

JS QA automation framework for API

Table of Contents

1 Test automation theory.....	3
2 Linux basics.....	3
3 Linux basics practice.....	3
4 SSH keys.....	3
5 Test project setup.....	5
6 Syntax for common test runners / Work with a new branch. Install BABEL. Pull request. First tests.....	8
7 Syntax for HTTP client / First API tests.....	13
8 Environment variables.....	13
9 Test runner (mocha) hooks.....	13
10 Wrappers for API tests.....	13
11 API tests practice.....	13
12 Setting up a mock server.....	13
13 CI/CD theory.....	13
14 Integrating API tests with GitHub Actions.....	13

1 Test automation theory

Lots of super boring stuff should be presented over here!

2 Linux basics

ls, cd, mkdir, cp, mv, rm, cat, nano, grep, regExp

3 Linux basics practice

ls, cd, mkdir, cp, mv, rm, cat, nano, grep, regExp

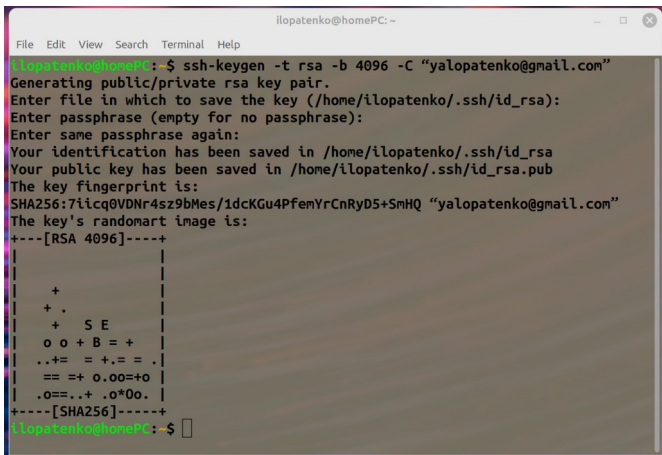
4 SSH keys

Open an terminal

CD to a user home directory

Run next command:

CLI=> `ssh-keygen -t rsa -b 4096 -C "ENTER_HERE_YOUR_EMAIL_OR_WHATEVER"`



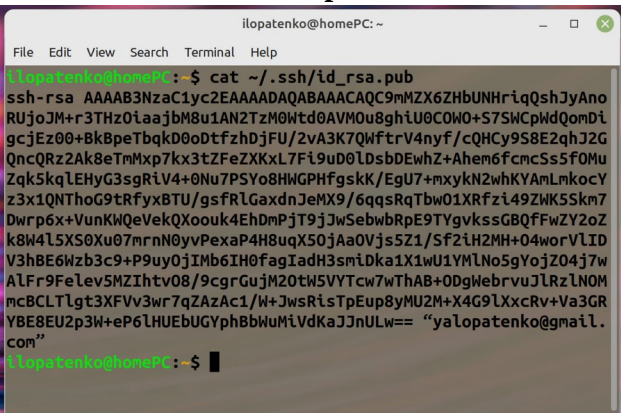
```
ilopatenko@homePC: ~  
$ ssh-keygen -t rsa -b 4096 -C "yalopatenko@gmail.com"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/ilopatenko/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ilopatenko/.ssh/id_rsa  
Your public key has been saved in /home/ilopatenko/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:7iicQ0VDNr4sz9bMes/1dcKGu4PfenYrCnRyD5+SmHQ "yalopatenko@gmail.com"  
The key's random image is:  
+---[RSA 4096]-----+  
+  
+ .  
+ S E  
+ o o + B = +  
..+ = +. = .  
== + 0.00 = + 0  
..o = .. + 0*00.  
+---[SHA256]-----+  
ilopatenko@homePC: $
```

Go to gitHub. Go to settings <https://github.com/settings/profile>

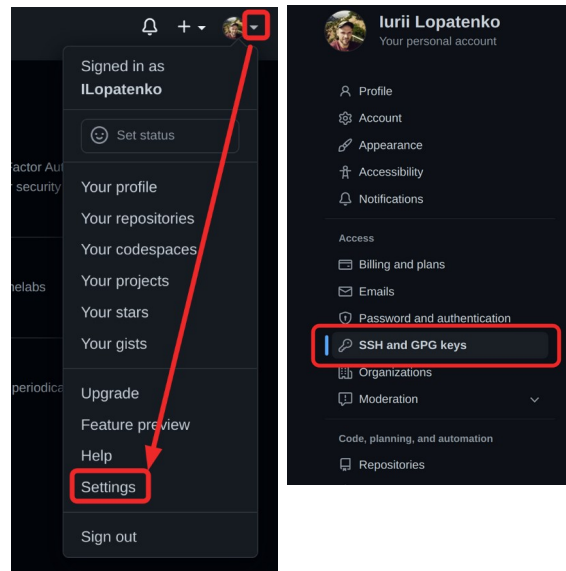
Go to SSH and GPG keys <https://github.com/settings/keys>

Copy SSH key to buffer:

CLI=> `cat ~/.ssh/id_rsa.pub`



```
ilopatenko@homePC: ~  
$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC9mMXZ6ZHbUNHriqQshJyAno  
RUjoJM+r3THz0iaaibM8u1AN2TzM0Wtd0AVM0u8ghiU0COW0+S7SWCpWdQomDi  
gcjEz00+Bk8PeTbqkD0oDtfzhDjFU/2vA3K7QWftrV4nyf/cQHCy9S8E2qhJ2G  
QncQRz2Ak8eTmMxp7kx3tZFeZXKxL7Fi9u0D0LdsbDEwhZ+Ahem6fcmcSs5f0Mu  
Zqk5kqLEHyG3sgRiV4+0Nu7PSYo8HWGPHfgskK/Egu7+mxykN2whKYAmLnkocY  
z3x1QNTHoG9tRfyxBTU/gsfRlGaxdnJeMX9/6qqsRqTbw01XRfzi49ZWK5SkM7  
Dwrp6x+VunKHQeVekQXoouk4EhDmPjT9jJwSebwRpe9TYgvksGBQFfwZY2oZ  
k8W4L5X50Xu07mrnN0yvPexaP4H8uqX50jAa0VjsZ1/Sf2iH2MH+04worVLID  
V3hBE6Wzb3c9+P9uy0jIMb6IH0fagIadH3smIDka1X1wU1YMLNo5gYojZ04j7w  
ALFr9FeLev5MZIhtv08/9cgrGuJM20tW5VYTcw7wThAB+0DgWebrvujLRzLNOM  
mcBCLTLgt3XFVv3wr7qZAzAc1/W+JwsRisTpEup8yMU2M+X4G9lXxcRv+Va3GR  
YBE8EU2p3W+eP6LHUEbUGYphBbWuMlVdKaJJnULW== "yalopatenko@gmail.  
com"  
ilopatenko@homePC: $
```



Click the button **Add SSH** and Insert an SSH key from buffer and click the button **Add SSH key**

and confirm (re enter your gitHub password).

SSH keys / Add new

Title

This is an SSH key for homePC

Key


```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAOC9mMZX6ZHbUNHriqQshJyAnoRUjoJM+r3THzOiaajbM8u1AN2TzM0Wtd
0AVMOu8ghiU0COWO+S7SWCpWdQomDigcjEz00+Bk8peTbqkD0oDtfzhDjFU/2vA3K7QWftrV4nyf/cQHCy9S8E2qhJ
2GQncQRz2Ak8eTmMxp7kx3tZFeZXKxL7Fi9uD0lDsbDEwhZ+Ahem6fcmcsS5fOMuZqk5kqlEHyG3sgRiV4+0Nu7PSYo
8HWGPHfqsK/EgU7+mxykN2whKYAmLmkocYz3x1QNTHoG9tRfyxBTU/gsfRIGaxdnJeMX9/6qqqsRqTbwO1XRfzi49Z
WK5SKm7Dwrrp6x+VunKWQeVekQXoouk4EhDmPjT9jJwSebwRpe9TYgvkssGBQfWZY2oZk8W4l5XS0Xu07mrnN0y
vPexaP4H8uqX5OjAaOVjs5Z1/Sf2IH2MH+O4worVlIDV3hBE6Wzb3c9+P9uyOjIMb6IH0fagladH3smiDka1X1wU1YMIN
o5gYojZO4j7wAlFr9Felev5MZlhtvO8/9cgrGujM2OtW5VYTCw7wThAB+ODgWebrvuJlRzINOMmcBCLTlgt3XFVv3wr7qZ
AzAc1/W+JwsRisTpEup8yMU2M+X4G9lXxcRv+Va3GRYBE8EU2p3W+eP6IHUEbUGYphBbWuMiVdKaJJnULw==
"yalopatenko@gmail.com"
```

Add SSH key

SSH keys

New SSH key

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



This is an SSH key for homePC

SHA256:7iicq0VDNr4sz9bMes/1dcKGu4PfemYrCnRyD5+SmHQ

Added on Feb 5, 2022

Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

5 Test project setup

Create a new gitHub repository for a new project (JS automation framework for API testing)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *


Repository name *

 ILOpatenko ▾ / QAA-API-framework ✓

Great repository names are short and memorable. Need inspiration? How about **solid-system**?

Description (optional)

JS test automation framework for API

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.


☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

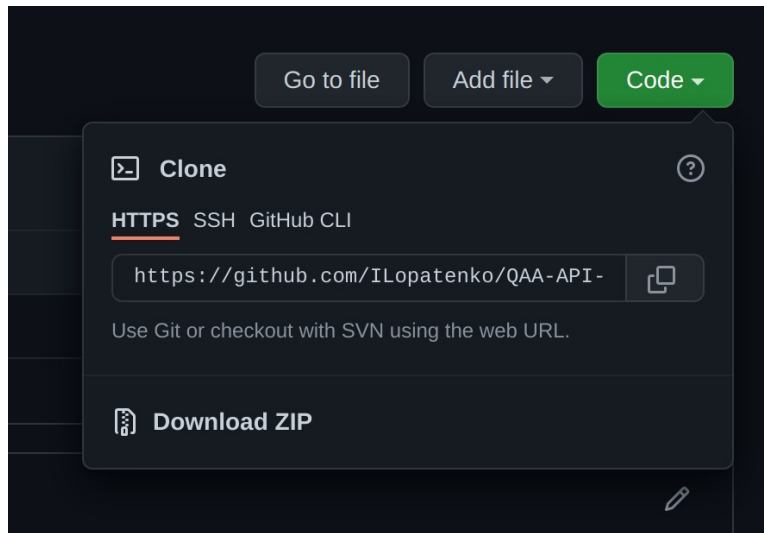
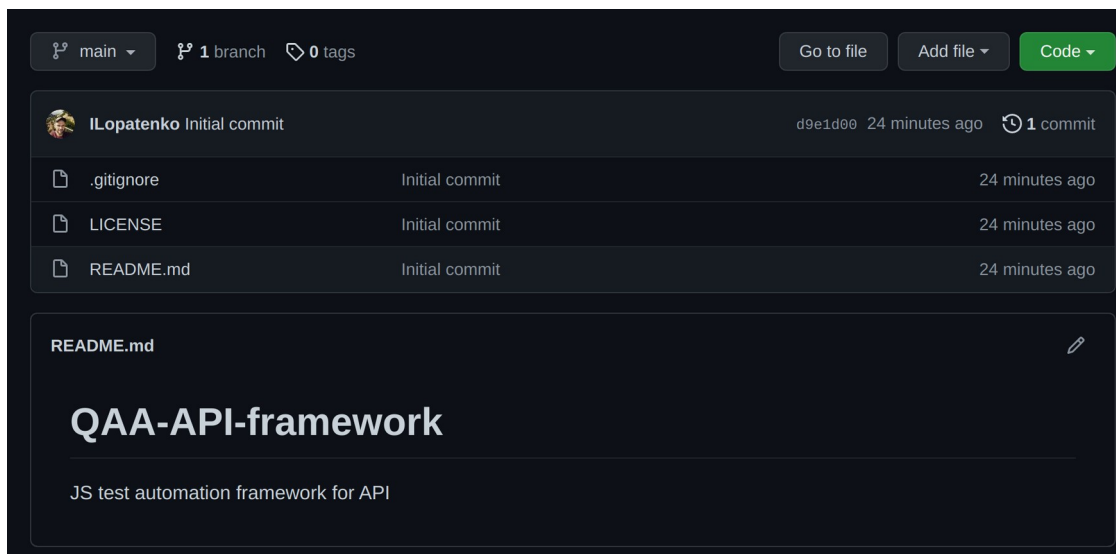
.gitignore template: Node ▾

☒ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

This will set  **main** as the default branch. Change the default name in your [settings](#).

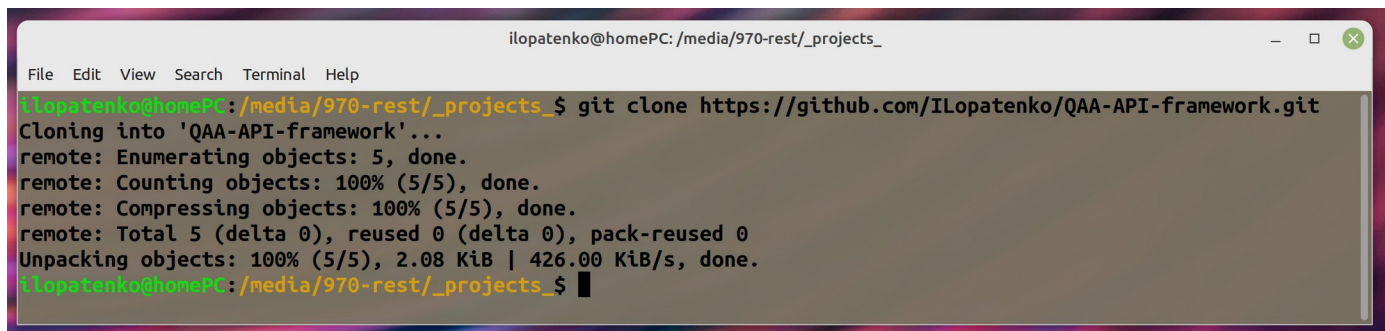
Create repository



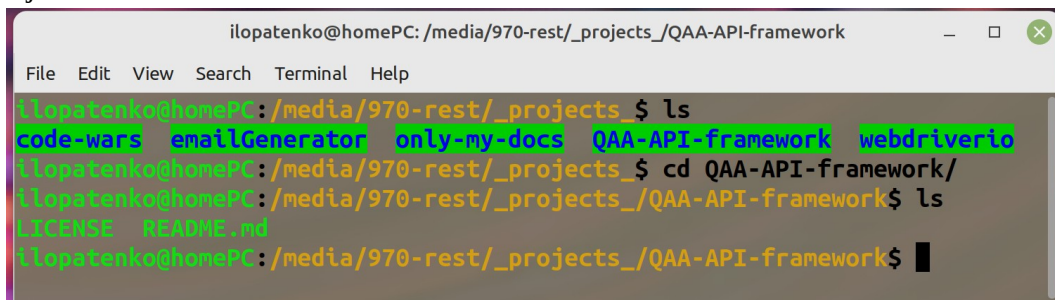
Copy your new repository's URL

Open terminal and **CD** to a directory where you want to keep your framework and clone it

CLI=> **git clone** <https://github.com/ILOpatenko/QAA-API-framework.git>

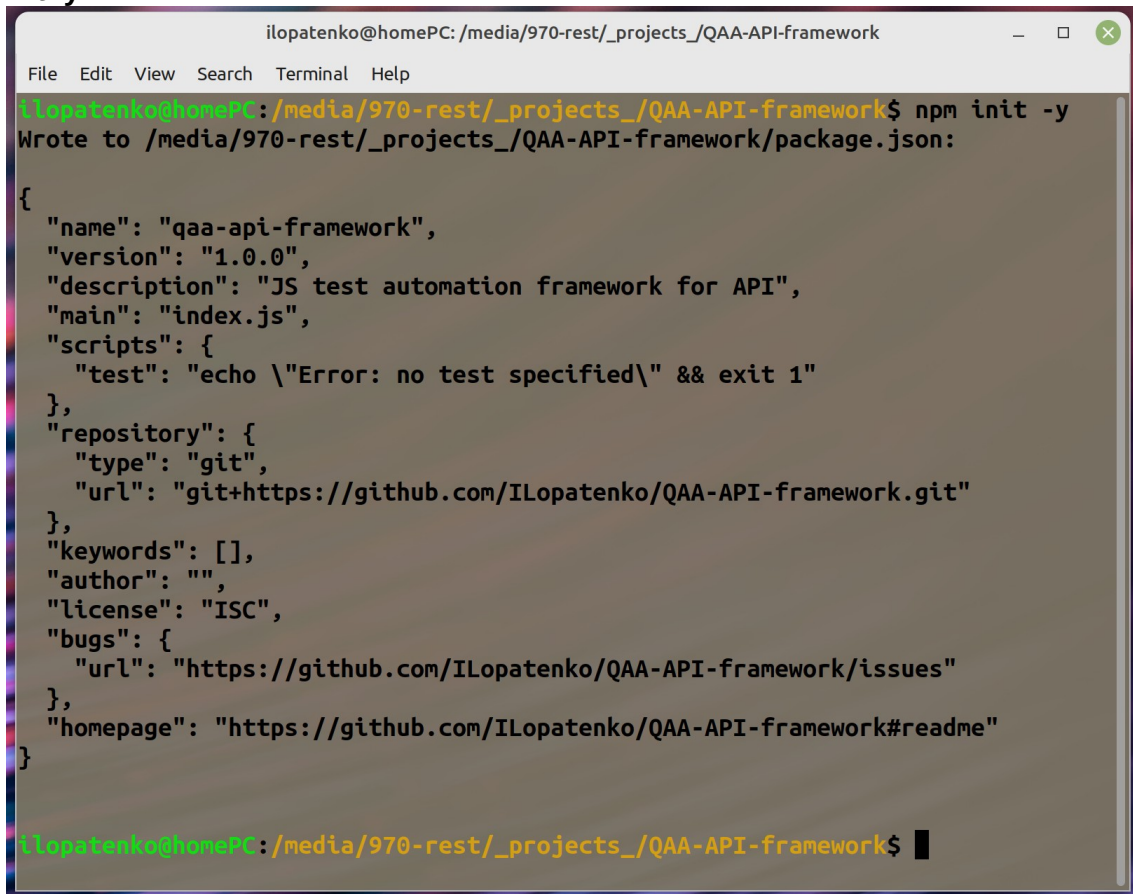


CD into your project's folder



Initialize a new NPM project

CLI=> **npm init -y**

A screenshot of a terminal window titled 'ilopatenko@homePC: /media/970-rest/_projects_/QAA-API-framework'. The terminal shows the command 'npm init -y' being executed, which results in a 'package.json' file being created. The content of the file is displayed as a JSON object. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The prompt is 'ilopatenko@homePC: /media/970-rest/_projects_/QAA-API-framework\$'.

```
ilopatenko@homePC: /media/970-rest/_projects_/QAA-API-framework$ npm init -y
Wrote to /media/970-rest/_projects_/QAA-API-framework/package.json:

{
  "name": "qaa-api-framework",
  "version": "1.0.0",
  "description": "JS test automation framework for API",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/ILopatenko/QAA-API-framework.git"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/ILopatenko/QAA-API-framework/issues"
  },
  "homepage": "https://github.com/ILopatenko/QAA-API-framework#readme"
}

ilopatenko@homePC: /media/970-rest/_projects_/QAA-API-framework$
```

6 Syntax for common test runners / Work with a new branch. Install BABEL.

Pull request. First tests

<https://dev.to/bormando/babel-setup-for-rest-api-tests-1dhf>

Create a new branch babel-setup (and switch to it)

CLI=> **git checkout -B babel-setup**

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$ git checkout -B babel-setup
Switched to a new branch 'babel-setup'
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$
```

Install Babel

CLI=> **npm i -D @babel/cli @babel/core @babel/plugin-transform-runtime @babel/preset-env @babel/register**

Create a new file **.babelrc** in the root directory of your project

CLI=> **touch .babelrc**

And add this text

```
{
  "presets": ["@babel/preset-env"],
  "plugins": [
    ["@babel/transform-runtime"]
  ]
}
```

CLI=> **nano .babelrc**

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

GNU nano 4.8
{
  "presets": ["@babel/preset-env"],
  "plugins": [
    ["@babel/transform-runtime"]
  ]
}
```

Now we can add all the changes to a STAGE. Create a new commit and push it to the repository.

CLI=> **git add .**

CLI=> **git commit -m 'Added Babel'**

CLI=> **git push --set-upstream origin babel-setup**

```
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$ git push --set-upstream origin babel-setup
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 44.15 KiB | 5.52 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'babel-setup' on GitHub by visiting:
remote:   https://github.com/ILopatenko/QAA-API-framework/pull/new/babel-setup
remote:
To https://github.com/ILopatenko/QAA-API-framework.git
 * [new branch]      babel-setup -> babel-setup
Branch 'babel-setup' set up to track remote branch 'babel-setup' from 'origin'.
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$
```


Go to your gitHub repository

The screenshot shows the GitHub repository page for 'babel-setup'. At the top, a notification bar indicates 'babel-setup had recent pushes less than a minute ago' with a green 'Compare & pull request' button. Below this, the repository navigation bar shows 'main' as the selected branch, '1 branch', and '0 tags'. There are buttons for 'Go to file', 'Add file', and 'Code'. The commit history table shows a recent commit by 'ILOpatenko' titled 'The new NPM project was initialized' (commit f3809d3, 30 minutes ago, 2 commits). The file list includes '.gitignore', 'LICENSE', 'README.md', 'package-lock.json', and 'package.json'. The 'README.md' file is expanded, showing the title 'QAA-API-framework' and the description 'JS test automation framework for API'.

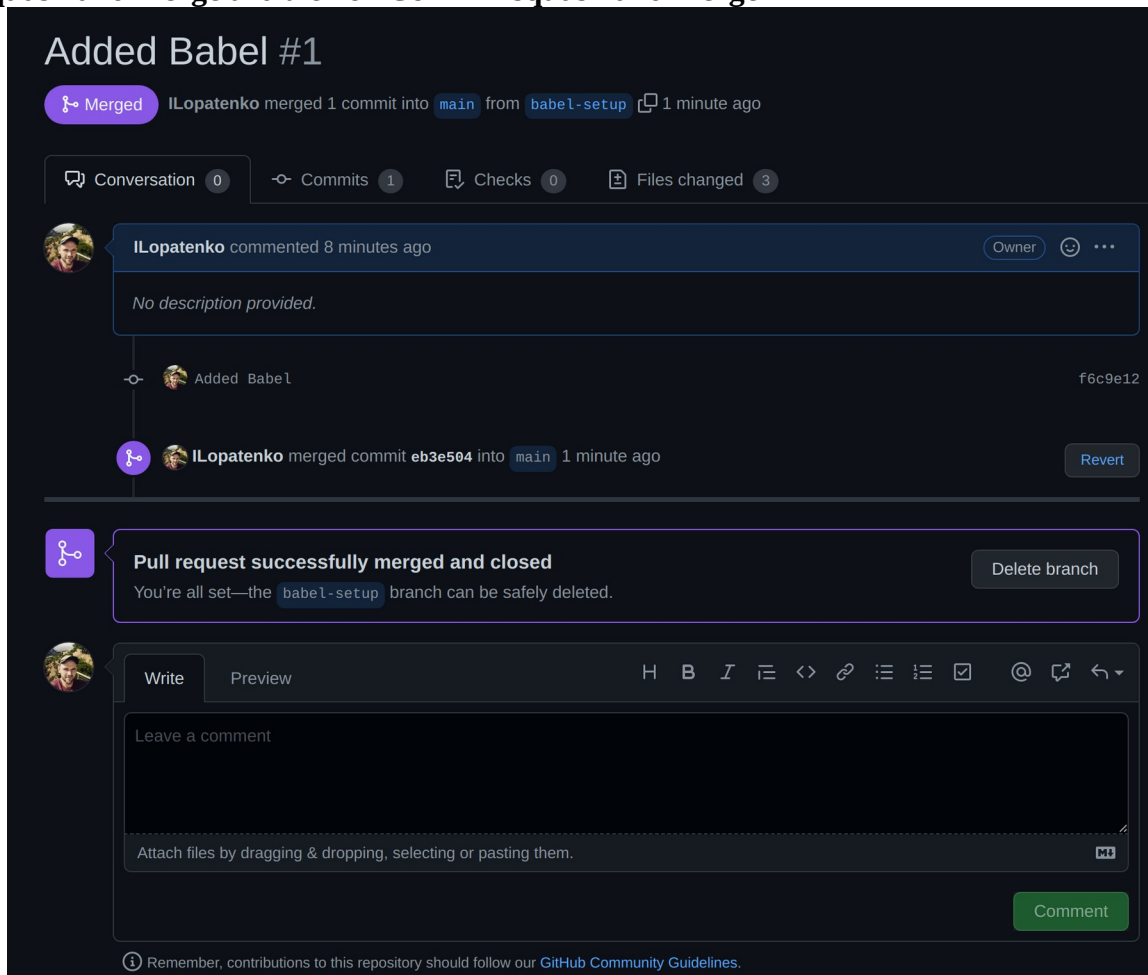
Click on **Compare & pull request** button

Check all the changes and if everything is good click on **Create pull request** button

The screenshot shows the 'Open a pull request' page. The title is 'Open a pull request' with a subtitle: 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).' The comparison bar shows 'base: main' and 'compare: babel-setup', with a green checkmark and the text 'Able to merge. These branches can be automatically merged.' The pull request title is 'Added Babel'. Below the title are 'Write' and 'Preview' tabs. The 'Write' tab is active, showing a large text area for the pull request description with a placeholder 'Leave a comment'. Above the text area is a rich text editor toolbar. Below the text area is a section for attaching files. At the bottom right is a green 'Create pull request' button. A footer note states: 'Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).'

Change default merge options to **Squash and merge**

Click on **Squash and merge** and then on **Confirm squash and merge**



Now we can delete branch **babel-setup** by click on **Delete branch** button

Change a branch to **main** in your IDE

```
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$
```

Pull all the changes to our local main branch from gitHub

CLI=> **git pull**

```
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$ git pull
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 1), reused 4 (delta 1), pack-reused 0
Unpacking objects: 100% (5/5), 44.52 KiB | 337.00 KiB/s, done.
From https://github.com/ILOpatenko/QAA-API-framework
   f3809d3..eb3e504  main       -> origin/main
Updating f3809d3..eb3e504
Fast-forward
 .babelrc           |    6 +
 package-lock.json  | 4866 ++++++
 package.json       |    9 +-
 3 files changed, 4878 insertions(+), 3 deletions(-)
 create mode 100644 .babelrc
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$
```

Create a new branch **first-tests**

CLI=> **git checkout -B first-tests**

Install Mocha and Chai packages

CLI=> **npm i -D mocha chai**

```
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$ npm i -D mocha chai
added 62 packages, and audited 278 packages in 2s

31 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$
```

Create a new directory **specs**

CLI=> **mkdir specs**

Create a new file **example.spec.js** in directory **specs**

CLI=> **touch specs/example.spec.js**

Open **specs/example.spec.js** with IDE import **Chai** and **Mocha** and start writing first tests

```
JS example.spec.js U X
specs > JS example.spec.js > ...
1  import { expect } from 'chai';
2
3  describe('Math functions', () => {
4    it('A + B = C', () => {
5      const a = 4;
6      const b = 7;
7      const c = a + b;
8      expect(c).to.eq(11);
9    });
10 });
11
```

Open **package.json** file and make some changes in test script:

```
"scripts": {
  "test": "npx mocha --config .mocharc.js"
},
```

Create a new file **.mocharc.js**

CLI=> **touch .mocharc.js**

And add next text

```
module.exports = {
  require: '@babel/register',
  spec: 'specs/**/*.spec.js',
};
```

Run a first test

CLI=> **npm run test**

```
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$ npm run test
> qaa-api-framework@1.0.0 test
> npx mocha --config .mocharc.js

Math functions
  ✓ A + B = C

1 passing (3ms)
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$
```

Create a new commit, push it to gitHub, Create a pull request, merge pull request and pull all the changes to a local computer

7 Syntax for HTTP client / First API tests

8 Environment variables

9 Test runner (mocha) hooks

10 Wrappers for API tests

11 API tests practice

12 Setting up a mock server

13 CI/CD theory

14 Integrating API tests with GitHub Actions