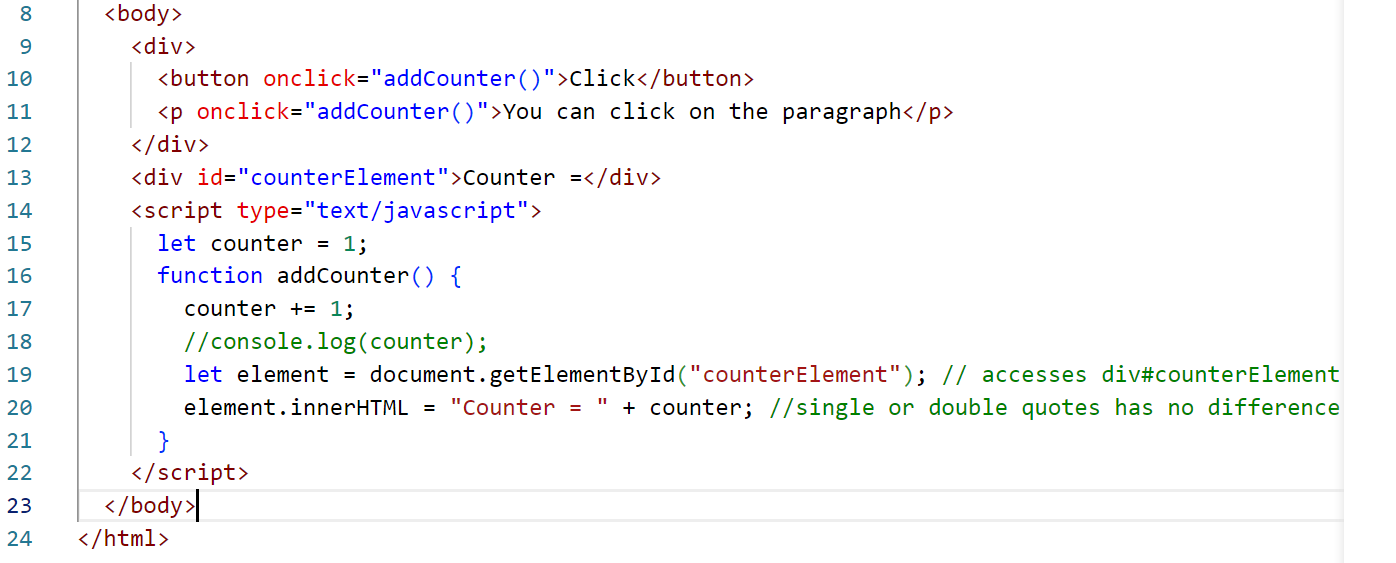
* Typescript
* React.js
* Vue.js
* Angular Framework
* Knockout.js
* OJET

Backend Javascript Technologies

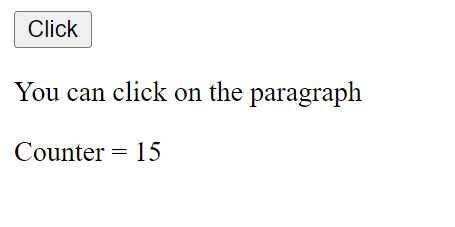
* Node.js: It is a runtime environment for Javascript to executed at the backend without browser

Lab1: Understand event handling

index.html



Output

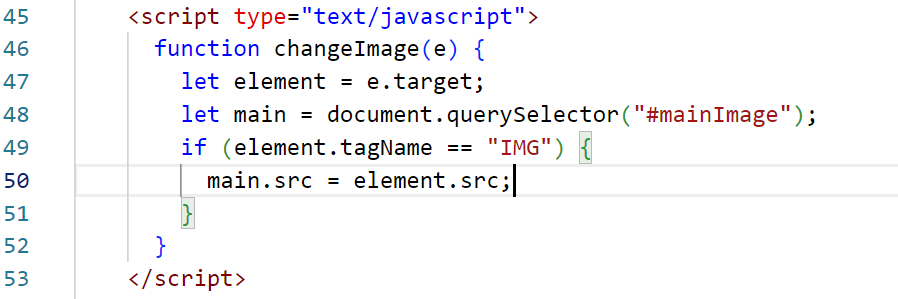


List of ways to access an element

1. document.getElementById(“id”)
2. document.getElementsByTagName(“tag”)
3. document.getElementsByClassName(“class”)
4. document.querySelector(“selector”); #id, .class, tag
5. event.target: To identify which element generated the event

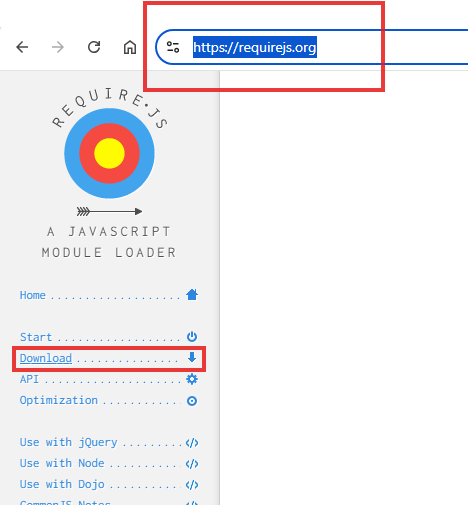
event.target helps you to identify on which element the event is generated





Require JS

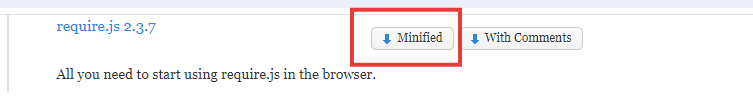
It is a Javascript & module loader, used to increase the speed of your application



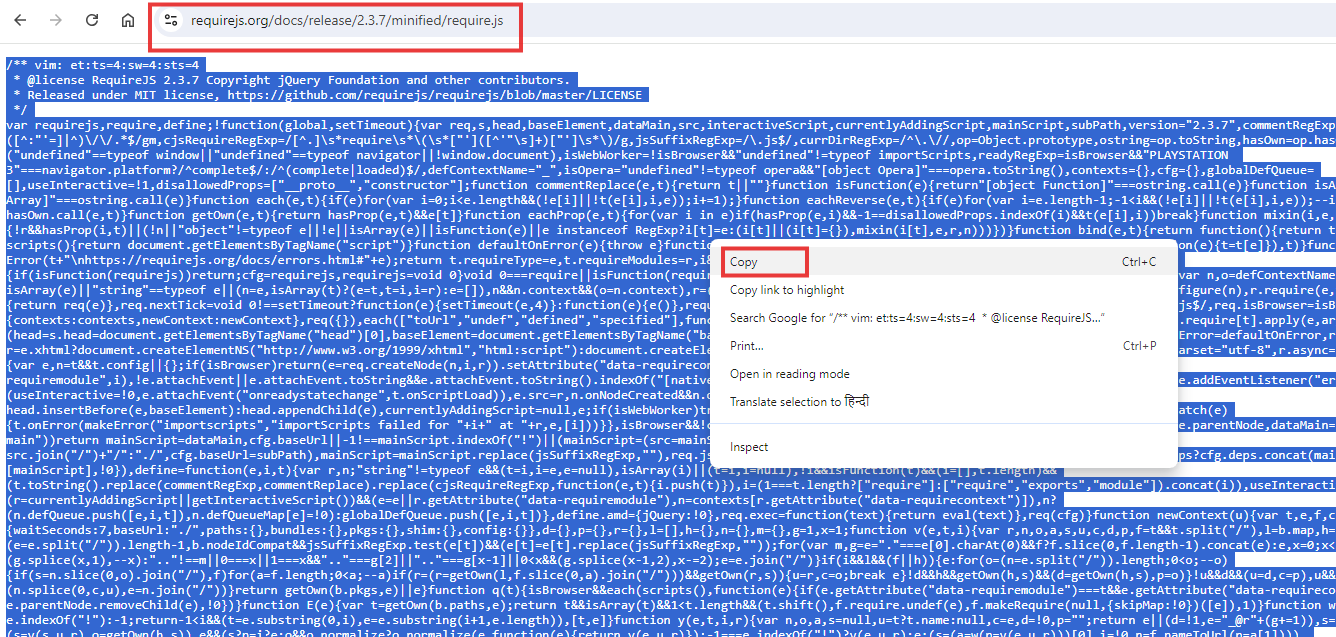
After you click on download select require.js



You need to click on minified button



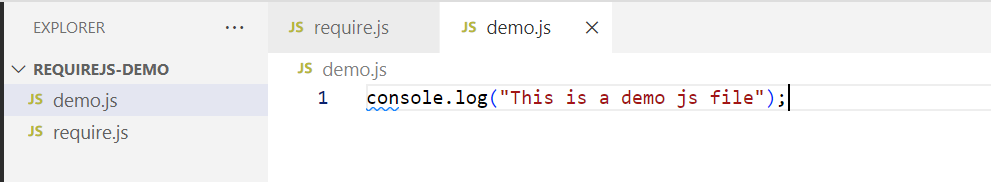
You get a javascript code that you need to copy to one js file



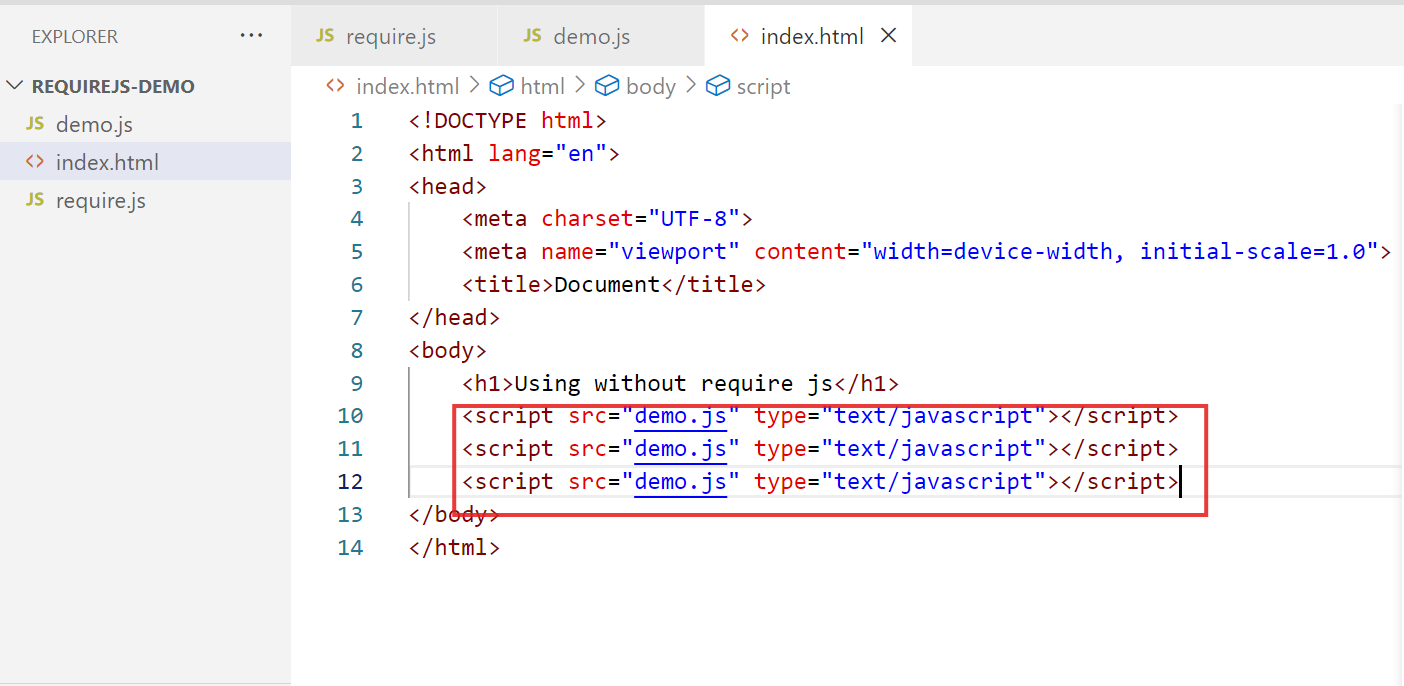
In VS Code you create require.js



Create a demo.js



Create an index.html



Output:



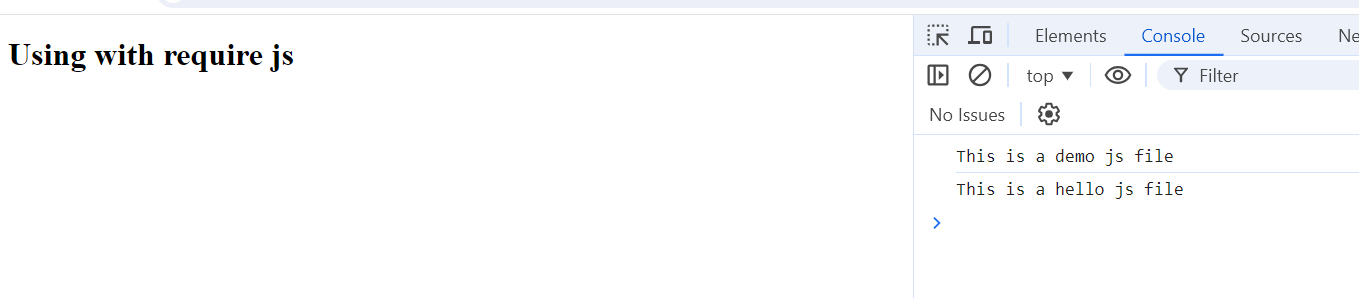
Note: Here the same script is loading more than once, if we use require.js this can be avoided, so that in an application when there are lot of Javascript files you are loading, it should be loaded only once in the browser to increase the speed of the execution.

How to use require js

1. Add the script of require.js
2. All the scripts you load in a require function instead of using script tag



Output:



Notice the require loading the javascript file however not more-than once, even if you repeat

List of technologies that use require

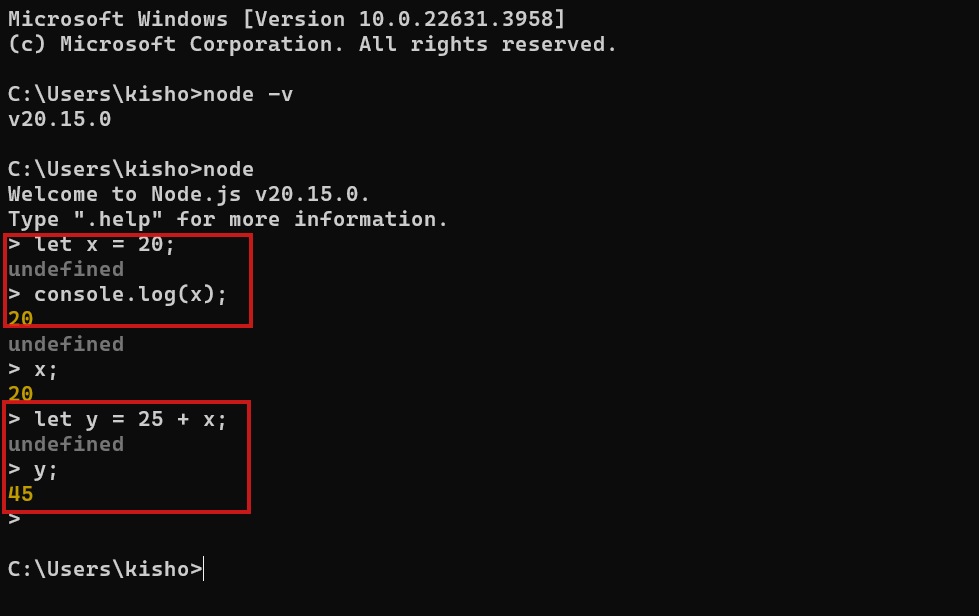
1. Node.js
2. Knockout.js
3. OJET
4. React
5. Angular Framework

Node.js

It is a Javascript runtime environment used to run the javascript code at the backend, so that you can do various operations like

1. creating servers
2. interacting with the OS
3. interacting with the File Systems
4. Networking
5. Interacting with the DB

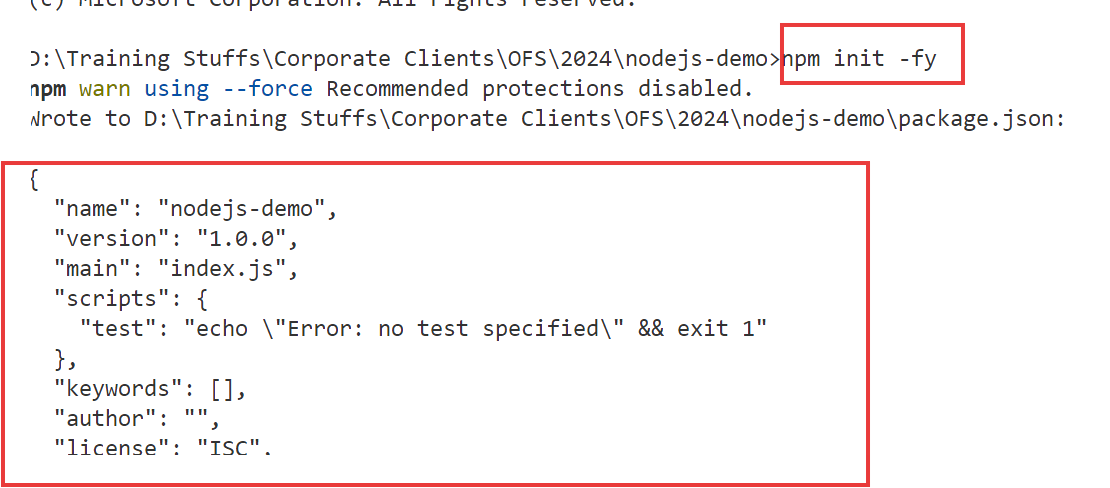
Node provides REPL (Read Eval Print Loop) to quickly write & run JS code



package.json: It is heart of node.js, it will have all the dependencies of the project, project meta-data, scripts to run the project

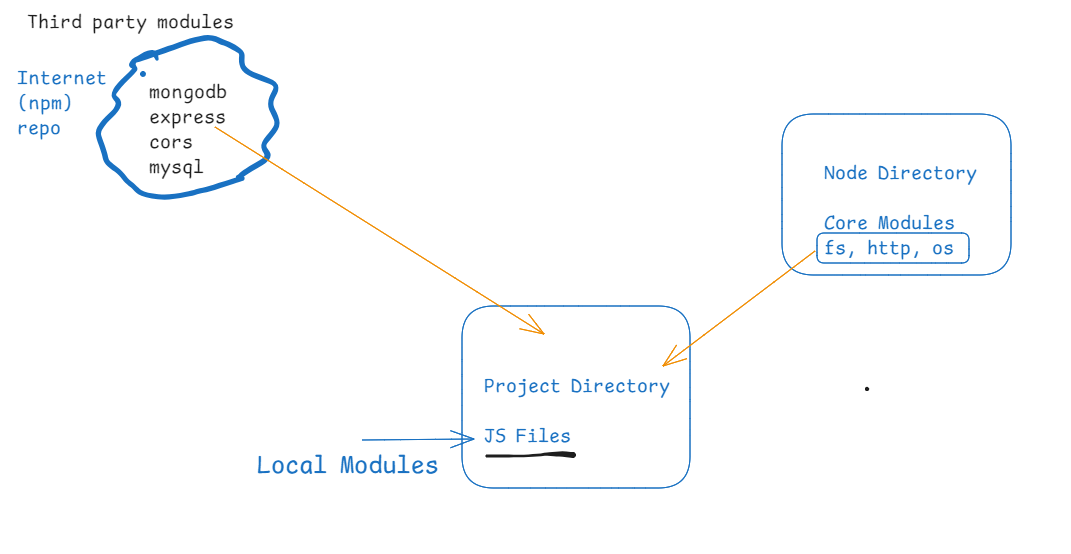
We must use a command to create package.json

npm init -fy



Node.js provides 3 types of libraries / modules

1. Core Module
2. Local Module
3. Third Party Module

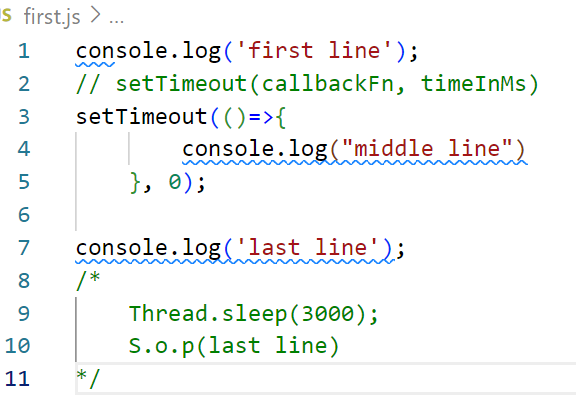


How to download the third party modules

npm install express cors // downloads express & cors libraries

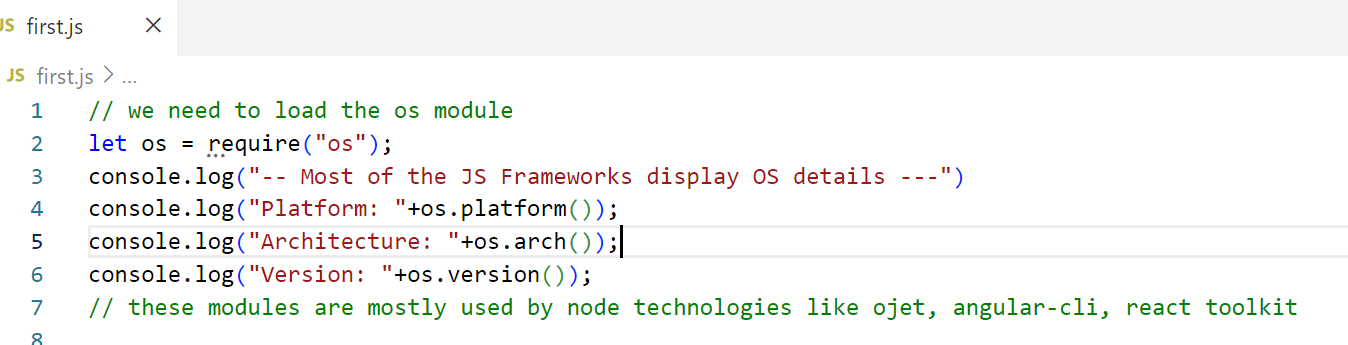
Is require inbuilt in Node.js ? : Yes

let fs = require(‘fs’);  
let os = require(‘os’);



Note: Thread.sleep() is not present in JS, it is present in Java, in Javascript you must use setTimeout for delays.

OS Modules



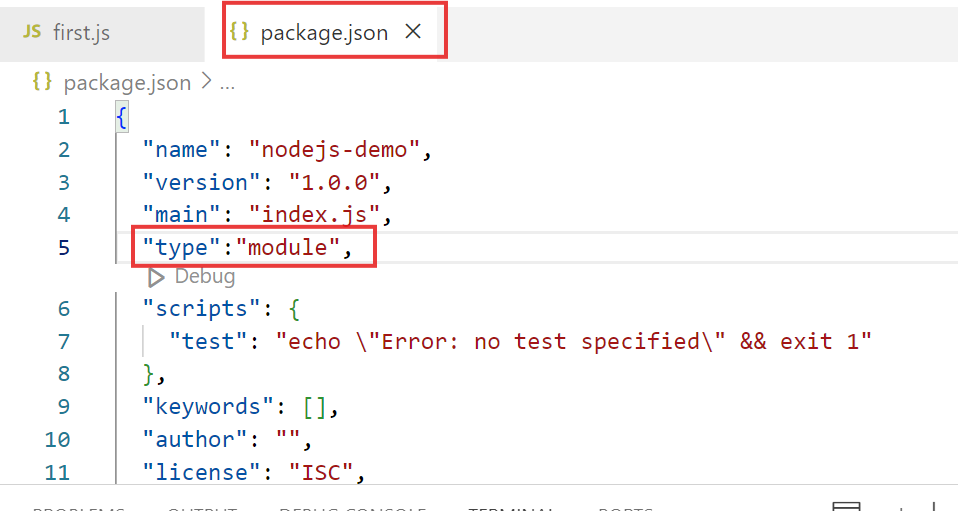
Using modern style import instead of require

ES 5 style:   
let os = require(“os”);

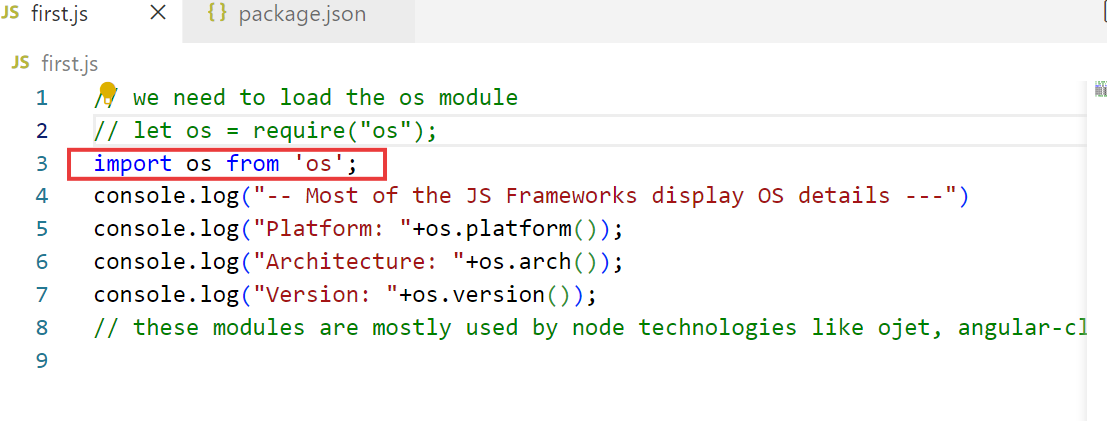
ES 6 style:

import os from ‘os’;

update package.json



Now on you can use import instead of require

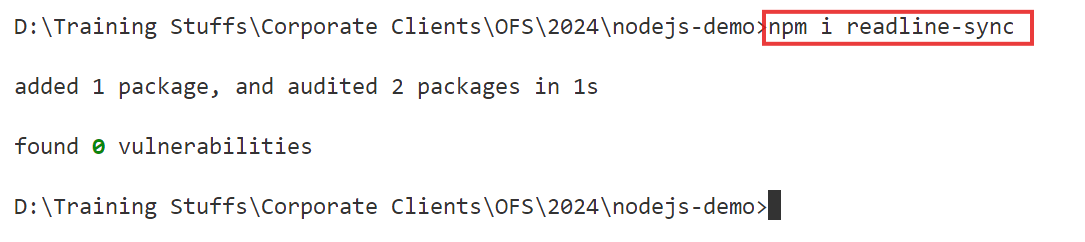


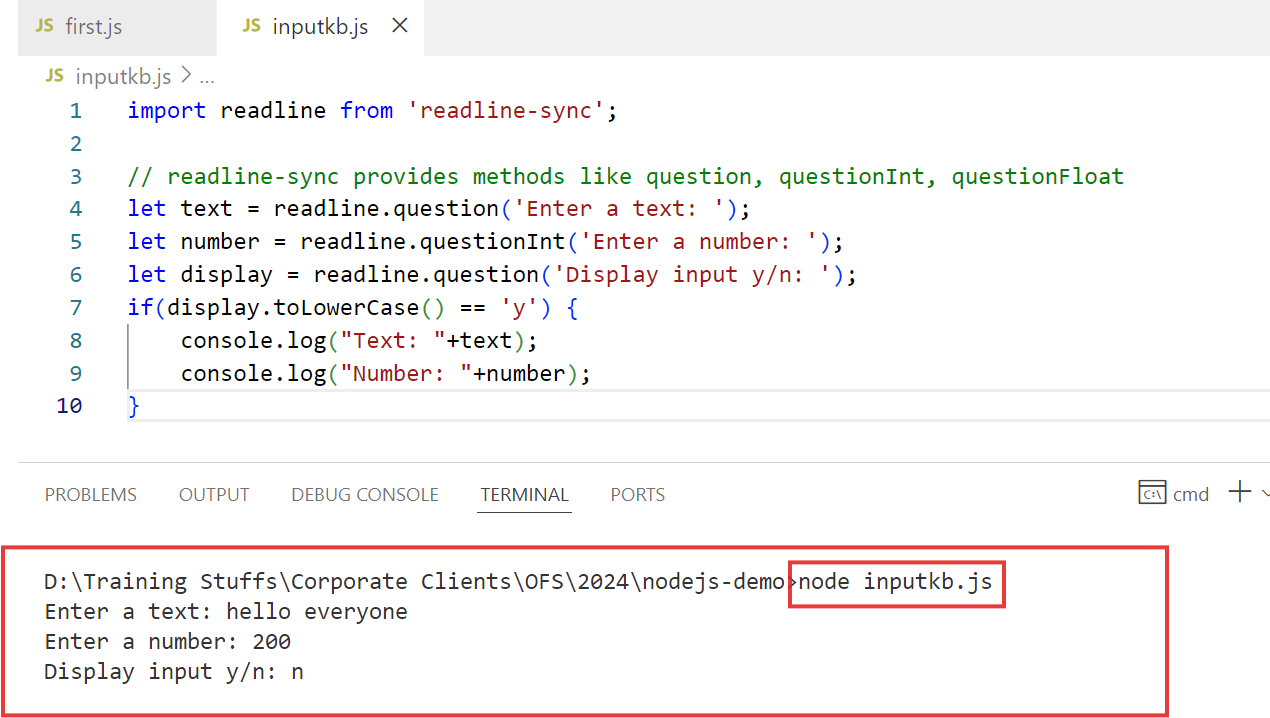
Let us use a third part module

readline-sync: It is used to take input from the keyboard in the terminal

Installing:

npm install readline-sync // we must download this inside the project folder that has package.json

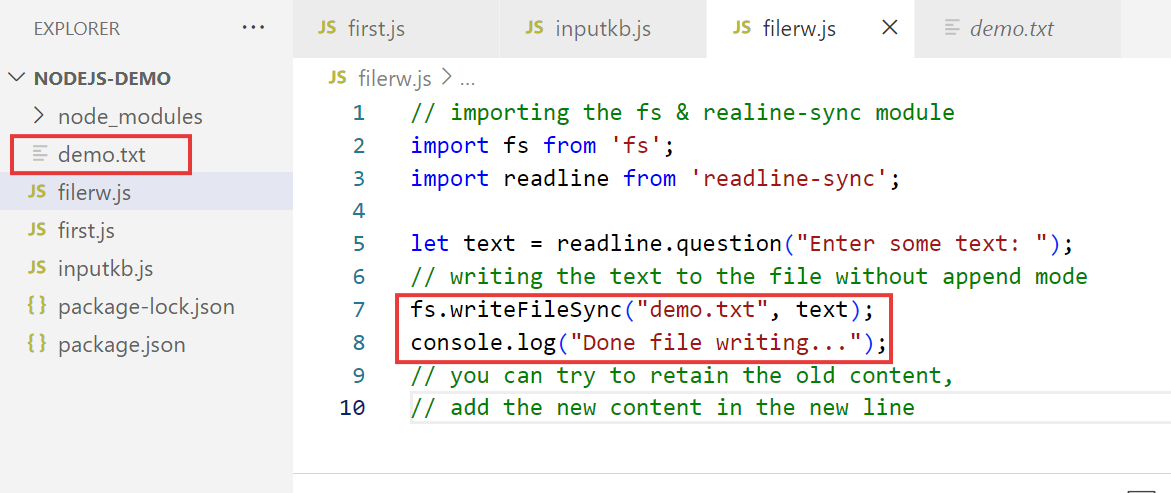




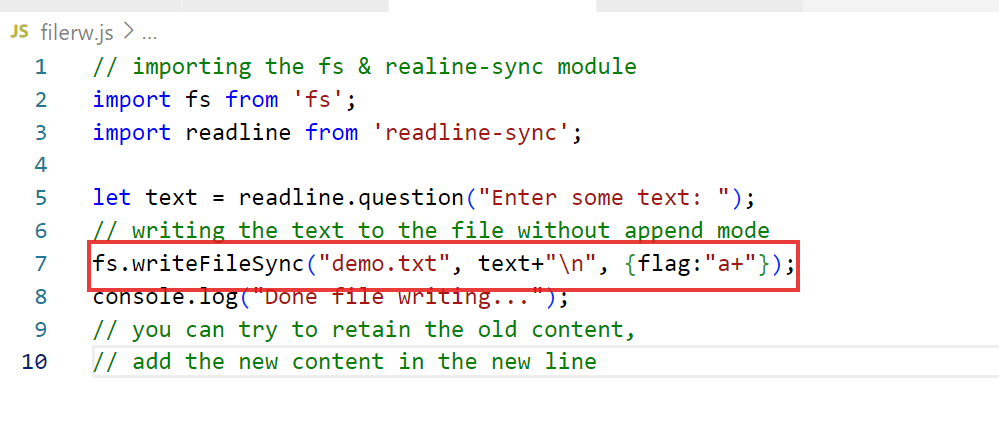
Try different functions like: questionNewPassword(..), questionEmail(…)

fs module: It provides methods to read & write files

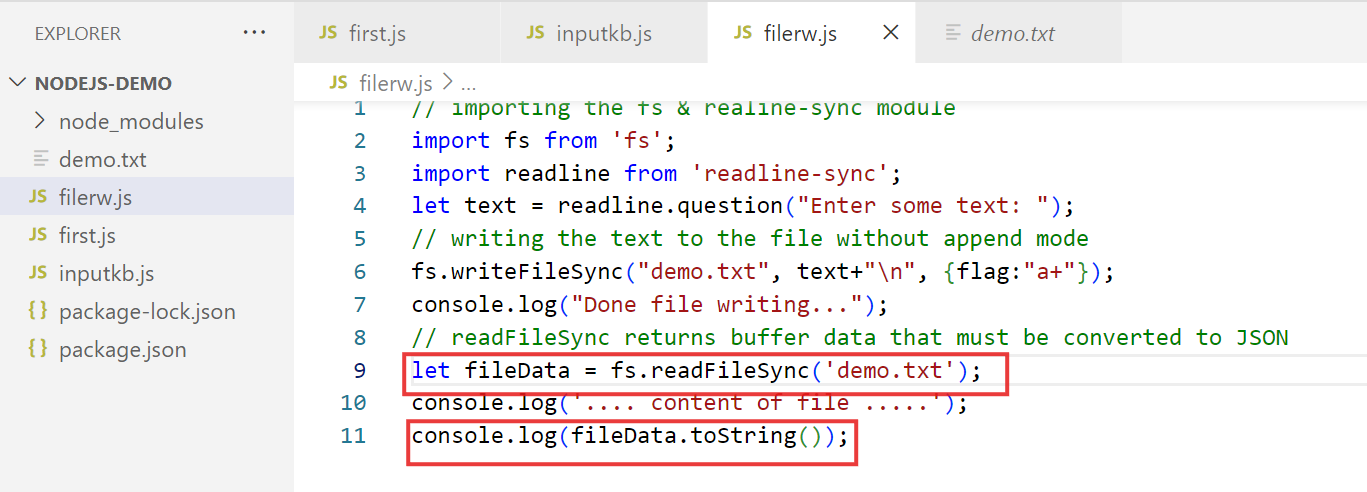
* readFileSync(“filename”)
* writeFileSync(“filename”, content) or writeFileSync(“filename”, content, {flag:”a+”})



appending to the old content



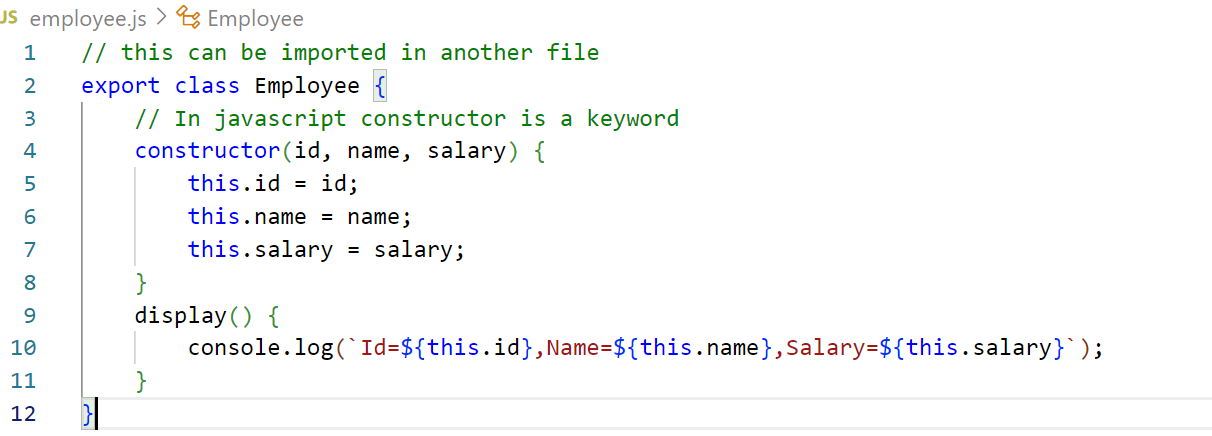
Reading the file content & converting the buffer to string



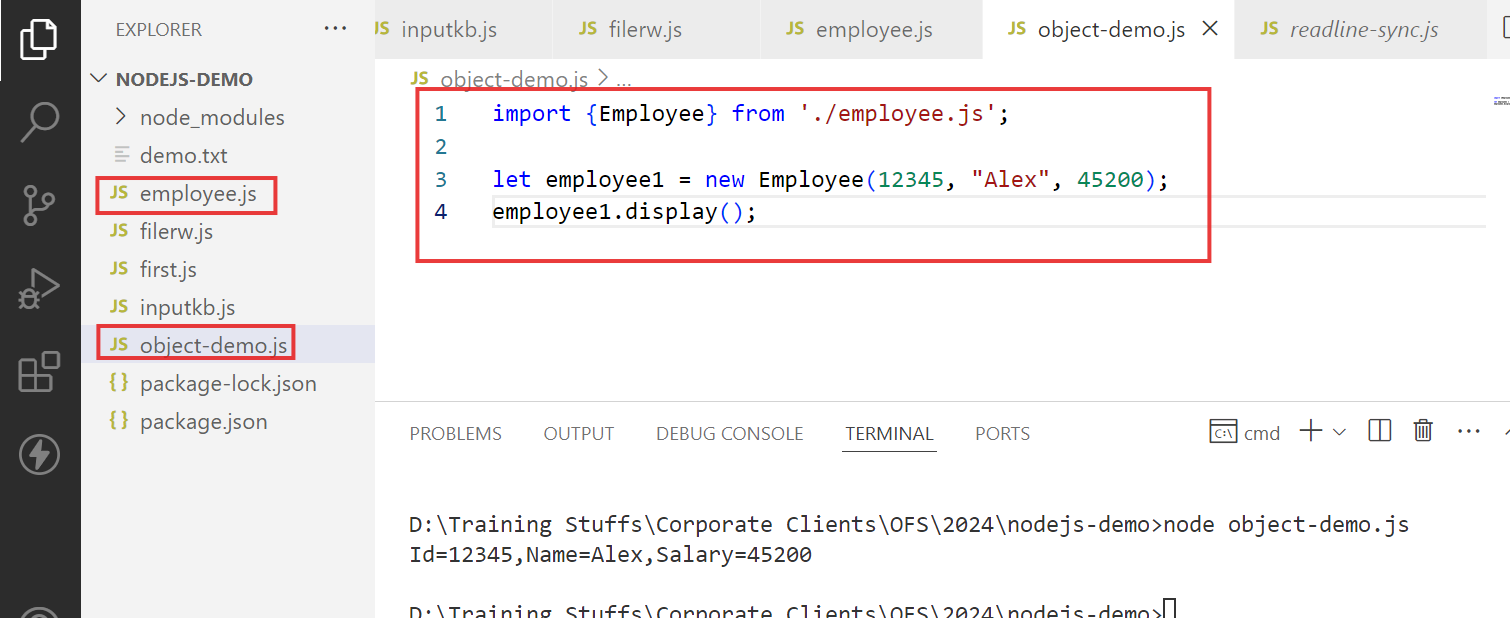
How to import Local module & write an object to a file in the form JSON

We can’t write javascript object to the file, we must convert the object to text i.e., JSON string

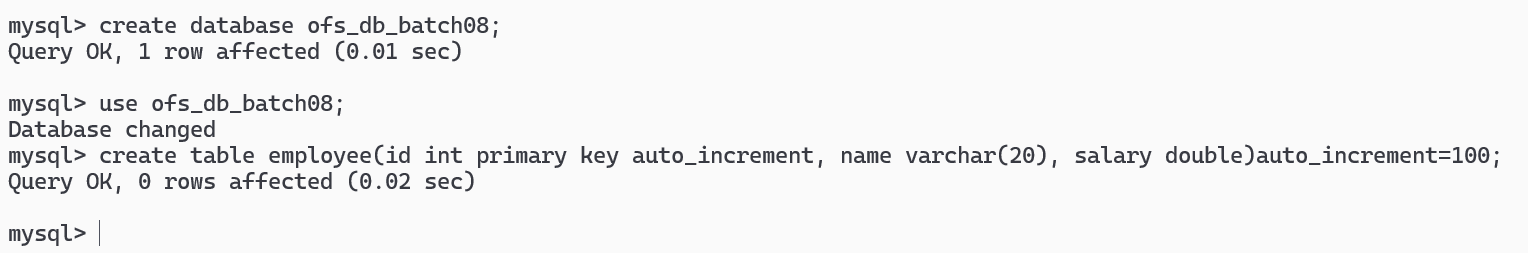
employee.js



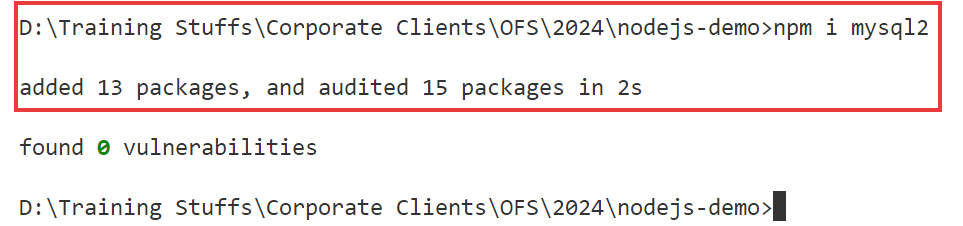
Note: The export can be written in 2 ways, the one which is written is a named export that can be imported with the same name i.e. import { Employee } from ‘employee.js’; there’s another type of export called default export, that can be imported in any name, its syntax doesn’t include { } i.e, import E from ‘employee.js’;

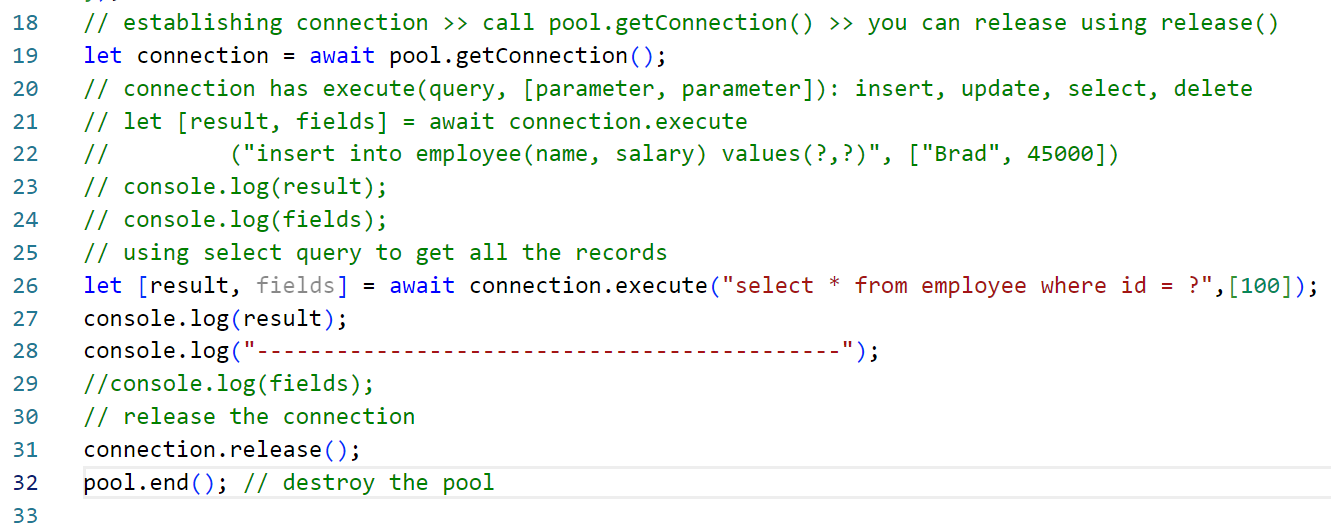
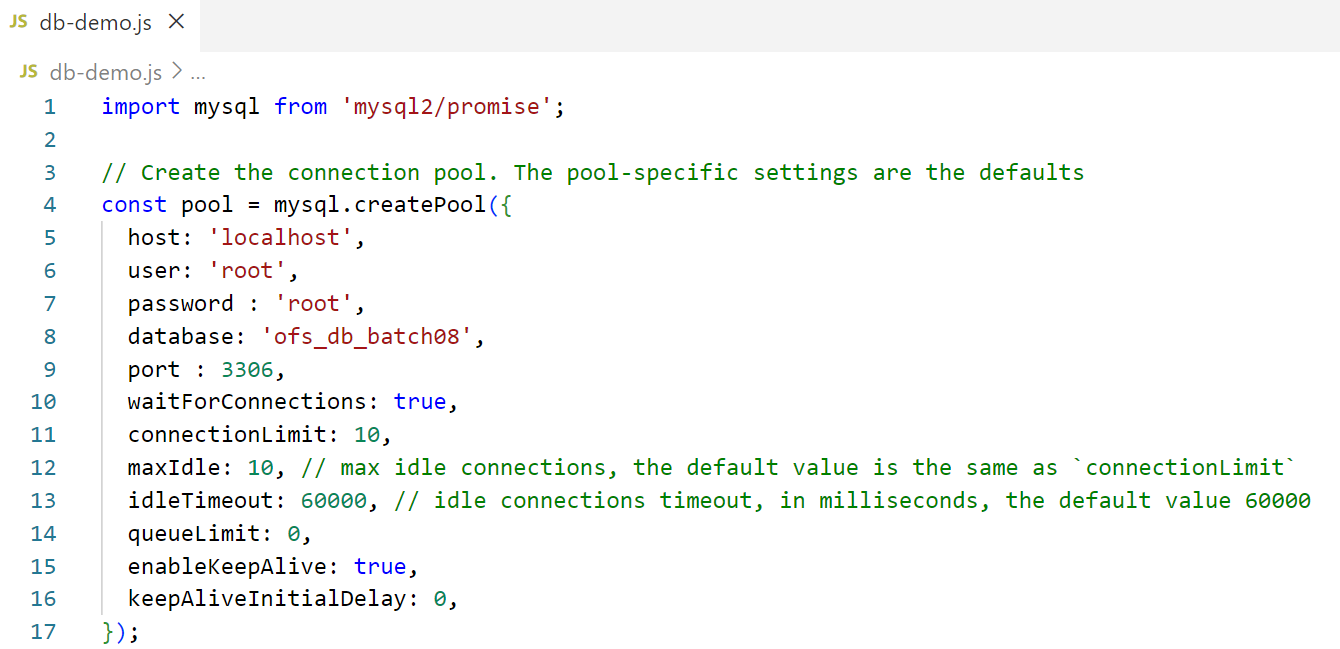


Create database table employee

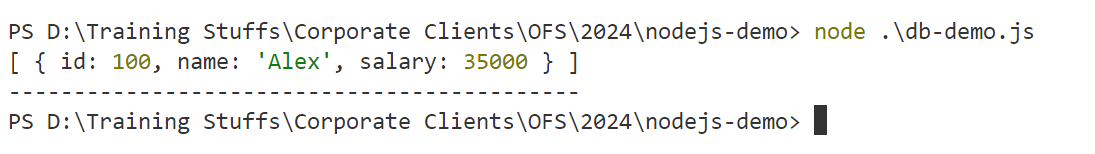


mysql2: lt is a library used to interact with mysql database





Output:



Summary:

* Require JS
* Node.js - REPL, Types of modules, fs, readline-sync, mysql2

Day 3 Agenda

* 1. Creating webservices using Node
  2. Knockout.js
  3. OJET

What are webservices

These are online services which can share the data across heterogenous applications

ex: IRCTC can share data to Phone pay or Google pay, these UPI Apps can share data to various banking services

There are 2 types of webservices

1. SOAP webservice: exchanges the data in XML format
2. RESTful webservice: exchanges the data in XML/JSON/CSV/TEXT formats

SOAP: Simple Object Access Protocol

ReST: Representational State Transfer

RESTful webservice uses HTTP protocol to share the data, it uses HTTP methods to let you create webservice, these are methods of HTTP

1. get: Read/Fetch
2. post: Storing/Creating new resource
3. put: Update the existing resource
4. delete: Deleting the resource

Different technologies use different libraries and framework to develop RESTful webservice

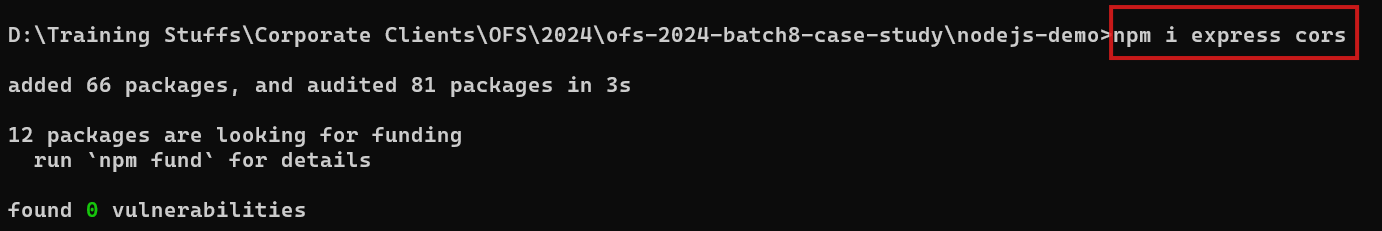
1. Java: Spring Boot / JAX-RS - Jersey
2. Node.js: Express.js
3. Python: Django & Flask
4. C#: ASP.NET

How to develop webservices using Node.js

1. Install express
2. If you are connecting to the backend install mysql2
3. If you want your webservice to be accessed by cross-origin clients, install cors library

All these libraries can be installed at once

npm install mysql2 express cors



Steps to create webservice

* Import the express & create its object
* Import the cors and add this to the express
* Create routes that can take HTTP requests
* Start the server - express itself has inbuilt server

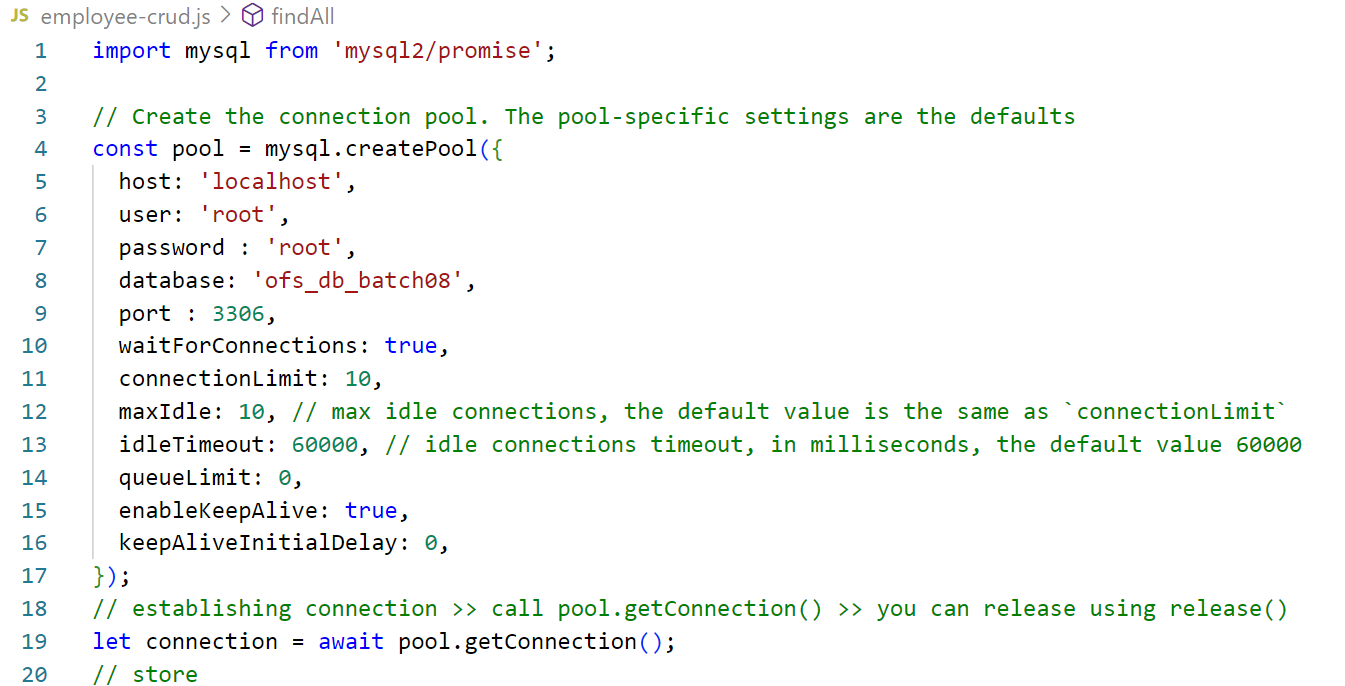
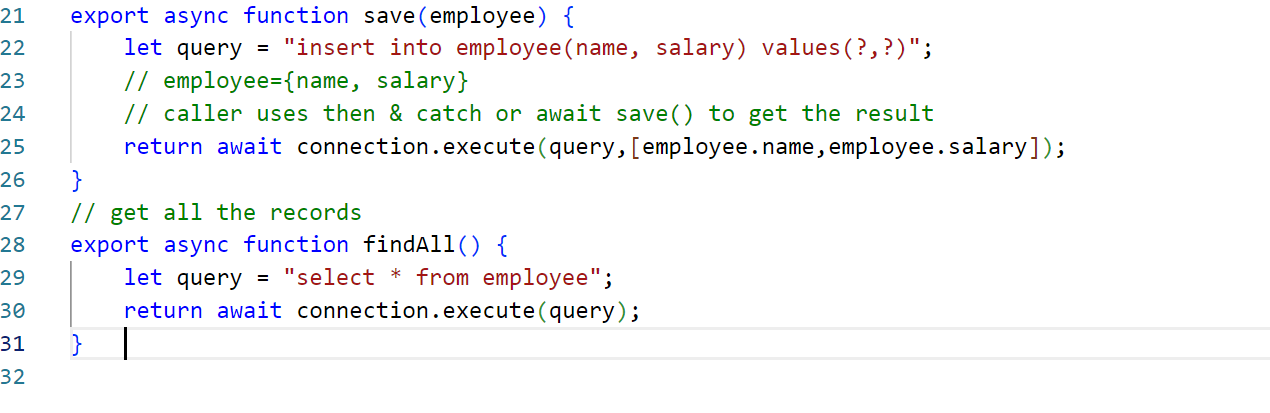
import express from ‘express’;  
import cors form ‘cors’;  
// creating express object  
let app = express();  
//adding cors to the express  
app.use( cors() ); // adds cors to the express

// creating HTTP routes  
app.get(url, (request, response) => { … handles request & generate response } );  
app.post(url, (request, response) => { .. handles request & generate response });

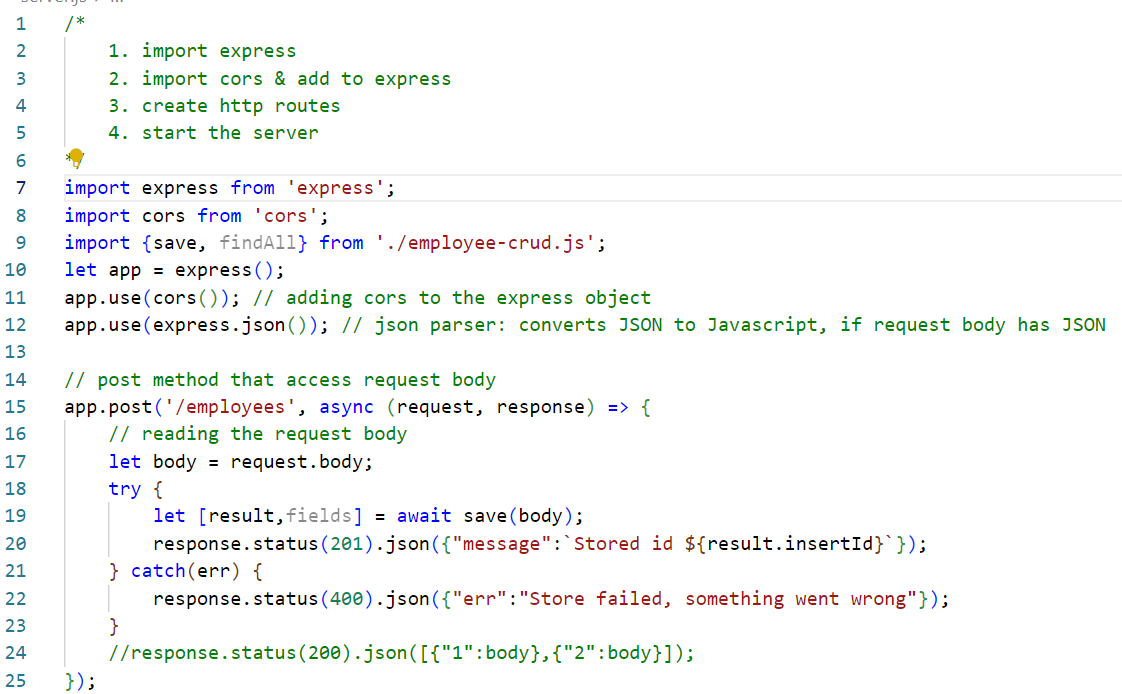
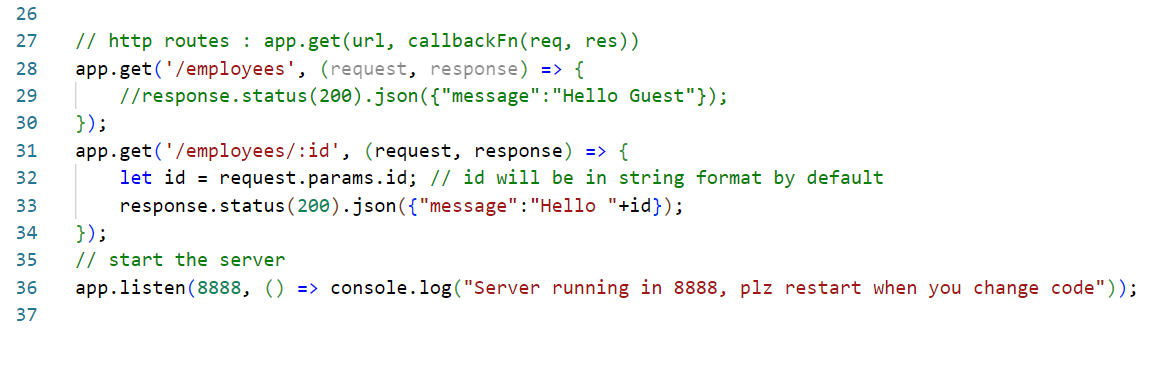
// start the server

app.listen(8888, () => { .. callbackFn that executes once server starts … })

employee-crud.js

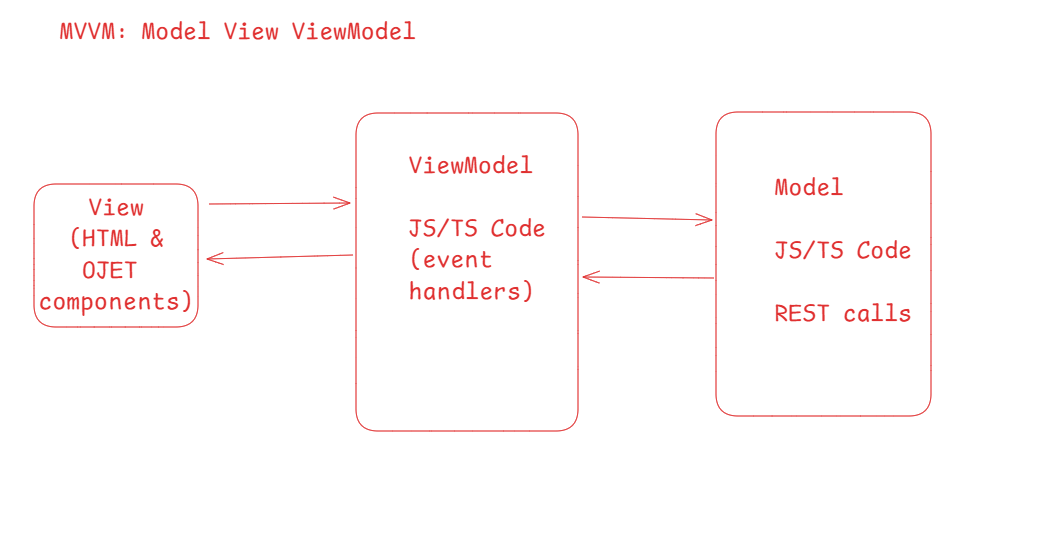
server.js

OJET: Oracle Javascript Extension Toolkit, it is built on top of Knockout.js to create client side applications like UI’s for web & mobile

OJET UI’s you cannot create by remembering the code, because there are lot of inbuilt tags & attributes, Oracle employees must use Oralce Cookbook for OJET & OJET official website to understand OJET.

OJET uses MVVM architecture, which is used even by Knockout.js



OJET project can be created using javascript/typescript, but typescript has an advantage of types & results are reliable

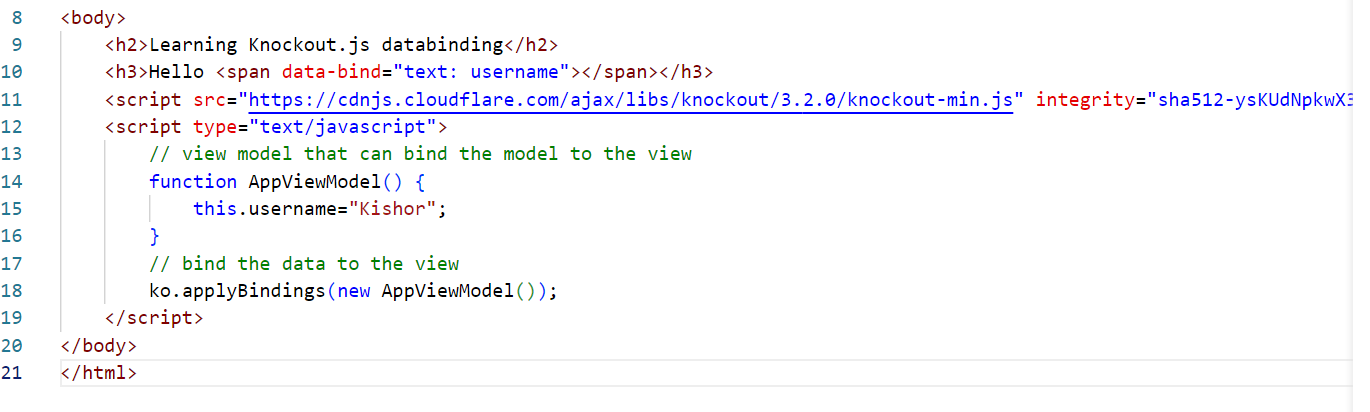
Software’s required

1. Node.js
2. NPM

Pre-requisites

* HTML
* CSS
* Javascript/Typescript
* Basics of Knockout.js

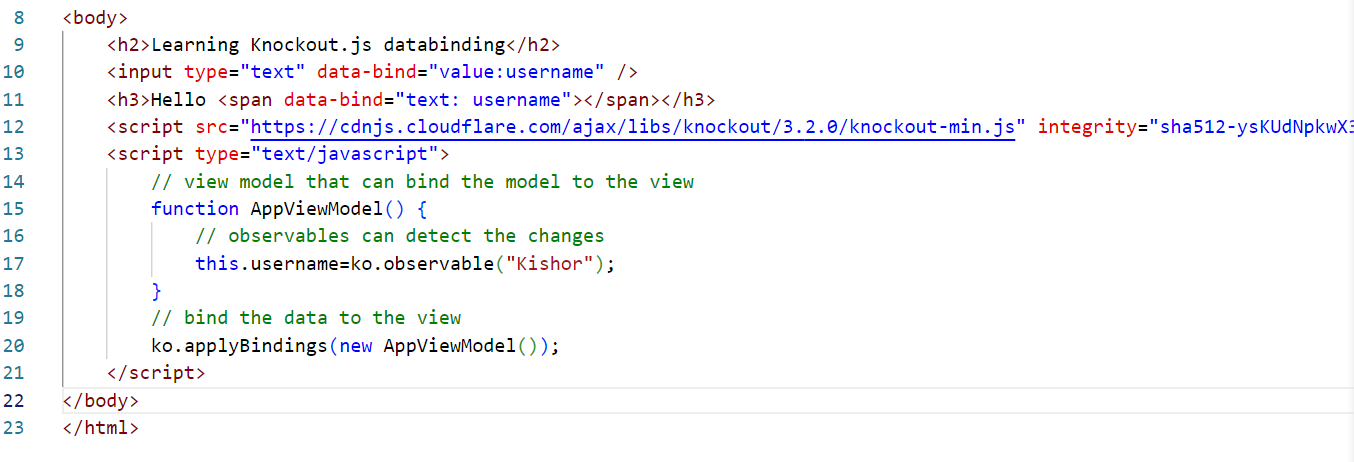
Hello world program on KO



Output:



Editing the data

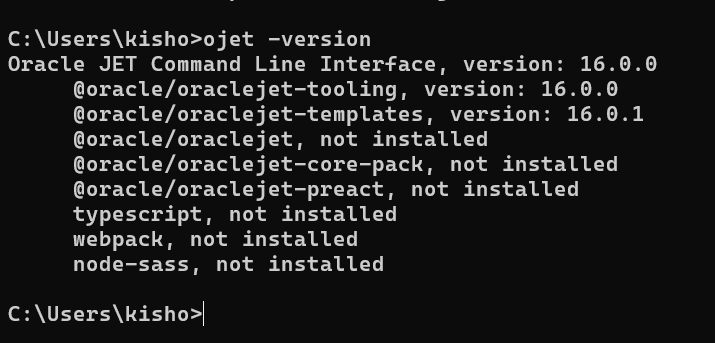
To edit the data we must use knockout js observables, simple text data variables cannot detect the changes done to the variables from the view  


OJET extensively uses knockout.js databinding

OJET gives you starter projects which will have inbuilt folder structures for views, viewModels, ojet libraries and basic navigations and etc.

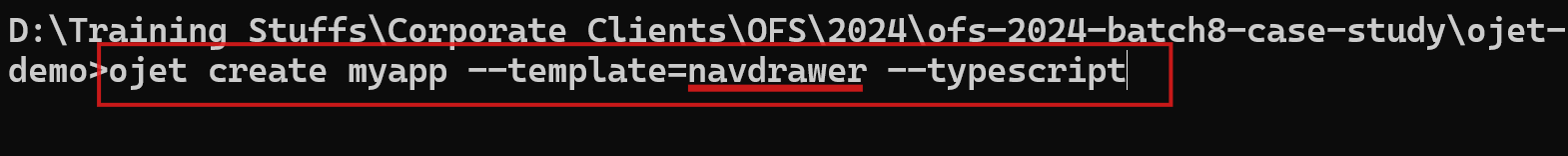
*npm install -g @oracle/ojet-cli@16*

Verifying the installation



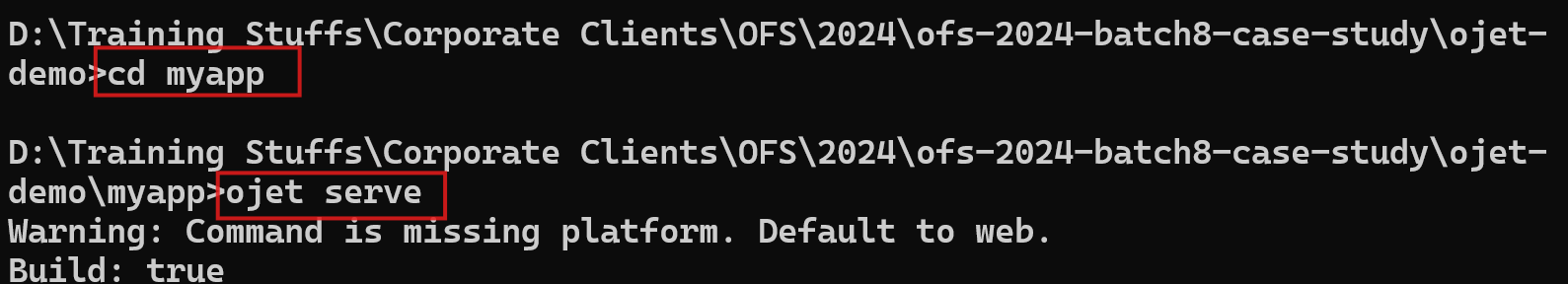
How to create ojet project

ojet create project-name --tempalte=navdrawer --typescript

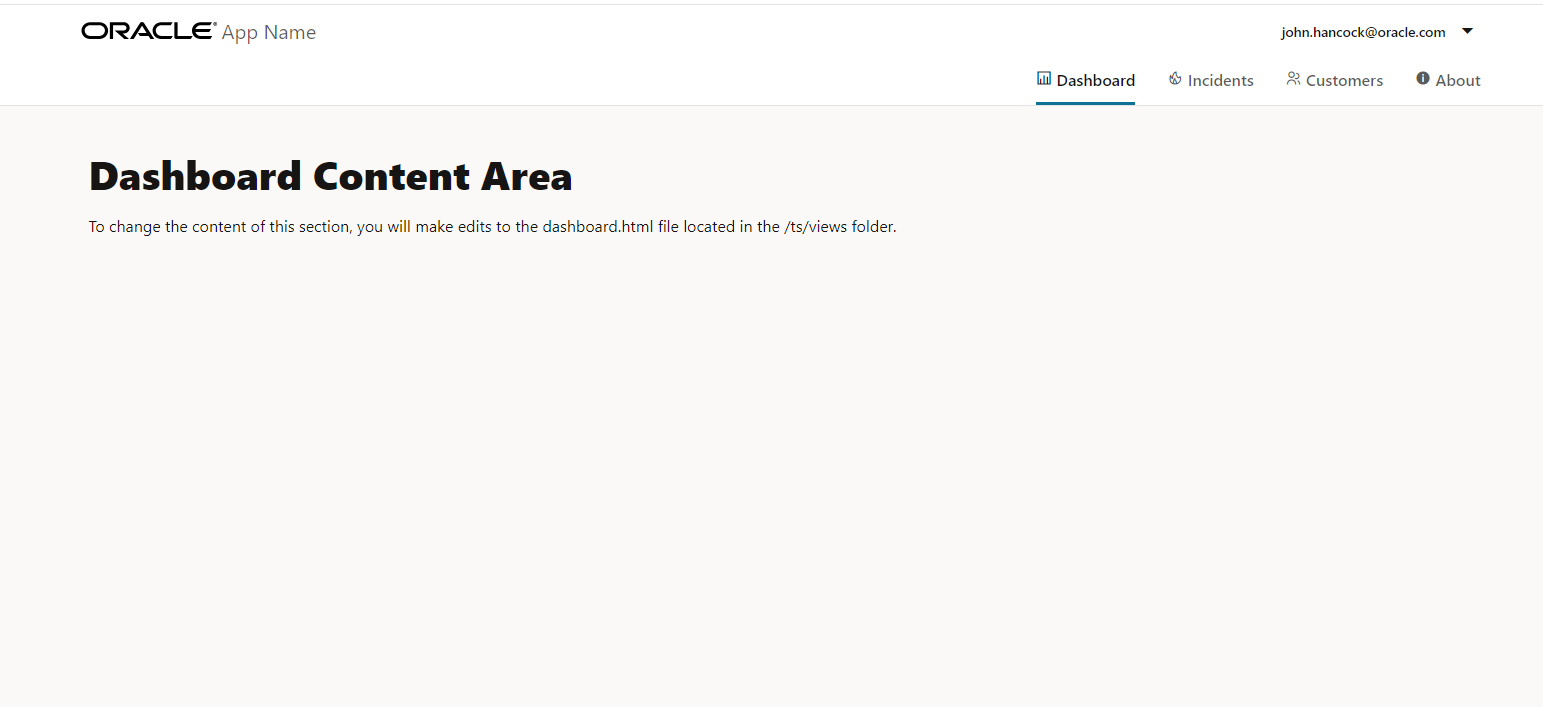


cd myapp

ojet serve: This runs the project & opens the application in the browser



Output:



OJET Cookbook

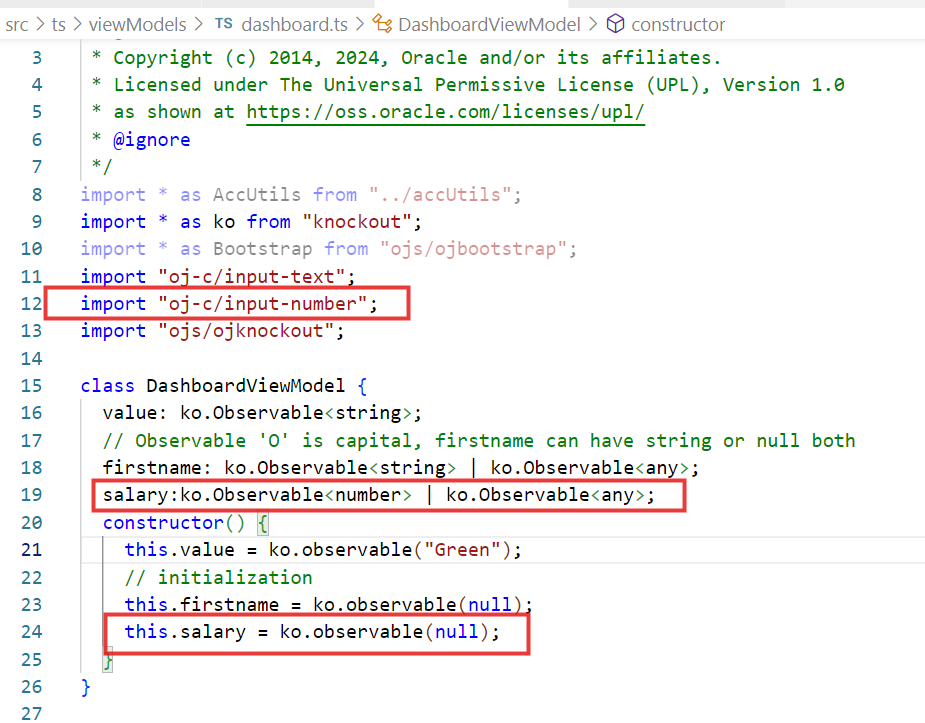
1. You will see all the ojet components with HTML(view) & corresponding JS/TS (viewModel)
2. You must look for the cookbook output and use it in your application.

[[ ]] is read-only, one way data-binding

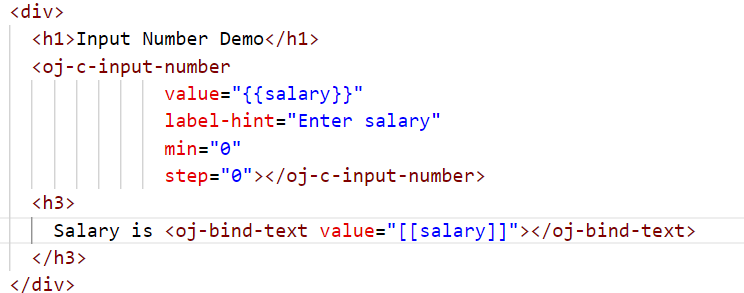
{{ }} is read & write: two way data-binding

Using Input Number

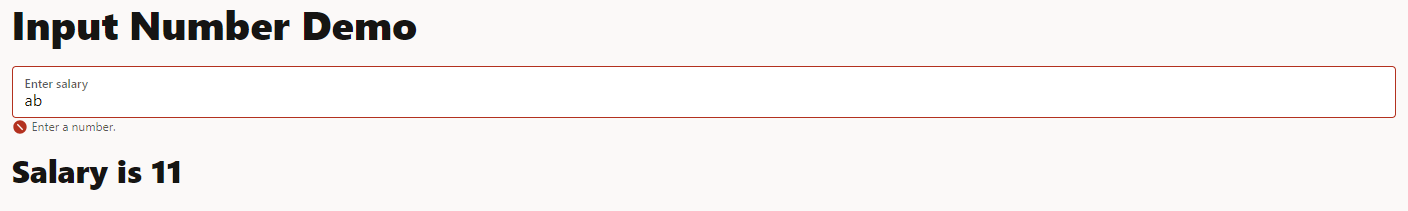
dashboard.ts



dashboard.html



Output:

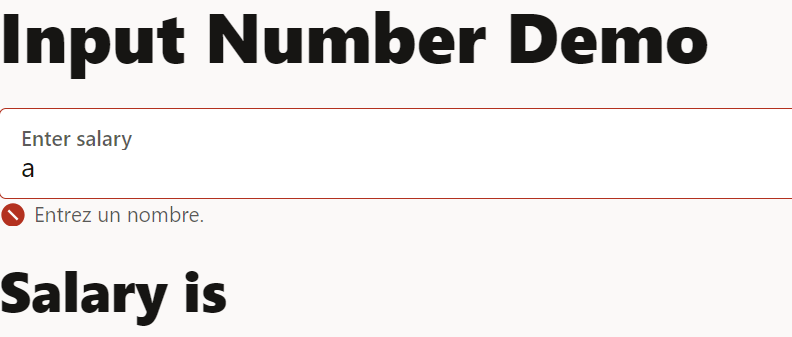


I18N(Internationalization): Ability to make software to adapt to different county languages

L10N(Localization): For every country specific locale resource will be there ex: fr-FR (French), us-en(English)



Output:



Objective:

Enhancements: You can navigate from one page to another page without reloading the entire page (Single page application).

Things to try before creating the UI for the case-study

1. Try Form components like: input text, input number, input password, input date, form layouts (activity-02)
2. Try Controls like: buttons, messages, progress (activity-03)
3. Try Collections like: table, list view, paging control (activity-04)
4. Try Framework like: RestDataProvider, (activity-04)

(or) you can create a single activity-all folder & upload html & TS files

After that try the below lessons provided by Oracle to understand how to develop OJET application from scratch to accessing backend with REST calls (activity-06)

<https://docs.oracle.com/en/middleware/developer-tools/jet/16/webapplications.html>

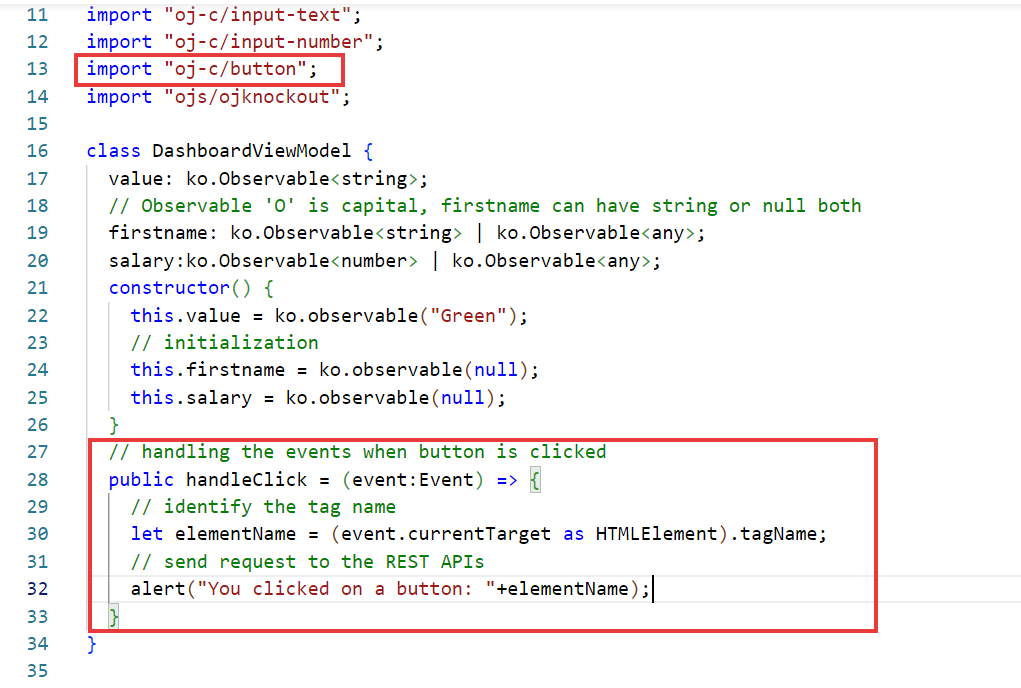
Lastly complete your case-study

OJET application must interact with Backend service (spring boot & node.js)

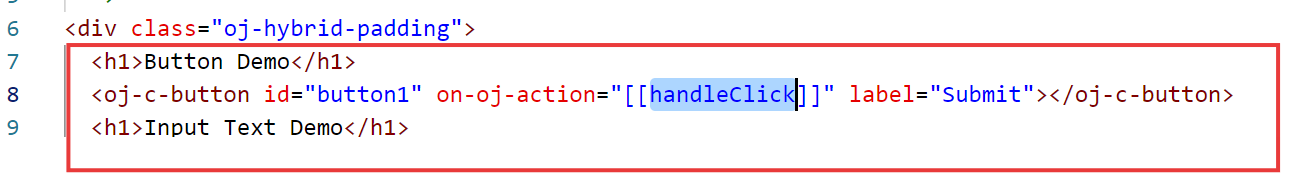
1. Minimum OJET components (dashboard) of spring boot:
   1. Registration
   2. Login
   3. Fetching any details & showing in the page
2. Minimum OJET component(customer) of node.js :
   1. Store
   2. Retrieve
   3. Find By Id

Event Handling

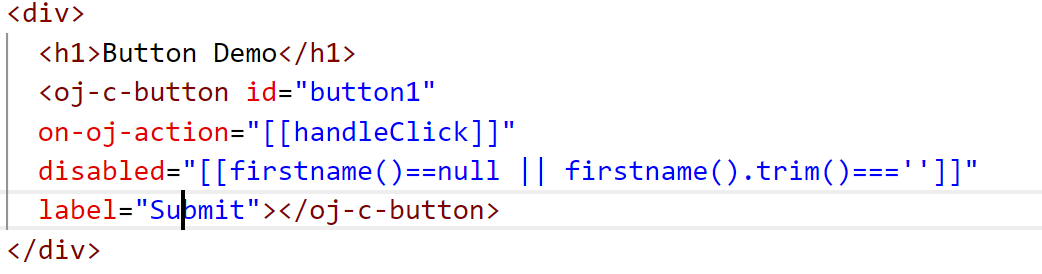
dashboard.ts



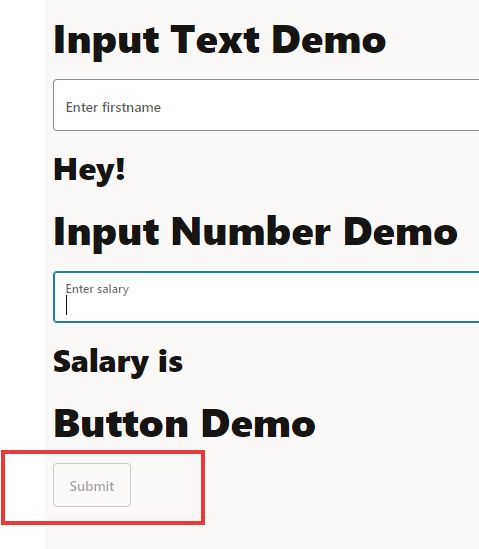
dashboard.html



How to disable the button



Output:



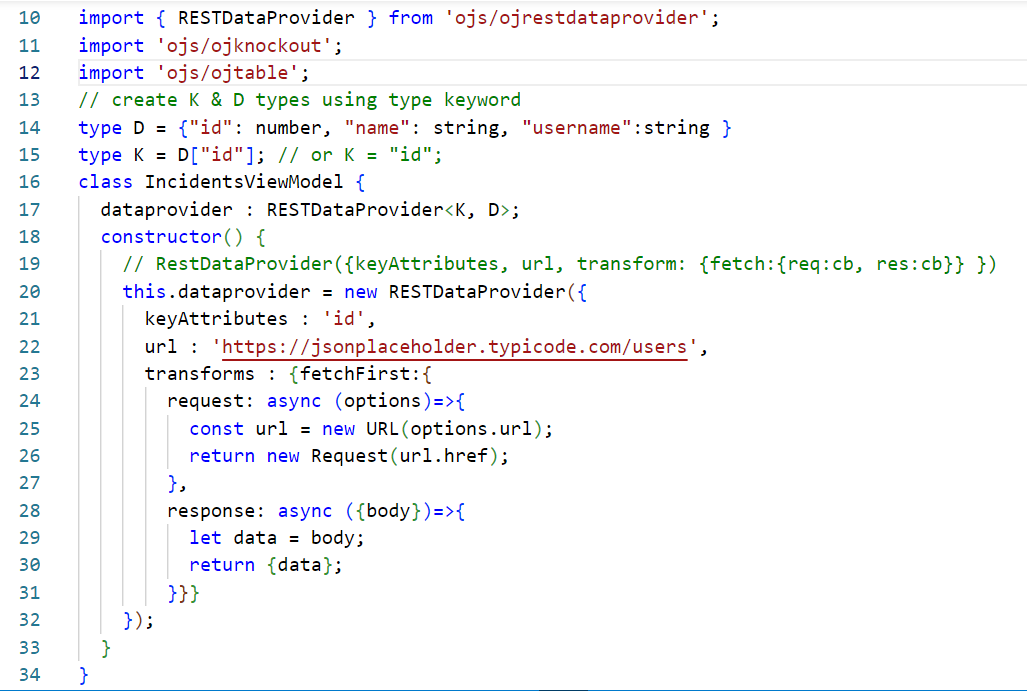
How to access fake json data from OJET

There are many ways

1. Using traditional fetch API
2. Using OJET component RestDataProvider that can help you to create dynamic content after calling the backend like tables, grids and etc

RestDataProvider: It is used to access the backend, it can directly map the data coming from the backend to the table or any other collections.

RestDatProvider( { keyAtrributes, url, transform : { fetch: {request, response } } } );



Two things you must learn

1. Making request with different HTTP methods and also if wants to update, then update the dataprovider using mutate of RestDataProvider
2. Core Router: For navigations