

node.js

Posted March 1st, 2024

node.js is a server that runs javascript on the server

1. bun is a replacement for node.js

<https://bun.sh>

2. generally nodejs is installed via yum, dnf, or whatever package manager you have

3. node.js is javascript engine that runs javascript on the server side

4. To setup a project

npm init

5. install express server

npm install express --save

see <https://expressjs.com>

6. Here is "Hello World!" using an express server

See <https://expressjs.com>

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

```
const app = express()
```

```
const port = 3000
```

```
##
```

```
## App get sets a route for /
```

```
app.get('/', (req, res) => {
```

```
  res.send('Hello World!')
```

```
})
```

```
app.listen(port, () => {
```

```
  console.log(`Example app listening on port ${port}`)
```

```
})
```

7. BUT - really nobody uses the simple version...

we use express-generator

install express-generator

>npm install -g express-generator

run install

>express

This generates a structure in the current directory including:

routes

views

bin

public

node_modules

a) public is for static html files

b) routes defines how to route urls

c) bin

bin/www is what starts the server

8. nodemon - sets up node so that it will reread files

upon change.

> npm install --save-dev nodemon

9. Fetching data from the database.

a) create an object that fetches data (file con.js)

```
var mysql = require('mysql2');
```

```
var con = mysql.createConnection({
```

```
  host:'localhost',
```

```
  user: 'xxxxxuser',
```

```
  password:'yyyypass',
```

```
  database:'foodb'
```

```
});
```

```
con.connect(function(err) {
```

```
  if (err) throw err;
```

```
});
```

```
async function sqlquery(sql)
{
const results = await con.promise().query(sql);
// returns a list of [data,meta]
return results[0];
}

module.exports = {
sqlquery
}
```

```
b) from routes/index.js
var con = require("../services/con");
router.get('/fubar', async function(req,res,next) {
let sql = 'show tables';
res.json(await con.sqlquery(sql));
});
```

c) await can only be used from an async function
the function being called needs to be async
and we do an await on a promise from con.query...
the results returned are a list of elements, data, and meta data
This appears complex.
basically the mysql query is run in the background unless
we force nodejs to wait for the results