

NodeJs demos - frontend - backend

- <https://www.w3schools.com/nodejs/>
- <https://nodejs.org/en/download>
- Install nodejs
- Create nodejs project
- Initialize nodejs folder
 - Frontend, Backend
 - src/index.js
 - scr/index.html
- npm init -y
- package.json
- Install, un-install javascript packages ie: uuid
- .gitignore
 - node_modules
 - .env

Node js backend /server side , webserver demo #1

1. create webserver folder
 \demo-nodejs-backend-server-1

2. create server index.js file
 src\index.js

3. server code:

```
function main()
{
  console.log("#Demo-nodejs-backend-server-1.0.0")
}
```

main()

4. run server
node src/index.js

node js frontend/client side website demo #1

1. create frontend app folder-

\demo-nodejs-frontend-webserver-1

2. create server index.js file

src\index.html

src\js

src\css

4. run server

vs.code live server

or

npm install http-server

brew install http-server

http-server -a localhost -p 8000

nodejs features:

- package.json - node js , configuration and js dependencies

create a package.json file

goto project folder before "src"

npm init

accept all defaults

or

npm init -y

node will create package.json

package.json example file

```
{
  "name": "demo-frontend-website-1",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}
```

node js backend /server side , webserver demo #1

1. create webserver folder
demo-nodejs-backend-server-1

2. create server index.js file
src\index.js

3. server code:

```
function main()
{
  console.log("#Demo-nodejs-backend-server-1.0.0")
}
```

main()

4. run server
node src/index.js

node js frontend/client side website demo #1

1. create frontend app folder
demo-nodejs-frontend-webserver-1

2. create server index.js file
src\index.html
src\js
src\css

4. run server
vs.code live server
or
npm install http-server
brew install http-server
http-server -a localhost -p 8000

nodejs features:
- package.json - node js , configuration and js dependencies

create a package.json file

goto project folder before "src"

npm init

accept all defaults

or

npm init -y

node will create package.json

package.json example file

```
{
  "name": "demo-frontend-website-1",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}
```

node js frontend/client side website demo #2

- add a nodejs javascript library
- example: uuid
 - uuid - generate unique id

use npm to install uuid javascript library

- npm install uuid

1. create frontend app folder
 \demo-nodejs-frontend-webserver-2

2. create server index.js file
 src\index.html
 src\js
 src\css

4. run server
 vs.code live server
 or
 npm install http-server
 brew install http-server
 http-server -a localhost -p 8000

nodejs features:
 - package.json - node js , configuration and js dependencies

create a package.json file

goto project folder before "src"

npm init

accept all defaults

or

npm init -y

node will create package.json

package.json example file

```
{
  "name": "demo-frontend-website-1",
```



```

"version": "1.0.0",
"main": "index.js",
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"author": "",
"license": "ISC",
"description": ""
}

```

Add uuid javascript library

- to generate unique ids
- from a folder before "src" folder
- same folder as package.json
- enter the following at the command line
- for Windows:
 - goto windows command line instead of powershell
- npm install uuid
- for macos
 - sudo npm install uuid

npm creates a "node_modules" folder with uuid library

- package.json also contains npm javascript library dependencies

```

{
  "name": "demo-frontend-website-2",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "uuid": "^11.0.3"
  }
}

```


installing nodejs npm javascript packages

- ie: npm install uuid

un-installing nodejs npm javascript packages

- ie: npm uninstall uuid

re-installing all package in package.json

- ie: npm install

introduction to .gitignore file

- .gitignore
- exclude files and/folders that you do not want to upload to github repo

example: .gitignore

node_modules

.env

node js backend /server side , webserver demo #1

1. create webserver folder
demo-nodejs-backend-server-1
2. create server index.js file
src\index.js

3. server code:

```
function main()
{
  console.log("#Demo-nodejs-backend-server-1.0.0")
}
```

main()

4. run server
node src/index.js

node js frontend/client side website demo #1

1. create frontend app folder
 \demo-nodejs-frontend-webserver-1

2. create server index.js file
 src\index.html
 src\js
 src\css

4. run server
 vs.code live server
 or
 npm install http-server
 brew install http-server
 http-server -a localhost -p 8000

nodejs features:
 - package.json - node js , configuration and js dependencies

create a package.json file

goto project folder before "src"

npm init

accept all defaults

or

npm init -y

node will create package.json

package.json example file

```
{
  "name": "demo-frontend-website-1",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
```

```
"description": ""
}
```

node js frontend/client side website demo #2

- add a nodejs javascript library
- example: uuid
 - uuid - generate unique id

use npm to install uuid javascript library

- npm install uuid

1. create frontend app folder
 \demo-nodejs-frontend-webserver-2

2. create server index.js file
 src\index.html
 src\js
 src\css

4. run server
 vs.code live server
 or
 npm install http-server
 brew install http-server
 http-server -a localhost -p 8000

nodejs features:
 - package.json - node js , configuration and js dependencies

create a package.json file

goto project folder before "src"

npm init

accept all defaults

or

npm init -y

node will create package.json

package.json example file

```
{
  "name": "demo-frontend-website-1",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Add uuid javascript library

- to generate unique ids
- from a folder before "src" folder
- same folder as package.json
- enter the following at the command line
- for Windows:
 - goto windows command line instead of powershell
- npm install uuid
- for macos
- sudo npm install uuid

npm creates a "node_modules" folder with uuid library

- package.json also contains npm javascript library dependencies

```
{
  "name": "demo-frontend-website-2",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "uuid": "^11.0.3"
  }
}
```

```
}  
}
```

installing nodejs npm javascript packages

- ie: npm install uuid

un-installing nodejs npm javascript packages

- ie: npm uninstall uuid

re-installing all package in package.json

- ie: npm install

introduction to .gitignore file

- .gitignore
- exclude files and/folders that you do not want to upload to github repo

example: .gitignore

Node_modules

.env