

The following exercises are related to the use of Node/Express.

1. Create a git repository for your answers to this problem sheet. Push the repository to GitHub. Make a commit and push it to GitHub after each exercise.
2. What is Express? Again add you answer to your README file. <https://expressjs.com>
3. Write an server application that listens on localhost:3000 and returns “Welcome to Data Representation & Querying”. Follow the following tutorial: <https://expressjs.com/en/starter/hello-world.html>
4. Add another route point using the Express Framework that will take an argument form the url path and return hello “argument” for the url <http://localhost:3000/hello> eg <http://localhost:3000/hello/Ted> returns hello Ted
5. Add a route point `/api/books` that will return the following json data:

```
{
  "books": [
    {
      "title": "Learn Git in a Month of Lunches",
      "isbn": "1617292419",
      "pageCount": 0,
      "thumbnailUrl":
        "https://s3.amazonaws.com/AKIAJC5RLADLUMVRPFDQ.book-thumb-images/umali.jpg",
      "status": "MEAP",
      "authors": ["Rick Umali"],
      "categories": []
    },
    {
      "title": "MongoDB in Action, Second Edition",
      "isbn": "1617291609",
      "pageCount": 0,
      "thumbnailUrl":
        "https://s3.amazonaws.com/AKIAJC5RLADLUMVRPFDQ.book-thumb-images/banker2.jpg",
      "status": "MEAP",
      "authors": [
        "Kyle Banker",
        "Peter Bakkum",
        "Tim Hawkins",
        "Shaun Verch",
        "Douglas Garrett"
      ],
      "categories": []
    }
  ],
}
```

```
{
  "title": "Getting MEAN with Mongo, Express, Angular, and Node",
  "isbn": "1617292036",
  "pageCount": 0,
  "thumbnailUrl":
    "https://s3.amazonaws.com/AKIAJC5RLADLUMVRPFDQ.book-thumb-images/sholmes.jpg",
  "status": "MEAP",
  "authors": ["Simon Holmes"],
  "categories": []
}
]
```

6. Add another route `/test` that returns a html page when Queried. Your file should called `index.html`. e.g `app.get`
7. Add a form to `index.html` with a two text boxes with `firstname` and `lastname` with the method `GET`. Set the form action to `"/name"` Add a submit button. In `server.js`, change the `"/name"` route to accept the name variable as a URL encoded `GET` variable. Run the server and test.
8. Add another form to `index.html` with a two text boxes with `firstname` and `lastname` with the method `POST`. Set the form action to `"/name"` Add a submit button. In `main.js`, change the `"/name"` route to accept the name variable as a body encoded `POST` variable. Run the server and test. There will be two methods in `node.js` one for `GET` and one for `POST`.