

# Програмні складові Node-RED.

---

## Використання принципів автоматичного тестування

---

Node-Red являє собою звичайне Node.JS application. Тому, для його тестування спроуємо принципи test-driven development

### Використовуємо такі бібліотеки

```
"chai": "^4.2.0",
"chai-http": "^4.3.0",
"jest": "^25.1.0",
"mocha": "^7.0.1",
"nock": "^11.8.2",
"sinon": "^8.1.1",
"sinon-chai": "^3.4.0",
"supertest": "^4.0.2"
```

Для підготовки тестових кейсів використовуємо 3 потоки, що дозволяють отримати поняття по автоматизоване тестування:

- простий http get
- http get з параметрами
- http post

Набір потоків знаходиться у файлі: [Flows/flow-test-cases.json](#). Для тестування потрібно створити тестове app

### Створення тестового application

```
## створення основних копопнентів
npm install init

## накачка пакетів
npm install chai
npm install chai-http
npm install jest
npm install mocha
npm install nock
npm install sinon
npm install sinon-chai
npm install supertest
```

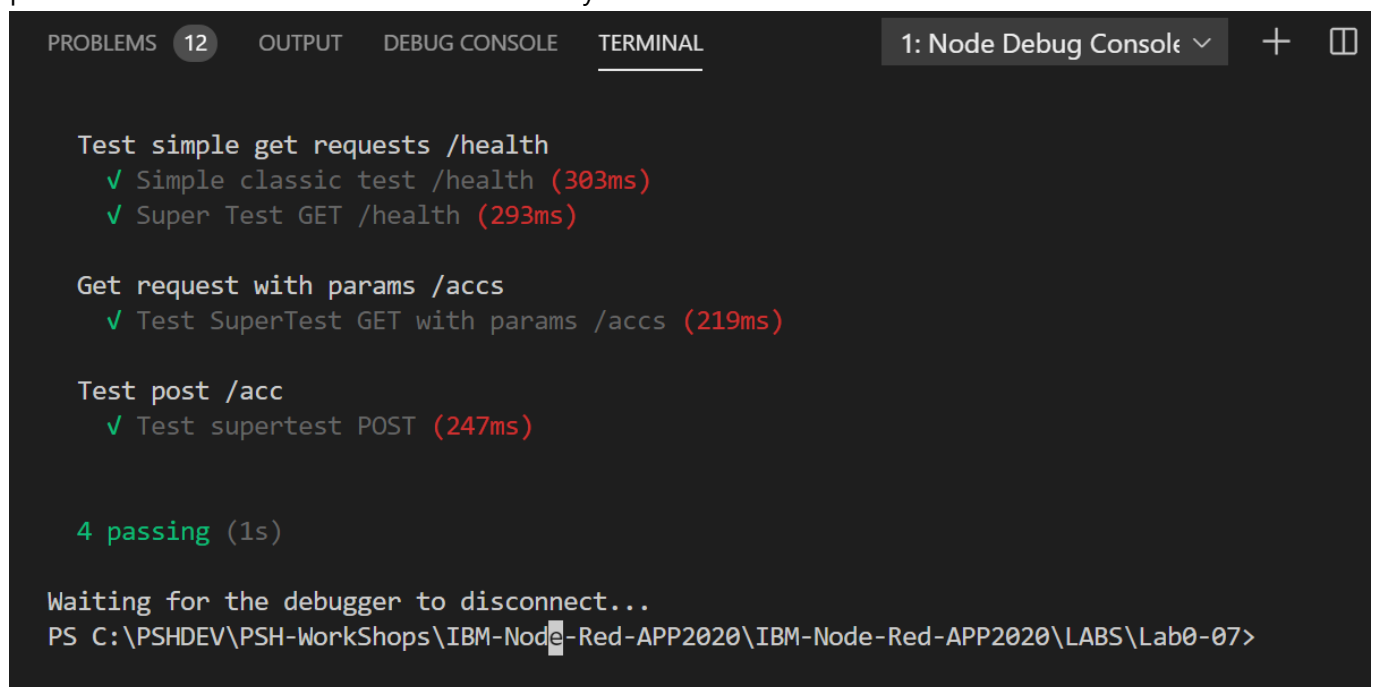
Створити каталог `test` і в нього покласти тестовий файл: [test/test-get-health.js](#)

В каталозі **.vscode** знаходиться **launch.json**, що дозволяє запустити тестовий кейс в режимі відладки.

Використовуємо такі інструменти

```
const mocha = require('mocha');
const chai = require('chai');
const request = require('supertest');
```

За допомогою `chai` та `supertest` можна легко побудувати серію тестових `http` запитів. Приклад логів роботи тестових кейсів показана на зображенні.



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL 1: Node Debug Console + □

Test simple get requests /health
  ✓ Simple classic test /health (303ms)
  ✓ Super Test GET /health (293ms)

Get request with params /accs
  ✓ Test SuperTest GET with params /accs (219ms)

Test post /acc
  ✓ Test supertest POST (247ms)

4 passing (1s)

Waiting for the debugger to disconnect...
PS C:\PSHDEV\PSH-WorkShops\IBM-Node-Red-APP2020\IBM-Node-Red-APP2020\LABS\Lab0-07>
```

Потрібно розуміти, що точність тестових кейсів залежить від початкового запиту. В процесі розробки тестові кейси можуть деалізуватися та розширбватися. Це і називається `test-driven development`.