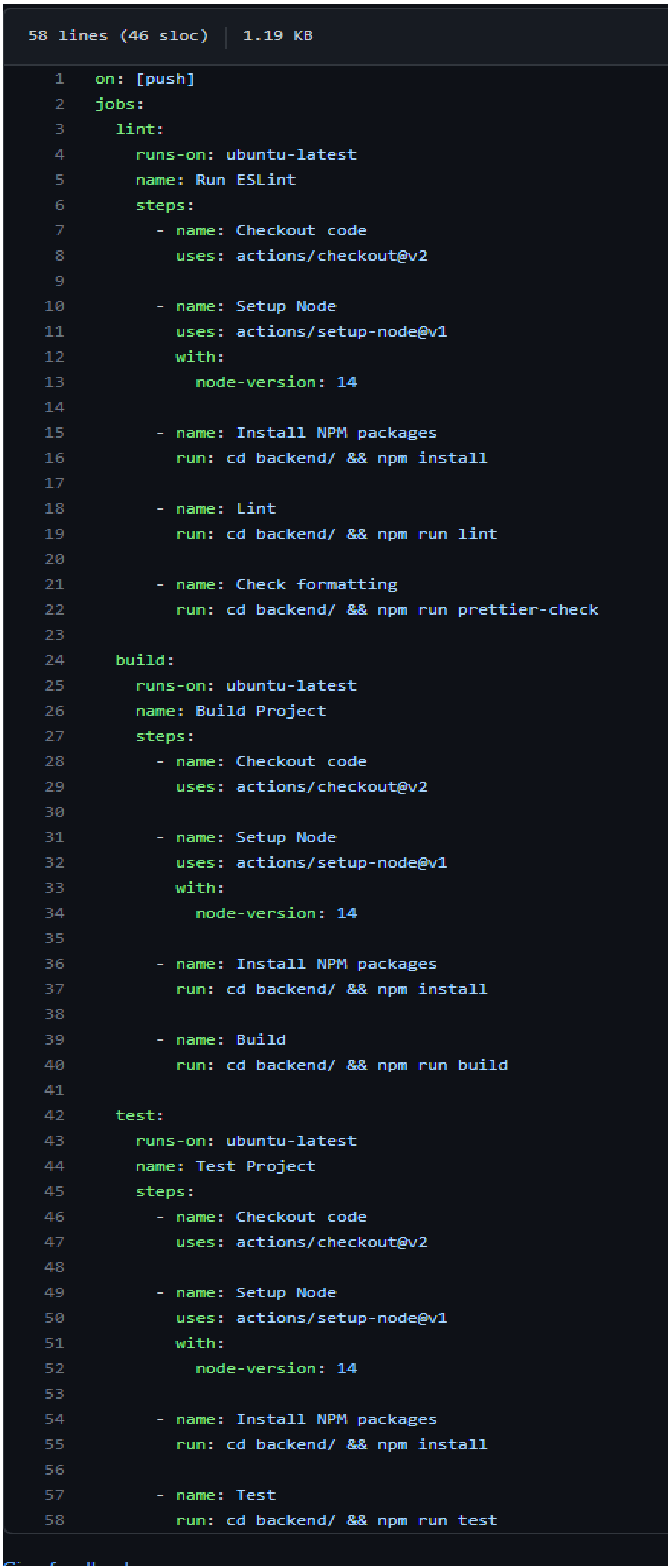
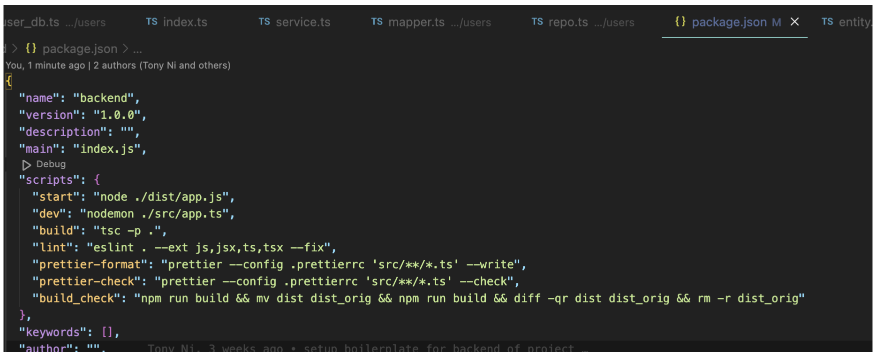


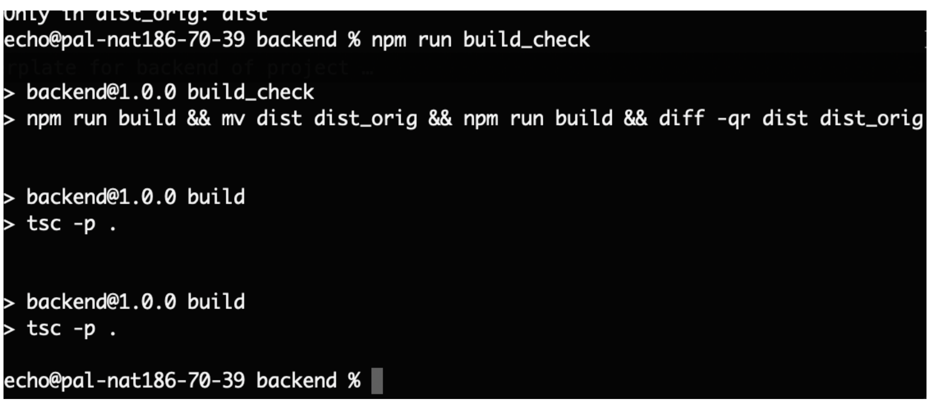
58 lines (46 sloc) | 1.19 KB

```yaml
on: [push]
jobs:
  lint:
    runs-on: ubuntu-latest
    name: Run ESLint
    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Setup Node
        uses: actions/setup-node@v1
        with:
          node-version: 14

      - name: Install NPM packages
        run: cd backend/ && npm install

      - name: Lint
        run: cd backend/ && npm run lint

      - name: Check formatting
        run: cd backend/ && npm run prettier-check

  build:
    runs-on: ubuntu-latest
    name: Build Project
    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Setup Node
        uses: actions/setup-node@v1
        with:
          node-version: 14

      - name: Install NPM packages
        run: cd backend/ && npm install

      - name: Build
        run: cd backend/ && npm run build

  test:
    runs-on: ubuntu-latest
    name: Test Project
    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Setup Node
        uses: actions/setup-node@v1
        with:
          node-version: 14

      - name: Install NPM packages
        run: cd backend/ && npm install

      - name: Test
        run: cd backend/ && npm run test
```

Our CI uses github actions to check our codebase. We have 3 jobs right now:  lint, build, and test. The linter uses eslint to lint our code and then uses checks if it is following our prettier.rc's format. For the build we just build our project using tsc which should compile our typescript code into javascript. For the test use are testing using jest and we will have a test directory that will implement both unit tests and integrations test.



```
ser_db.ts .../users   TS index.ts   TS service.ts   TS mapper.ts .../users   TS repo.ts .../users   {} package.json M ×   TS entity
> {} package.json > ...
You, 1 minute ago | 2 authors (Tony Ni and others)
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "start": "node ./dist/app.js",
    "dev": "nodemon ./src/app.ts",
    "build": "tsc -p .",
    "lint": "eslint . --ext js,jsx,ts,tsx --fix",
    "prettier-format": "prettier --config .prettierrc 'src/**/*.ts' --write",
    "prettier-check": "prettier --config .prettierrc 'src/**/*.ts' --check",
    "build_check": "npm run build && mv dist dist_orig && npm run build && diff -qr dist dist_orig && rm -r dist_orig"
  },
  "keywords": [],
  "author": "",        Tony Ni, 3 weeks ago • setup boilerplate for backend of project ...
```



```
Only in dist_orig: dist
echo@pal-nat186-70-39 backend % npm run build_check

> backend@1.0.0 build_check
> npm run build && mv dist dist_orig && npm run build && diff -qr dist dist_orig


> backend@1.0.0 build
> tsc -p .


> backend@1.0.0 build
> tsc -p .

echo@pal-nat186-70-39 backend % 
```

As of right now our build is reproducible as two different build instances yielded the same javascript build directory. The only way we could improve on our development process to maintain this is by using clean-installs for all of our node packages to ensure that their installation is not dependent on any files previously held on our development systems.