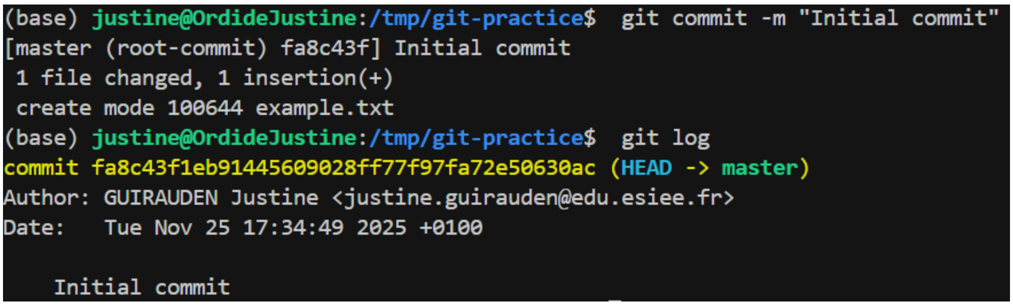
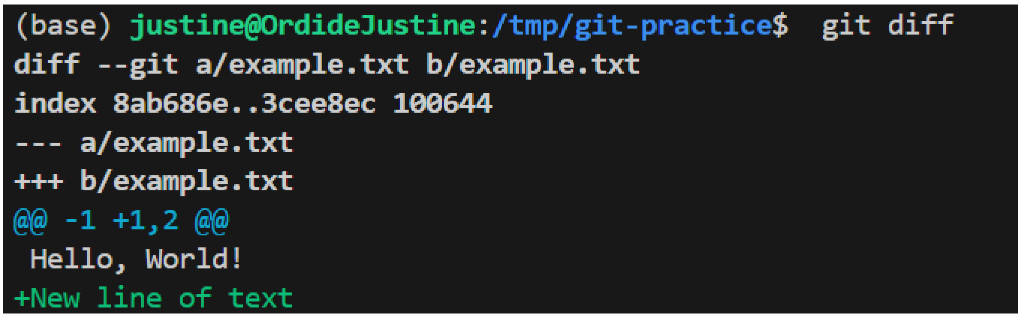
DevOps - Justine Guirauden and Volcy Desmazures

Lab 4:

git config --global user.name "GUIRAUDEN Justine"
git config --global user.email "justine.guirauden@edu.esiee.fr"
git init
git remote add origin https://github.com/guiraudj/devops
git add README.md
git commit -m "Add README"
git push origin update-readme

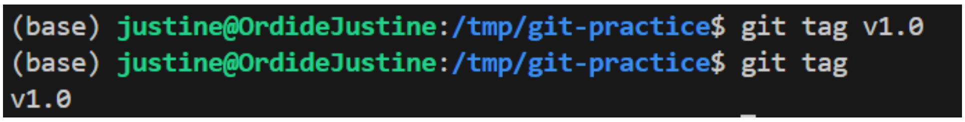


```
(base) justine@OrdideJustine:/tmp/git-practice$  git commit -m "Initial commit"
[master (root-commit) fa8c43f] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 example.txt
(base) justine@OrdideJustine:/tmp/git-practice$  git log
commit fa8c43f1eb91445609028ff77f97fa72e50630ac (HEAD -> master)
Author: GUIRAUDEN Justine <justine.guirauden@edu.esiee.fr>
Date:   Tue Nov 25 17:34:49 2025 +0100

    Initial commit
```



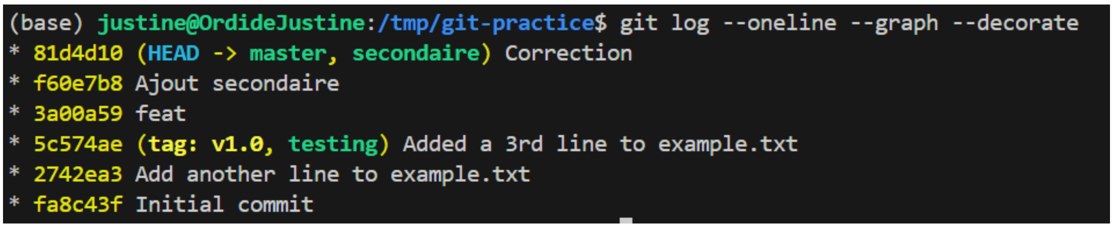
```
(base) justine@OrdideJustine:/tmp/git-practice$  git diff
diff --git a/example.txt b/example.txt
index 8ab686e..3cee8ec 100644
--- a/example.txt
+++ b/example.txt
@@ -1 +1,2 @@
 Hello, World!
+New line of text
```

Exercise 1:



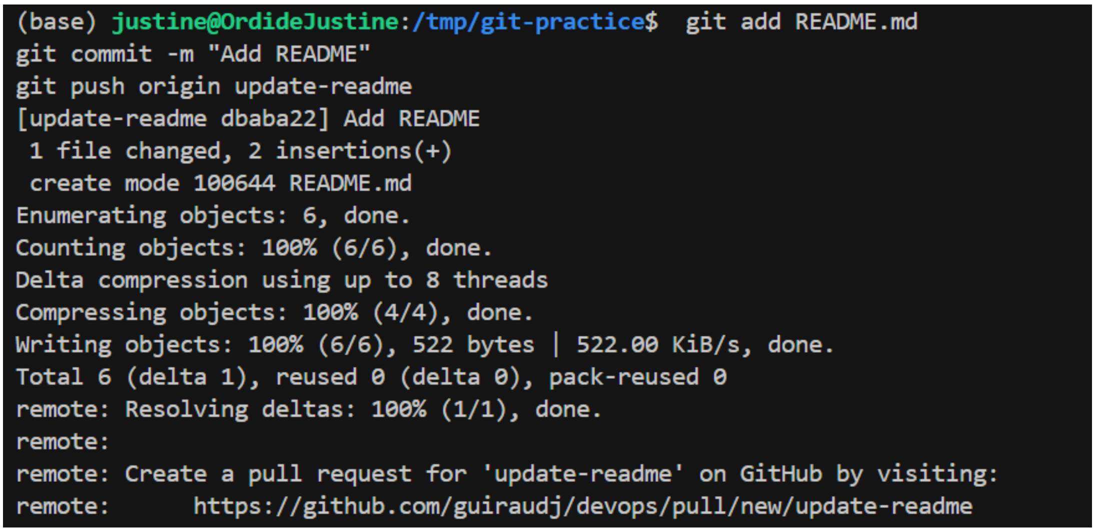
```
(base) justine@OrdideJustine:/tmp/git-practice$ git tag v1.0
(base) justine@OrdideJustine:/tmp/git-practice$ git tag
v1.0
```

Exercise 2:

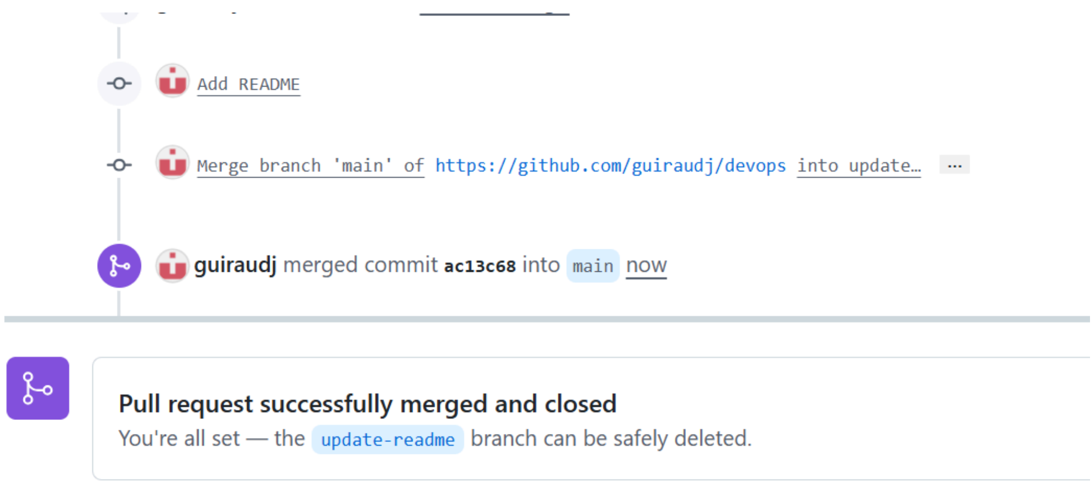


```
(base) justine@OrdideJustine:/tmp/git-practice$ git log --oneline --graph --decorate
* 81d4d10 (HEAD -> master, secondaire) Correction
* f60e7b8 Ajout secondaire
* 3a00a59 feat
* 5c574ae (tag: v1.0, testing) Added a 3rd line to example.txt
* 2742ea3 Add another line to example.txt
* fa8c43f Initial commit
```

git merge preserves the actual history by creating a merge commit. The history becomes a graph (non-linear).
git rebase rewrites history by moving and recreating the feature branch's commits onto the target branch. The history becomes a straight line (linear and clean).



```
(base) justine@OrdideJustine:/tmp/git-practice$  git add README.md
git commit -m "Add README"
git push origin update-readme
[update-readme dbaba22] Add README
 1 file changed, 2 insertions(+)
 create mode 100644 README.md
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 522 bytes | 522.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'update-readme' on GitHub by visiting:
remote:      https://github.com/guiraudj/devops/pull/new/update-readme
```



Add README

Merge branch 'main' of https://github.com/guiraudj/devops into update… ···

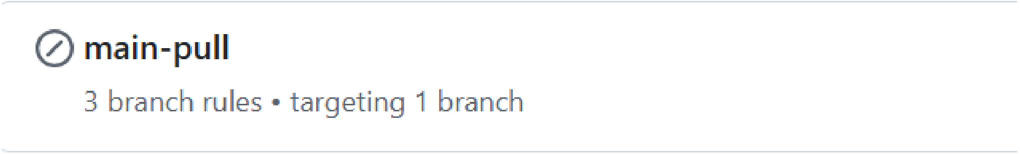
guiraudj merged commit ac13c68 into main now

Pull request successfully merged and closed

You're all set — the update-readme branch can be safely deleted.

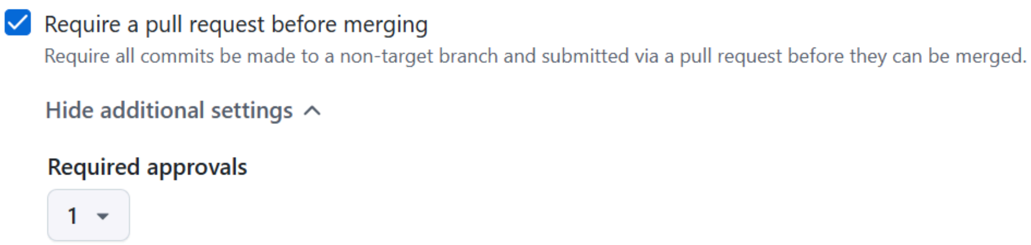
The pull request successfully went.

Exercise 3:



main-pull

3 branch rules • targeting 1 branch



☑ Require a pull request before merging

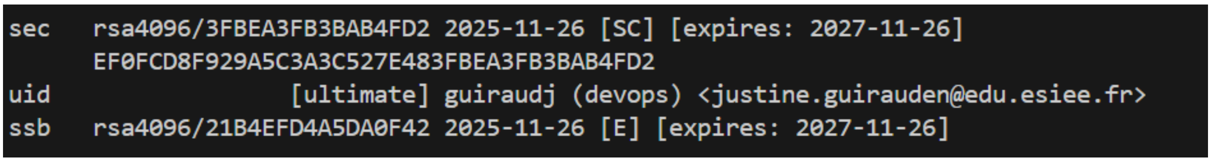
Require all commits be made to a non-target branch and submitted via a pull request before they can be merged.

Hide additional settings ^

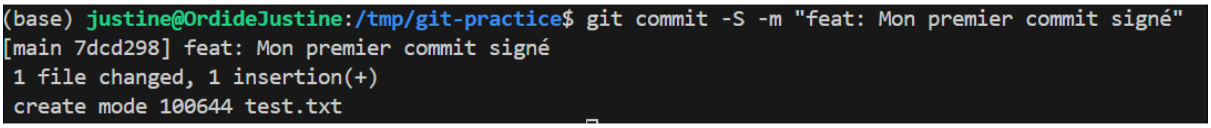
Required approvals

1 ▾

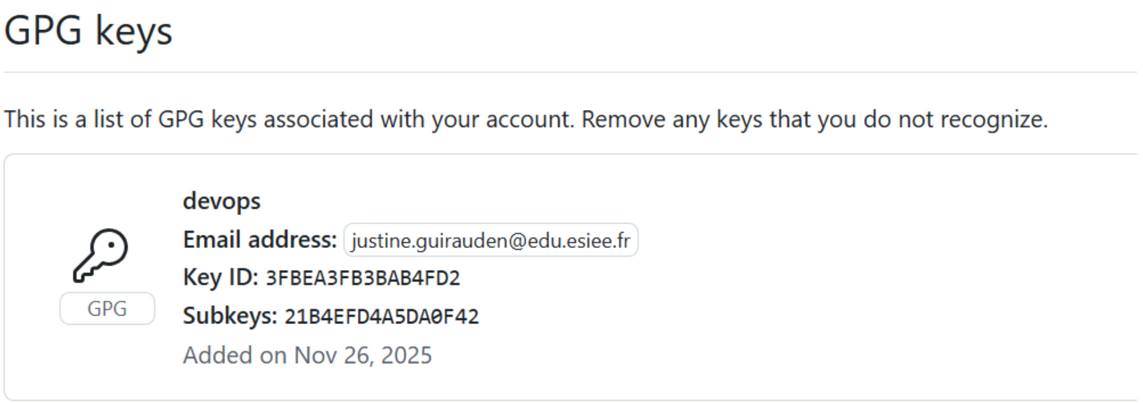
Exercise 4:



```
sec   rsa4096/3FBEA3FB3BAB4FD2 2025-11-26 [SC] [expires: 2027-11-26]
      EF0FCD8F929A5C3A3C527E483FBEA3FB3BAB4FD2
uid                 [ultimate] guiraudj (devops) <justine.guirauden@edu.esiee.fr>
ssb   rsa4096/21B4EFD4A5DA0F42 2025-11-26 [E] [expires: 2027-11-26]
```



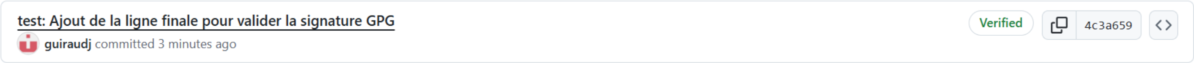
```
(base) justine@OrdideJustine:/tmp/git-practice$ git commit -S -m "feat: Mon premier commit signé"
[main 7dcd298] feat: Mon premier commit signé
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
```



GPG keys

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.

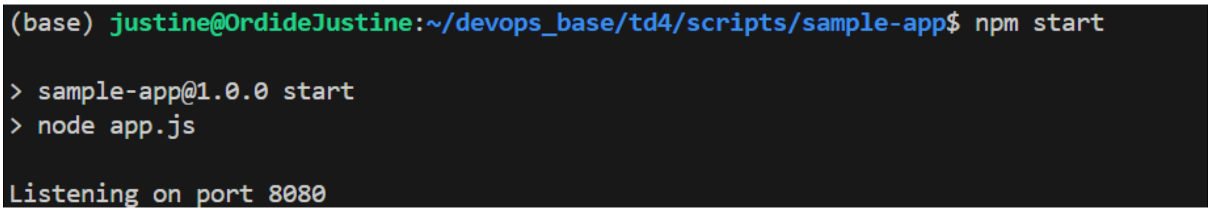
devops
Email address: justine.guirauden@edu.esiee.fr
Key ID: 3FBEA3FB3BAB4FD2
Subkeys: 21B4EFD4A5DA0F42
Added on Nov 26, 2025



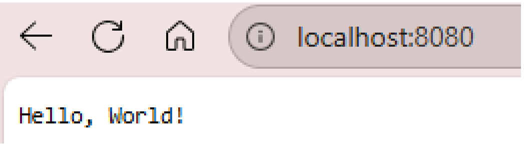
test: Ajout de la ligne finale pour valider la signature GPG
guiraudj committed 3 minutes ago

Section 3: Setting Up a Build System with NPM

NPM:

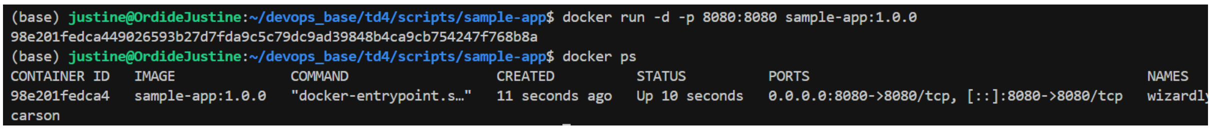


```
(base) justine@OrdideJustine:~/devops_base/td4/scripts/sample-app$ npm start

> sample-app@1.0.0 start
> node app.js

Listening on port 8080
```

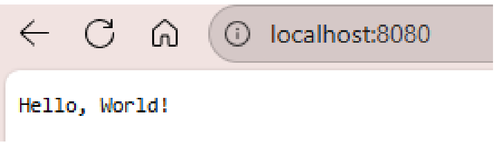


Hello, World!

Docker:



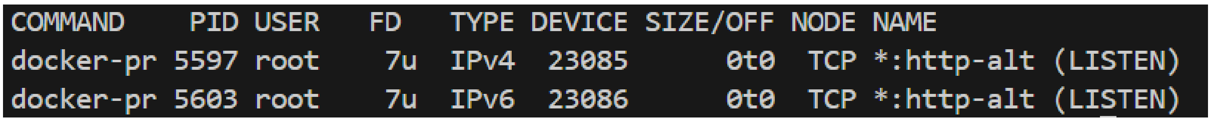
```
(base) justine@OrdideJustine:~/devops_base/td4/scripts/sample-app$ docker run -d -p 8080:8080 sample-app:1.0.0
98e201fedca449026593b27d7fda9c5c79dc9ad39848b4ca9cb754247f768b8a
(base) justine@OrdideJustine:~/devops_base/td4/scripts/sample-app$ docker ps
CONTAINER ID   IMAGE              COMMAND                   CREATED          STATUS          PORTS                                         NAMES
98e201fedca4   sample-app:1.0.0   "docker-entrypoint.s…"   11 seconds ago   Up 10 seconds   0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp   wizardly
carson
```



Hello, World!

Exercise 5:

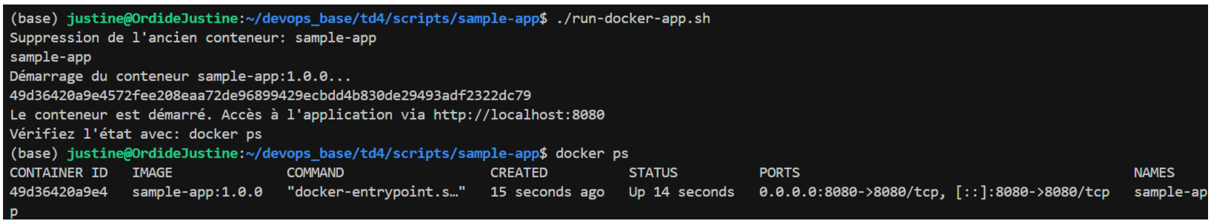
We modified Dockerfile content with a specific patch version tag (21.7.0) instead of tag (21.7).



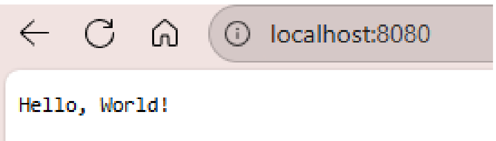
```
COMMAND     PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
docker-pr 5597 root    7u  IPv4  23085      0t0  TCP *:http-alt (LISTEN)
docker-pr 5603 root    7u  IPv6  23086      0t0  TCP *:http-alt (LISTEN)
```

Pinning the version ensures **reproducibility**. It guarantees that the exact same Node.js environment is used for every build, preventing unexpected bugs or behavioral changes from automatic updates.

Exercise 6:

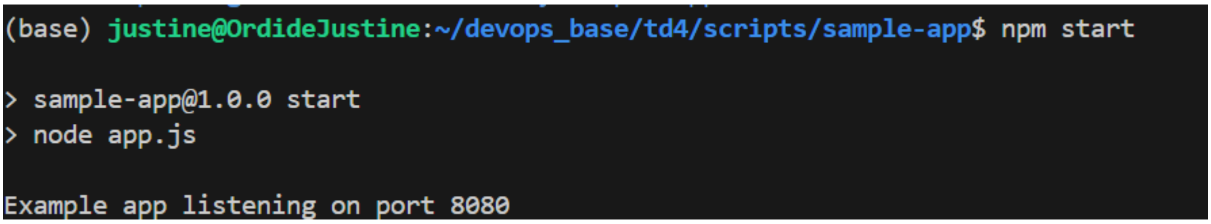


```
(base) justine@OrdideJustine:~/devops_base/td4/scripts/sample-app$ ./run-docker-app.sh
Suppression de l'ancien conteneur: sample-app
sample-app
Démarrage du conteneur sample-app:1.0.0...
49d36420a9e4572fee208eaa72de96899429ecbdd4b830de29493adf2322dc79
Le conteneur est démarré. Accès à l'application via http://localhost:8080
Vérifiez l'état avec: docker ps
(base) justine@OrdideJustine:~/devops_base/td4/scripts/sample-app$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                         NAMES
49d36420a9e4   sample-app:1.0.0   "docker-entrypoint.s…"   15 seconds ago   Up 14 seconds   0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp   sample-ap
p
```



```
Hello, World!
```

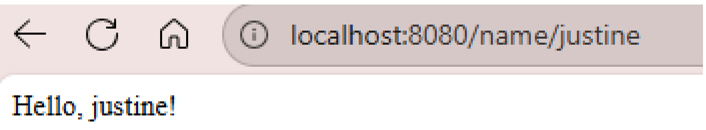
Section 4: Managing Dependencies with NPM



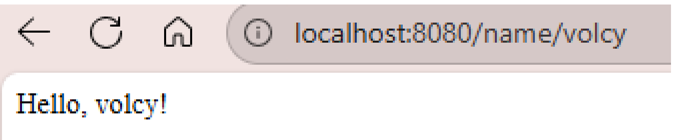
```
(base) justine@OrdideJustine:~/devops_base/td4/scripts/sample-app$ npm start

> sample-app@1.0.0 start
> node app.js

Example app listening on port 8080
```

Using the dependency express and npm ci --only=production in the Dockerfile significantly speeds up the image build time by avoiding the installation of all unnecessary development dependencies.

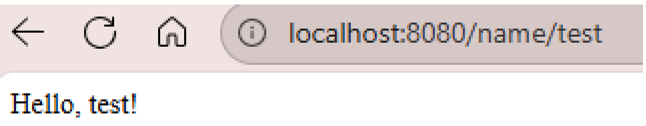
Exercise 7:



Hello, justine!



Hello, volcy!



Hello, test!

Exercise 8:

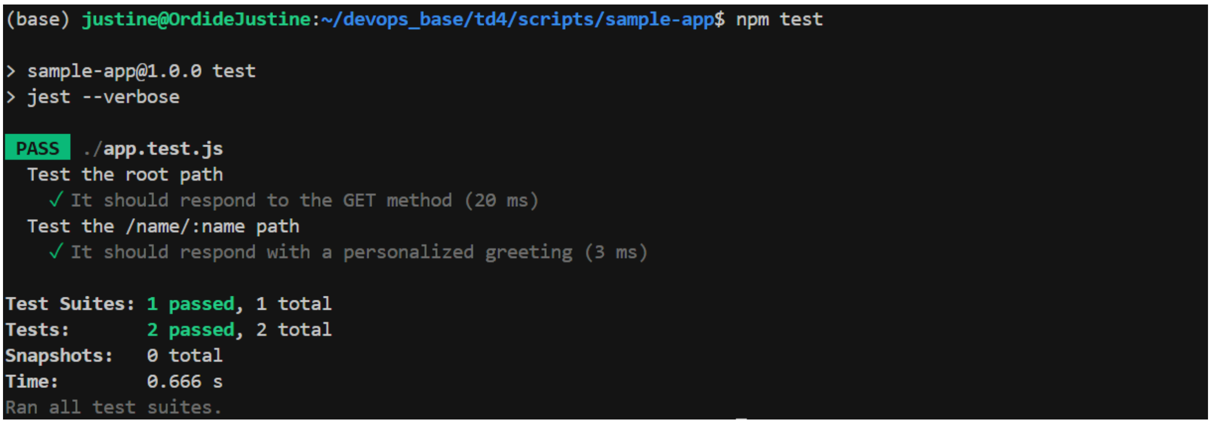
To add a development dependency (e.g., Mocha), we use npm install mocha --save-dev.

Differences:

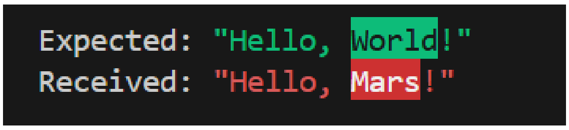
- dependencies are essential packages required for the application to run in production.
- devDependencies are packages only needed for development tasks (testing, linting, or building).

Importance for Docker: Separating them allows us to use npm ci --only=production in the Dockerfile to dramatically reduce the final image size and attack surface by excluding unnecessary development tools.

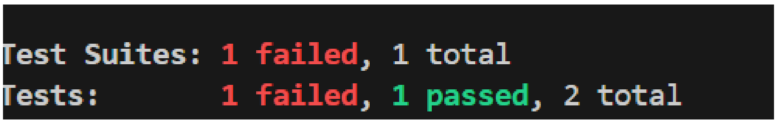
Section 5: Automated Testing



```
(base) justine@OrdideJustine:~/devops_base/td4/scripts/sample-app$ npm test

> sample-app@1.0.0 test
> jest --verbose

 PASS  ./app.test.js
  Test the root path
    ✓ It should respond to the GET method (20 ms)
  Test the /name/:name path
    ✓ It should respond with a personalized greeting (3 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.666 s
Ran all test suites.
```

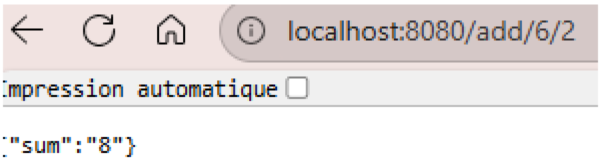


```
Expected: "Hello, World!"
Received: "Hello, Mars!"
```



```
Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 passed, 2 total
```

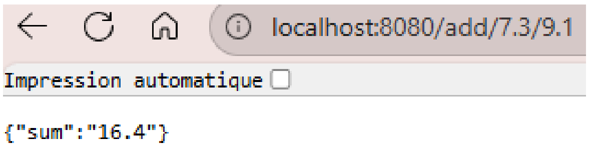
Exercise 9:



localhost:8080/add/6/2

Impression automatique □

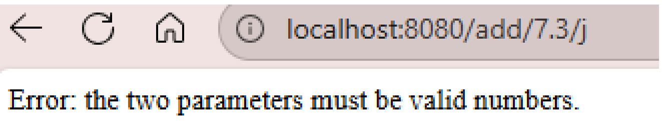
```
{"sum":"8"}
```



localhost:8080/add/7.3/9.1

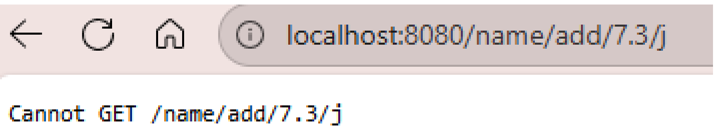
Impression automatique □

```
{"sum":"16.4"}
```



localhost:8080/add/7.3/j

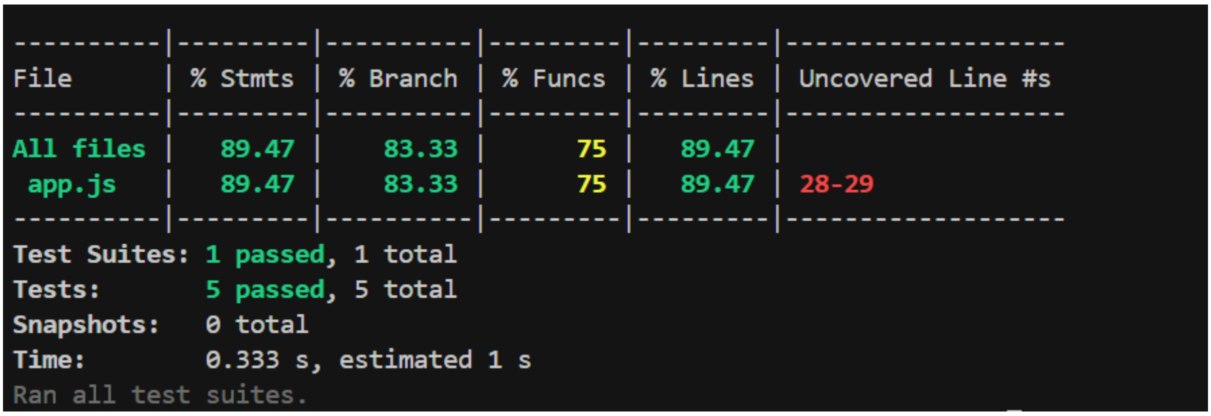
Error: the two parameters must be valid numbers.



localhost:8080/name/add/7.3/j

```
Cannot GET /name/add/7.3/j
```

Exercise 10:

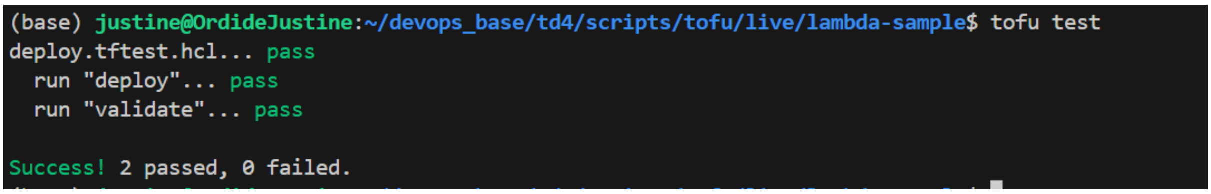


```
----------|---------|----------|---------|---------|-------------------
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------|---------|----------|---------|---------|-------------------
All files |   89.47 |    83.33 |      75 |   89.47 |
 app.js   |   89.47 |    83.33 |      75 |   89.47 | 28-29
----------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        0.333 s, estimated 1 s
Ran all test suites.
```

To implement code coverage, we added "test:coverage": "jest --coverage" to package.json.

Our analysis shows a high coverage of 89.47%. The only uncovered lines (28-29) correspond to the server startup code (app.listen), which is expected because supertest tests the application logic directly without binding to a network port.
This result confirms that 100% of our actual business logic (endpoints and calculations) is tested. Monitoring coverage is crucial as it reveals untested paths and dead code, providing confidence that refactoring won't break existing features.

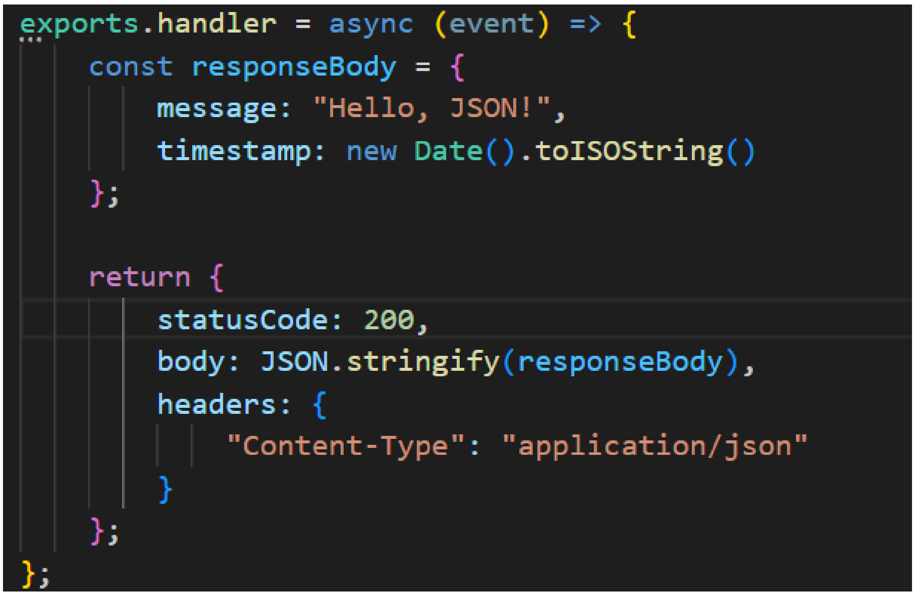
Section 6: Automated Testing for OpenTofu Code



```
(base) justine@OrdideJustine:~/devops_base/td4/scripts/tofu/live/lambda-sample$ tofu test
deploy.tftest.hcl... pass
  run "deploy"... pass
  run "validate"... pass

Success! 2 passed, 0 failed.
```

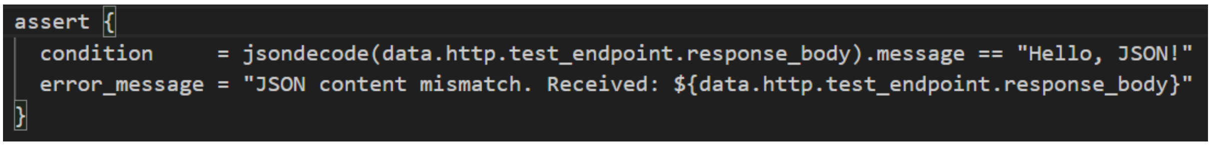
Exercise 11:

Modifications of index.js:



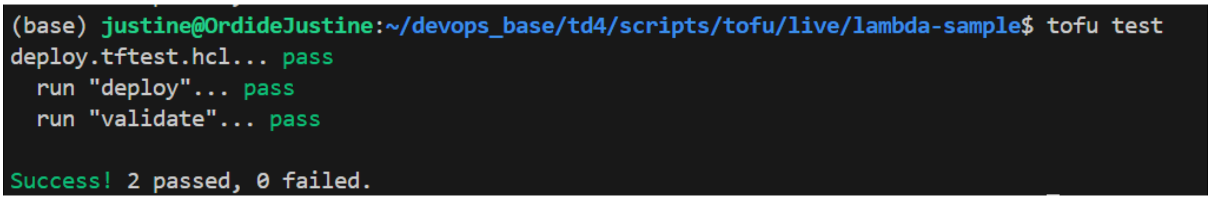
```
exports.handler = async (event) => {
    const responseBody = {
        message: "Hello, JSON!",
        timestamp: new Date().toISOString()
    };

    return {
        statusCode: 200,
        body: JSON.stringify(responseBody),
        headers: {
            "Content-Type": "application/json"
        }
    };
};
```

Modifications of deploy.tftest.hcl:



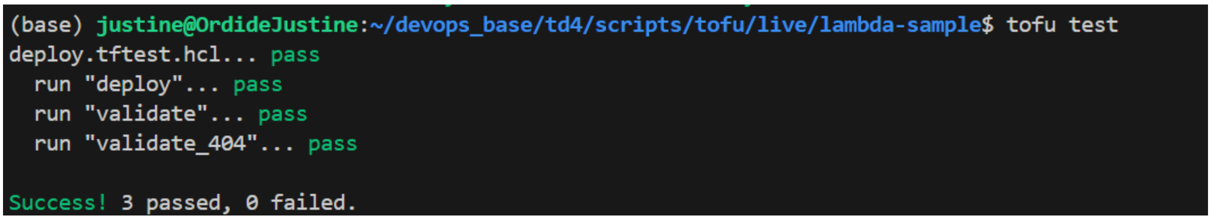
```
assert {
  condition     = jsondecode(data.http.test_endpoint.response_body).message == "Hello, JSON!"
  error_message = "JSON content mismatch. Received: ${data.http.test_endpoint.response_body}"
}
```

tofu test went correctly:



```
(base) justine@OrdideJustine:~/devops_base/td4/scripts/tofu/live/lambda-sample$ tofu test
deploy.tftest.hcl... pass
  run "deploy"... pass
  run "validate"... pass

Success! 2 passed, 0 failed.
```

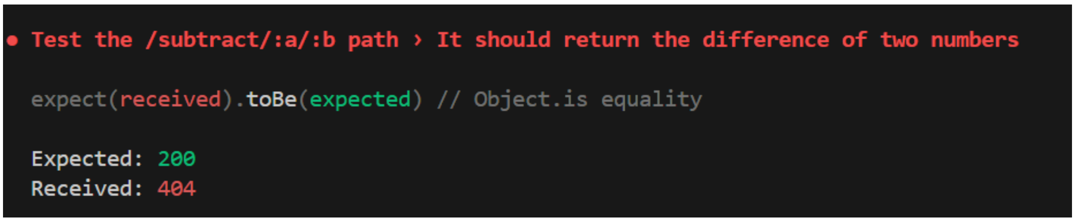
Exercise 12:



```
(base) justine@OrdideJustine:~/devops_base/td4/scripts/tofu/live/lambda-sample$ tofu test
deploy.tftest.hcl... pass
  run "deploy"... pass
  run "validate"... pass
  run "validate_404"... pass

Success! 3 passed, 0 failed.
```

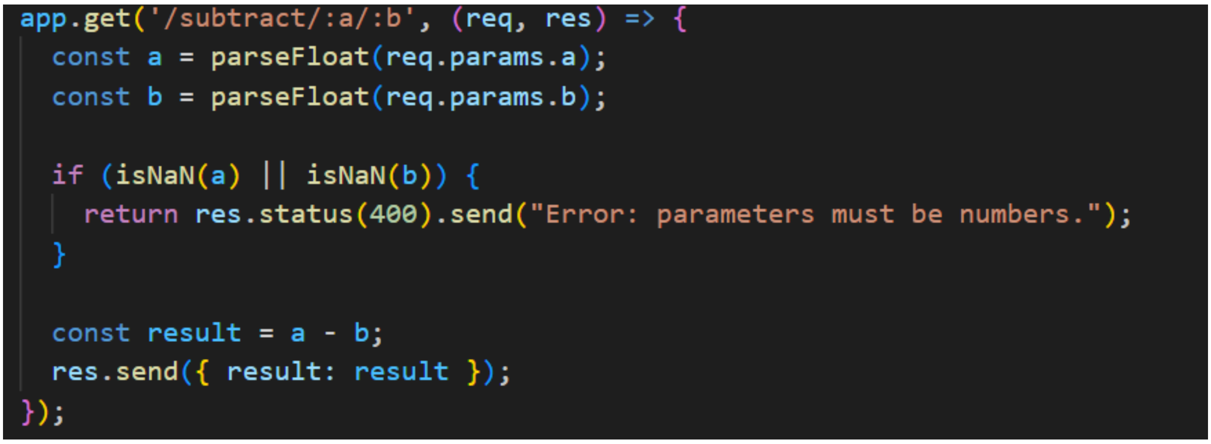
Section 7: Testing Recommendations

First, we write the test for a function of subtraction of 2 numbers in app.test.js so it fails (because it is not yet implemented):



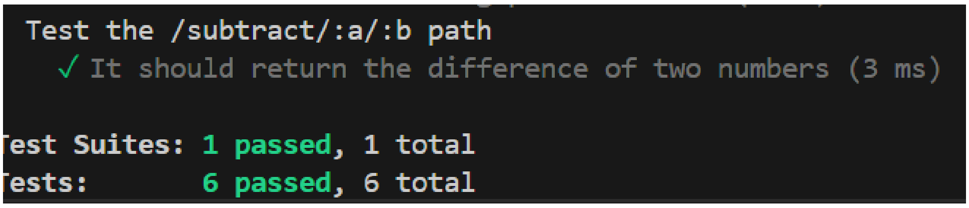
```
● Test the /subtract/:a/:b path › It should return the difference of two numbers

  expect(received).toBe(expected) // Object.is equality

  Expected: 200
  Received: 404
```

Then we implement the feature in app.js:



```
app.get('/subtract/:a/:b', (req, res) => {
  const a = parseFloat(req.params.a);
  const b = parseFloat(req.params.b);

  if (isNaN(a) || isNaN(b)) {
    return res.status(400).send("Error: parameters must be numbers.");
  }

  const result = a - b;
  res.send({ result: result });
});
```

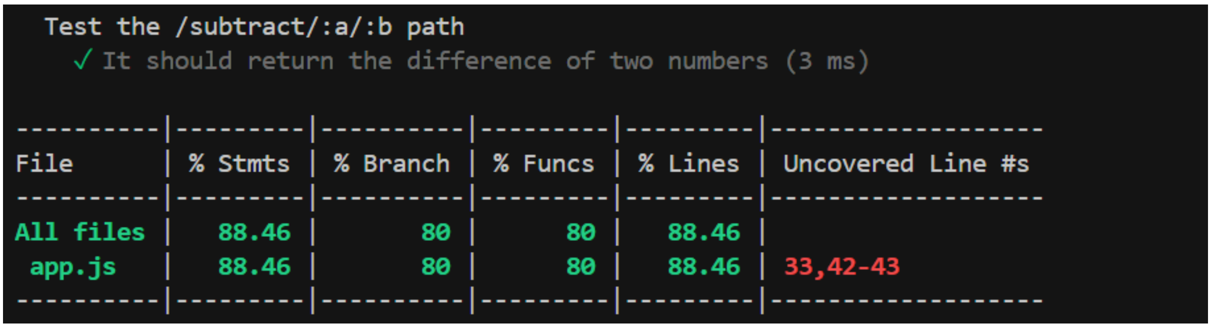
The test for subtraction went well:



```
Test the /subtract/:a/:b path
  ✓ It should return the difference of two numbers (3 ms)

Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
```

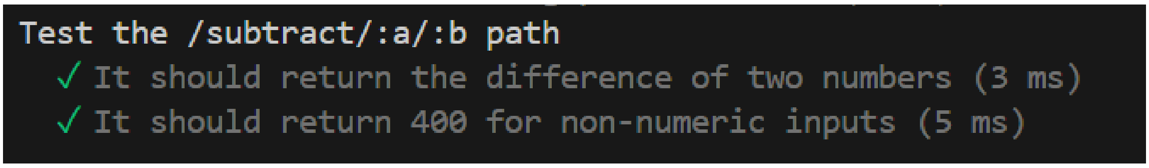
Exercise 14:

We analyzed the code coverage and identified a gap. While we implemented the subtract feature using TDD, we only wrote a test for the successful subtraction. The error handling logic inside the if (isNaN...) block remains untested. To improve coverage, we need to add a negative test case that sends invalid inputs to the /subtract endpoint to verify it correctly returns a 400 error.

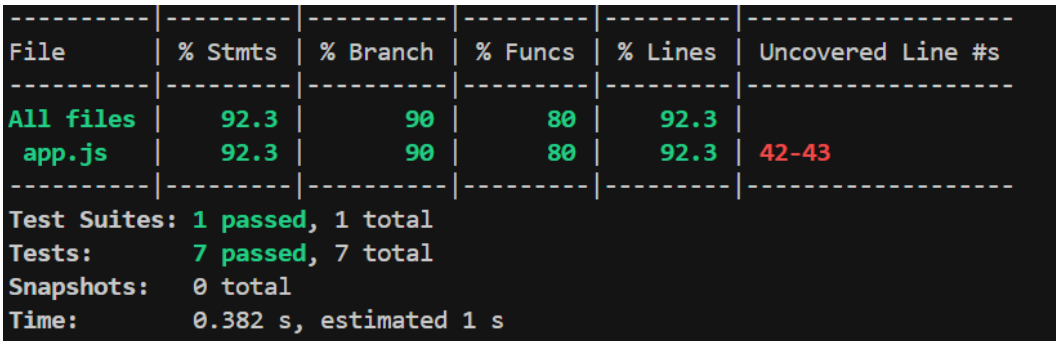


```
  Test the /subtract/:a/:b path
    ✓ It should return the difference of two numbers (3 ms)

----------|---------|----------|---------|---------|-------------------
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------|---------|----------|---------|---------|-------------------
All files |   88.46 |       80 |      80 |   88.46 |
 app.js   |   88.46 |       80 |      80 |   88.46 | 33,42-43
----------|---------|----------|---------|---------|-------------------
```

So we fix this case:



```
Test the /subtract/:a/:b path
  ✓ It should return the difference of two numbers (3 ms)
  ✓ It should return 400 for non-numeric inputs (5 ms)
```



```
----------|---------|----------|---------|---------|-------------------
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------|---------|----------|---------|---------|-------------------
All files |    92.3 |       90 |      80 |    92.3 |
 app.js   |    92.3 |       90 |      80 |    92.3 | 42-43
----------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        0.382 s, estimated 1 s
```

The only uncovered lines (42-43) are the server startup lines (app.listen), which is the normal and expected behavior with these testing tools