

```
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
todo list RESTful API server started on: 3000
[nodemon] restarting due to changes
```

```
1  {
2    "name": "todolistapi",
3    "version": "1.0.0",
4    "main": "server.js",
5    "scripts": {
6      "test": "echo \\\"Error: no test specified\\\" && exit 1",
7      "start": "nodemon server.js"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC",
12   "description": "",
13   "dependencies": {
14     "express": "^4.21.1",
15     "mongoose": "^6.2.4"
16   }
17 }
```

```
1  const express = require('express'),
2    app = express(),
3    port = process.env.PORT || 3000,
4    mongoose = require('mongoose'),
5    Task = require('./api/models/todoListModel'), // created model loading here
6    bodyParser = require('body-parser');
7
8  // mongoose instance connection url connection
9  mongoose.Promise = global.Promise;
10 mongoose.connect('mongodb+srv://tintdgc5220477:ductin123@fgmweb2.jpmpq.mongodb.net/?retryWrites=true&w=majority&appName=FGMWeb2');
11
12 app.use(bodyParser.urlencoded({ extended: true }));
13 app.use(bodyParser.json());
14
15 const routes = require('./api/routes/todoListRoutes'); // importing route
16 routes(app); // register the route
17
18 app.listen(port);
19
20 console.log('todo list RESTful API server started on: ' + port);
21
```

```

1  const mongoose = require('mongoose');
2  const Schema = mongoose.Schema;
3  ⚠
4  const TaskSchema = new Schema({
5    name: {
6      type: String,
7      required: 'Kindly enter the name of the task'
8    },
9    Created_date: {
10     type: Date,
11     default: Date.now
12   },
13   status: {
14     type: [{
15       type: String,
16       enum: ['pending', 'ongoing', 'completed']
17     }],
18     default: ['pending']
19   }
20 });
21
22 module.exports = mongoose.model('Tasks', TaskSchema);
23

```

```

1  const mongoose = require('mongoose'),
2    Task = mongoose.model('Tasks');
3
4  exports.list_all_tasks = function(req, res) {
5    Task.find({}, function(err, task) {
6      if (err)
7        res.send(err);
8      res.json(task);
9    });
10 };
11
12 exports.create_a_task = function(req, res) {
13   const new_task = new Task(req.body);
14   new_task.save(function(err, task) {}
15     if (err)
16       res.send(err);
17     res.json(task);
18   });
19 };
20
21 exports.read_a_task = function(req, res) {
22   Task.findById(req.params.taskId, function(err, task) {
23     if (err)
24       res.send(err);
25     res.json(task);
26   });
27 };
28
29 exports.update_a_task = function(req, res) {
30   Task.findOneAndUpdate({_id: req.params.taskId}, req.body, {new: true}, function(err, task) {
31     if (err)
32       res.send(err);
33     res.json(task);
34   });
35 };
36 exports.delete_a_task = function(req, res) {
37   Task.remove({
38     _id: req.params.taskId
39   }, function(err, task) {
40     if (err)
41       res.send(err);
42     res.json({ message: 'Task successfully deleted' });
43   });
44 };

```

POST http://localhost:3000/tasks Send

Params Authorization Headers (4) Body Options Code

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION	
<input checked="" type="checkbox"/>	name	Do soyak is a good player	Description	
	Key	Value	Description	

Body Headers JSON Status: 200 OK Duration: 93 ms

```

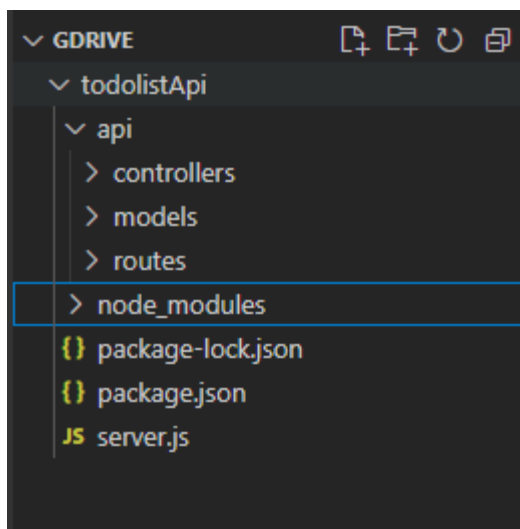
1  [{"name": "Do soyak is a good player", "status": ["pending"], "_id": "672381ac3d380c6d8d4613af", "Created_date": "2024-10-31T04:03:56.283Z", "__v": 0}]

```

```

1  const mongoose = require('mongoose');
2  const Schema = mongoose.Schema;
3  ⚡
4  const TaskSchema = new Schema({
5    name: {
6      type: String,
7      required: 'Kindly enter the name of the task'
8    },
9    Created_date: {
10     type: Date,
11     default: Date.now
12   },
13   status: {
14     type: [{
15       type: String,
16       enum: ['pending', 'ongoing', 'completed']
17     }],
18     default: ['pending']
19   }
20 });
21
22 module.exports = mongoose.model('Tasks', TaskSchema);
23

```



```
1  const express = require('express');
2  const port = process.env.PORT || 3000;
3  const app = express();
4
5  app.listen(port);
6
7  app.get('/', (req,res) =>{
8    res.send('Hello World');
9  });
10
11 console.log('Server started on port ' + port);|
```

```
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
Server started on port 3000
```