# Tutorial 9: NodeJS

- Javascript based
- Single threaded
- Non-blocking
- Express: web framework

# Introduction

This will extend the previous tutorial and guide you to build a complete CRUD for both backend and frontend

**Create  a package.json file**
**Create app.js file**

*Alternative: you can generate a package.json file by running a command:* ***npm init***

Open your folder using Visual Studio Code

# Start a nodejs server

Update app.js with the following content:

```
var http = require('http')


app = http.createServer(function(req, res){
   res.end('Hello world')
}).listen(3000 )
```

**You now can start a Restful web server by typing: node app.js**
**Open your browser and type: localhost:3000**

# Use express framework

Express framework makes life much easier to build a web application. Install it by running command: **npm install express**

```
const express = require('express')
```

```
const app = express()

app.get('/', function(req, res){
   res.send('Hello express')
})

app.listen(3000)
```

**You now can start a Restful web server by typing: node app.js**
**Open your browser and type: localhost:3000**

***Practice: try to define 5 endpoints for CRUDs using express framework (post, delete, put, get one, get all)***

# Defining modules

Modular design helps an app become better organized.

To define a module use:

```
exports.function_name = function(){ … }
```

Save it in a file, i.e. greeting.js

To use a module, first require it (remember the dot and slash ./):

```
const greeting = require(‘./greeting’)
```

Now use it: `greeting.function_name()`

For example: `greeting.hello(), greeting.bye()`

# Write a middleware

Middleware is a function that runs in between req-res cycles.

```
const express = require('express')
const app = express()
```

```
const logger = function(req, res, next){
   console.log(req.url)
   next()
}

const datetime = function(req, res, next){
  req.accessed = Date()
   next()
}

app.use(logger)
app.use(datetime)

app.get('/', function(req, res){
   res.send('Hello world at ' + req.accessed)
})

app.get('/about', function(req, res){
   res.send('About page at '+req.accessed)
})

app.listen(3000)
```

# Use Router # React Router DOM

This router is different from React Router.
Express Router is used to modularize an NodeJS
Instead of defining all the routes in a single app.js file, we can separate them into different files.

**bird.js**

```
var express = require('express')
var router = express.Router()

// middleware that is specific to this router
router.use(function timeLog (req, res, next) {
  console.log('Time: ', Date.now())
  next()
})
```

```
// define the home page route
router.get('/', function (req, res) {
  res.send('Birds home page')
})
// define the about route
router.get('/about', function (req, res) {
  res.send('About birds')
})

module.exports = router
```

And in the **app.js**, here is how we use it:

```
var birds = require('./birds')

// ...

app.use('/birds', birds)
```

More pls read here:
https://expressjs.com/en/guide/routing.html

**Practice: Do similar routers for students, employees etc.**

Notes: for more info about **modules.exports** and **exports.function_name**, please read here:
https://www.sitepoint.com/understanding-module-exports-exports-node-js/