

How to make a testable node app to be tested using github actions workflow

✓ 1. Project Structure

```
testable-node-app/  
├── .github/  
│   └── workflows/  
│       └── nodejs.yml    ← GitHub Actions workflow file  
├── routes/  
│   └── index.js          ← Express route  
├── tests/  
│   └── app.test.js       ← Automated test (Jest)  
├── app.js                ← Express app  
├── package.json  
└── README.md
```

✓ 2. app.js (Express App)

```
const express = require('express');  
const app = express();  
  
const indexRouter = require('./routes/index');  
  
app.use(express.json());  
app.use('/', indexRouter);  
  
const PORT = process.env.PORT || 3000;  
if (process.env.NODE_ENV !== 'test') {  
  app.listen(PORT, () => {
```

```
    console.log(`Server is running on port ${PORT}`);
  });
}

module.exports = app; // for testing
```

✓ 3. routes/index.js

```
const express = require('express');
const router = express.Router();

router.get('/', (req, res) => {
  res.status(200).json({ message: 'Hello from Express!' });
});

module.exports = router;
```

✓ 4. tests/app.test.js (using Jest + Supertest)

```
const request = require("supertest");
const app = require("../app");

describe("GET /", () => {
  it("should return a hello message", async () => {
    const res = await request(app).get("/");
    expect(res.statusCode).toBe(200);
    expect(res.body).toEqual({ message: "Hello from Express!" });
  });
});
```

✓ 5. package.json

```
{
  "name": "testable-node-app",
  "version": "1.0.0",
  "main": "app.js",
  "scripts": {
    "start": "node app.js",
    "test": "jest"
  },
  "dependencies": {
    "express": "^4.18.2"
  },
  "devDependencies": {
    "jest": "^29.7.0",
    "supertest": "^6.3.3"
  },
  "jest": {
    "testEnvironment": "node"
  }
}
```

✅ 6. .github/workflows/nodejs.yml (GitHub Actions Workflow)

```
name: Node.js CI

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  build-and-test:
    runs-on: ubuntu-latest
```

```
strategy:
  matrix:
    node-version: [18.x]

steps:
  - name: Checkout code
    uses: actions/checkout@v3

  - name: Setup Node.js
    uses: actions/setup-node@v3
    with:
      node-version: ${ matrix.node-version }

  - name: Install dependencies
    run: npm install

  - name: Run tests
    run: npm test
```

✓ 7. To Run Locally

```
npm install
npm start # run the app at http://localhost:3000
npm test  # run tests
```

Alright, let's walk through how to write a **GitHub Actions workflow** file from scratch — step by step — to **automate testing** your Express app with **Jest**.

1. Where does it go?

Create this folder and file in your project:

```
.github/workflows/nodejs.yml
```

In VS Code:

```
mkdir -p .github/workflows
touch .github/workflows/nodejs.yml
```

2. What does it do?

It tells GitHub:

"Every time someone pushes or makes a pull request to main, spin up a temporary Ubuntu server, install Node.js, install my dependencies, and run the tests."

3. Complete Working Workflow

```
name: Node.js CI

on:
  push:
    branches: [main]
  pull_request:
    branches: [main]

jobs:
  build-and-test:
    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [18.x] # You can test multiple versions like [16.x, 18.x]
```

steps:

- name: Checkout code
uses: actions/checkout@v3
- name: Setup Node.js
uses: actions/setup-node@v3
with:
node-version: \${{ matrix.node-version }}
- name: Install dependencies
run: npm install
- name: Run tests
run: npm test

Explanation of Each Step

Step	What it does
<code>on:</code>	Tells GitHub when to run this workflow (push or PR to <code>main</code>).
<code>jobs:</code>	Defines the list of things to do. You can have multiple jobs like test, deploy, etc.
<code>runs-on:</code>	Runs your job on an Ubuntu virtual machine.
<code>matrix.node-version</code>	Allows you to test against multiple versions of Node.js (optional).
<code>checkout</code>	Pulls your code from GitHub into the runner.
<code>setup-node</code>	Installs the version of Node you want.
<code>npm install</code>	Installs all dependencies.
<code>npm test</code>	Runs your test script from <code>package.json</code> .

How to See It in Action

Once you **push this file** to your `main` branch, GitHub will **automatically run this workflow**.

You can see the results:

1. Go to your repo on GitHub.
 2. Click on the **Actions** tab.
 3. See the workflow run logs in real-time.
-