**Nodejs**

**How to clone repository?**

Graphical user interface, text, application, email, website

Description automatically generated

🡪 git clone [path]

Text

Description automatically generated

**package.json**

JSON file contains meta data or information about your application. Ex: name, version, authors, address of git repository, dependencies.

🡪 before adding any node package to you app, first you need create package.json.

How to create package.json file? -🡪 npm init

After npm init, answer all questions such as package name, version, description, etc.

For your practice, there is another way allows you to answer all questions by default values using a flag --yes

🡪 npm init –yes

**How to install third party node package?**

First, we can search for any package using this link 🡪 <https://www.npmjs.com/>

Graphical user interface, text, application

Description automatically generated

by clicking on express link, we can find

Graphical user interface, text, application

Description automatically generated

As we can see all the information about express package

To install express🡪 you can find it in the right section 🡪install

npm i express

we can also see how many dependencies that express depend on.

There is an explanation for how to use express in your code

After installing Express, we can find it in dependencies in package.json

Text

Description automatically generated

Note: npm install get the latest version of packages and installed them in node\_modules folder

Graphical user interface, text

Description automatically generated

node\_ modules is too large so 🡪 to exclude it 🡪 add it to .gitignore file

**.gitigonre** file🡪 tells git which files to ignore or exclude from pushing them to GitHub

**How to use third party package in code?**

**Ex: Express**

After installing express by running 🡪 npm i express

🡪Use Express in code to build our server, and do CRUD operations (Create, Read, Update, Delete).

Import express using **require** keyword

const express = require(‘express’) 🡪 returns a function

const app = express () 🡺 create an object of **express** and store it in **app**

**app** object has a lot of useful methods

app.get () 🡪responds to http **get** request to get or read data

app.post () 🡪 responds to http **post** request to post or create new data

app.put () 🡪 responds to http **put** request to update or modify the all entire resource

Note: **app.patch()** 🡪 responds to http **patch** request to update or modify part of the data without modifying the entire data.

app.delete () 🡪 responds to http **delete** request to delete data

Ex:

**app.get (‘/’, (req, res) =>{**

**//code block**

**res.send(‘Hello world’)**

**})**

**EX:**

**app.get(‘/api/products’, (req, res)=>{**

**// getting data from db**

**}**

1st argu : / 🡺 endpoint which is root

2nd arg: 🡪 callback function which be called when http get request occures

The callback fn has 2 args

🡪 1st : **req** which is request object

🡪2nd: **res** which is response object

Go to 🡪 <https://expressjs.com/en/4x/api.html>

Graphical user interface, text, website

Description automatically generated

In the left side 🡪 request and response object, so we can see all the available properties for these objects

Graphical user interface, text, application

Description automatically generated

**How to listen on a given port?**

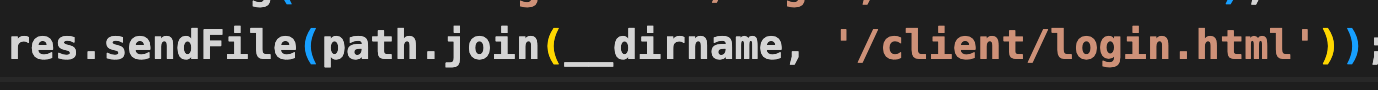
**app.listen(3000, ()=>{**

**console.log(‘listening on port 3000’)**

**})**

**How to send or transfer File?**

<https://expressjs.com/en/4x/api.html#res.sendFile>



sendFile 🡪 methods which sends or transfers the file at the given path

path.join 🡪 join the arguments together to get the path

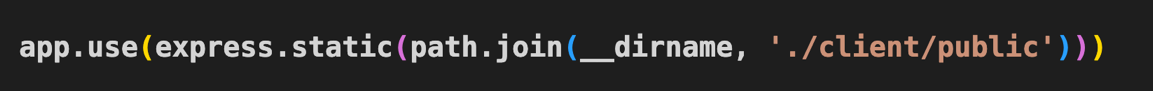
\_\_dirname 🡪 get the absolute path of the directory

**What is a middleware function?**

a function that takes the http request object, handles it, and then returns the response to the client or passes the control to another middleware function.

**How to use middleware function?**

app.use () 🡪 use method takes request handlers



**What is express.static ?**

express.static 🡪 built-in middleware function in express to serve static files or assets. Such as images, CSS, JavaScript files

<https://expressjs.com/en/starter/static-files.html>

**What is nodemon?**

nodemon is a tool that helps developers to restart the node app when file changes without running node index.js every time

to install nodemon 🡪 npm i nodemon

then we can find it in devDependencies in package.json file



**Deployment on vercel.com**

<https://vercel.com/login>

<https://shadowsmith.com/how-to-deploy-an-express-api-to-vercel>

1-create or sign up an account

2-import git repository, choose

3- configure your project

4- click deploy

5- continue to dash-board

Graphical user interface, application

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, application

Description automatically generated