

TIC3001 Task 2 - Web Application

- Name: Ke Yule
- Student Number: A0211495H E0493826
- Github: <https://github.com/keyule/3001-Task2>

View the markdown version for better formatting at:

<https://github.com/keyule/3001-Task2/blob/main/Report/report.md>

Task 2.1 - Simple Backend

Running

1. `git clone https://github.com/keyule/3001-Task2.git`
2. `docker-compose up --build -d`
3. API sits on port 3000
4. Web server sits on port 8080
5. Tests run on port 3001

Set up

API: Port 3000

End Point	Type	Description	Parameters
/todos	GET	Returns all todos	-
/todos	POST	Creates a new todo	{"title": "test"}
/todos/:id	PUT	Marks as completed	{"completed": true}
/todos/:id	DELETE	Deletes a todo	-

Demo

Create a new todo

```
POST http://localhost:3000/todos
Content-Type: application/json
```

```
{
  "title": "test2"
}
```

```
HTTP/1.1 201 Created
X-Powered-By: Express
Access-Control-Allow-Origin: *
```

```
Content-Type: application/json; charset=utf-8
Content-Length: 42
ETag: W/"2a-hgqRctPA1nM3eRiwZa5wDSpLOb8"
Date: Tue, 18 Apr 2023 12:58:31 GMT
Connection: close
```

```
{
  "id": 1,
  "title": "test2",
  "completed": false
}
```

Get all todos

```
GET http://localhost:3000/todos
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 44
ETag: W/"2c-LrHCQ7xiXo80kBCPLp6A2+Gi/64"
Date: Tue, 18 Apr 2023 12:59:34 GMT
Connection: close
```

```
[
  {
    "id": 1,
    "title": "test2",
    "completed": false
  }
]
```

Mark todo as done

```
PUT http://localhost:3000/todos/1
Content-Type: application/json
```

```
{
  "completed": true
}
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
```

```
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 25
ETag: W/"19-6EweMRu5PbT4kJc8ETQm2u2NerE"
Date: Tue, 18 Apr 2023 12:59:57 GMT
Connection: close

{
  "id": 1,
  "completed": true
}
```

Delete todo

```
DELETE http://localhost:3000/todos/1
```

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 8
ETag: W/"8-h5EdGu1QmHe40kjsU292jNzSLfE"
Date: Tue, 18 Apr 2023 13:00:33 GMT
Connection: close

{
  "id": 1
}
```

Error Handling

Post - without title

```
POST http://localhost:3000/todos
Content-Type: application/json

{

}
```

```
HTTP/1.1 400 Bad Request
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 29
```

```
ETag: W/"1d-531IJ951G13GPLg/Tko6BPJr+/c"
Date: Tue, 18 Apr 2023 13:03:49 GMT
Connection: close

{
  "error": "Title is required"
}
```

Put - without completed

```
PUT http://localhost:3000/todos/1
Content-Type: application/json

{

}
```

```
HTTP/1.1 400 Bad Request
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 39
ETag: W/"27-LdA69xHke9IR1+chUsILYaj1Zf0"
Date: Tue, 18 Apr 2023 13:06:27 GMT
Connection: close

{
  "error": "Completed field is required"
}
```

Put - on todo that doesnt exist

```
PUT http://localhost:3000/todos/99999
Content-Type: application/json

{
  "completed": true
}
```

```
HTTP/1.1 404 Not Found
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
```

```
Content-Length: 45
ETag: W/"2d-ToTFM2da1kZ0wEjJokORMjU5CU0"
Date: Tue, 18 Apr 2023 13:07:29 GMT
Connection: close

{
  "error": "Todo item with ID 99999 not found"
}
```

Delete - on todo that doesnt exist

```
DELETE http://localhost:3000/todos/9999
```

```
HTTP/1.1 404 Not Found
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 44
ETag: W/"2c-TDLPryOpkhPgL+ZEKhHzQPi6Mb0"
Date: Tue, 18 Apr 2023 13:08:17 GMT
Connection: close

{
  "error": "Todo item with ID 9999 not found"
}
```

Task 2.1 - Testing

2.2 Testing with Mocha/Chai

- `docker-compose exec app npm run test`
- test file located at: <https://github.com/keyule/3001-Task2/blob/main/api/test/server.test.js>

```

PS C:\Users\Yule Ke\Desktop\Task2> docker-compose exec app npm run test

> api@1.0.0 test
> PORT=3001 mocha test/*.js --exit

Listening on port 3001...

TODO API
GET /todos
  ✓ It should GET all the todos (38ms)
  ✓ It should NOT GET all the todos
POST /todos
  ✓ It should POST a new todo
  ✓ It should NOT POST a new todo without the title property
PUT /todos/:id
  ✓ It should PUT an existing todo
  ✓ It should NOT PUT a todo without the completed property
DELETE /todos/:id
  ✓ It should DELETE an existing todo
  ✓ It should NOT DELETE a todo that is not in the database

8 passing (119ms)

```

2.3 Testing with travis

.travis.yml

- file located at: <https://github.com/keyule/3001-Task2/blob/main/.travis.yml>

```

language: node_js
node_js:
  - "14"
services:
  - docker
before_install:
  - docker-compose up --build -d
  - docker-compose ps
script:
  - docker-compose exec app npm run test
after_script:
  - docker-compose down

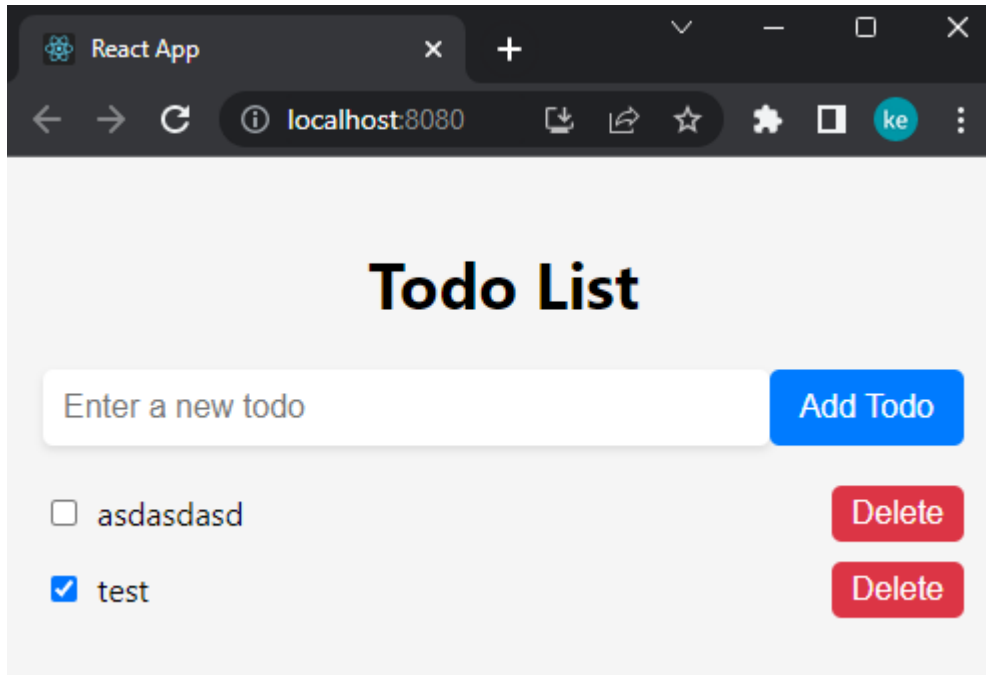
```

```

209
210 $ docker-compose up --build -d
211
212 $ docker-compose ps
213
214 $ docker-compose exec app npm run test
215
216 > api@1.0.0 test
217 > PORT=3001 mocha test/*.js --exit
218
219 Listening on port 3001...
220
221
222 TODO API
223 GET /todos
224   ✓ It should GET all the todos (60ms)
225   ✓ It should NOT GET all the todos
226 POST /todos
227   ✓ It should POST a new todo (46ms)
228   ✓ It should NOT POST a new todo without the title property
229 PUT /todos/:id
230   ✓ It should PUT an existing todo
231   ✓ It should NOT PUT a todo without the completed property
232 DELETE /todos/:id
233   ✓ It should DELETE an existing todo
234   ✓ It should NOT DELETE a todo that is not in the database
235
236 8 passing (170ms)
237
238 The command "docker-compose exec app npm run test" exited with 0.
239 store build cache
240
241 $ docker-compose down
242
243 Done. Your build exited with 0.

```

Task 2.4 - Frontend



Appendix

docker-compose.yml

```
version: "3"
services:
  db:
    image: postgres
    environment:
      POSTGRES_PASSWORD: password123
      POSTGRES_USER: user123
      POSTGRES_DB: db123
    volumes:
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql

  app:
    build: ./api
    ports:
      - 3000:3000

  web:
    build: ./web
    ports:
      - "8080:8080"
    restart: always
```