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

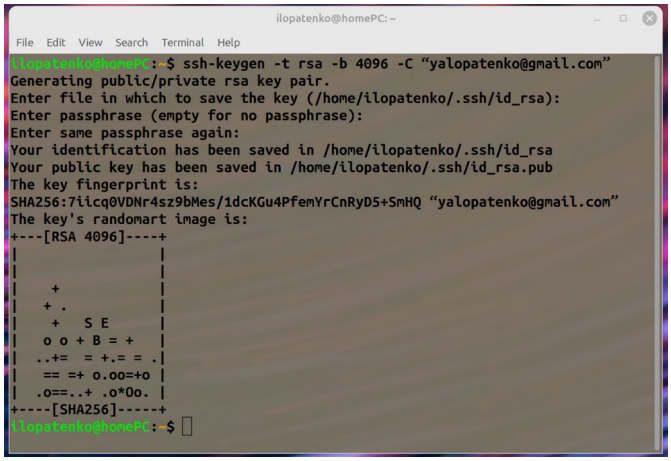
4 SSH keys

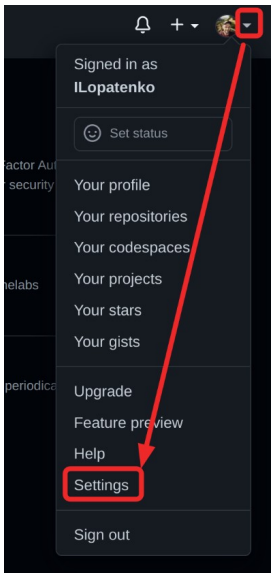
Open an terminal

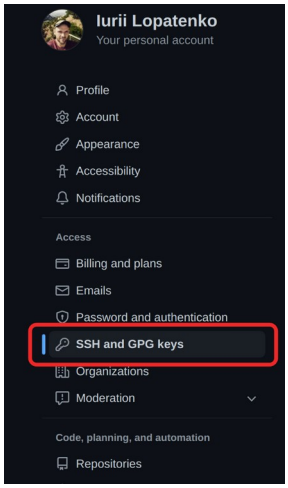
CD to a user home directory

Run next command:

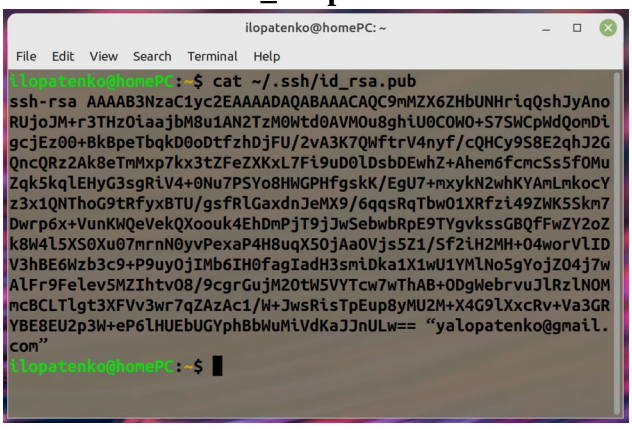
CLI=> **ssh-keygen -t rsa -b 4096 -C “ENTER_HERE_YOUR_EMAIL_OR_WHATEVER”**





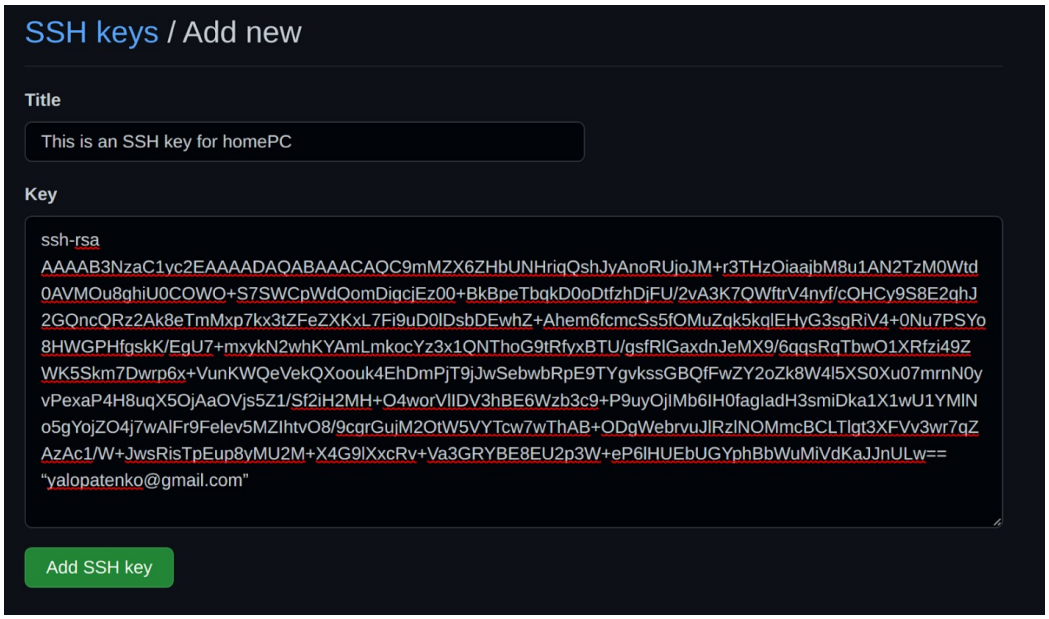


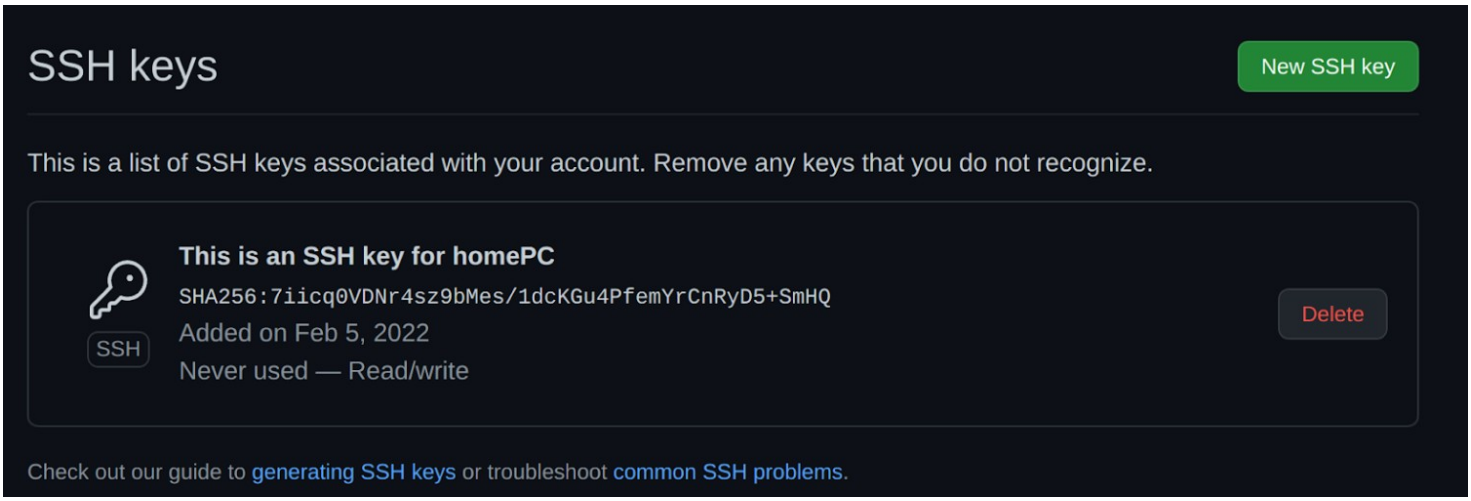
Go to gitHub. Go to settings https://github.com/settings/profile
Go to SSH and GPG keys https://github.com/settings/keys
Copy SSH key to buffer:
CLI=> **cat ~/.ssh/id_rsa.pub**



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

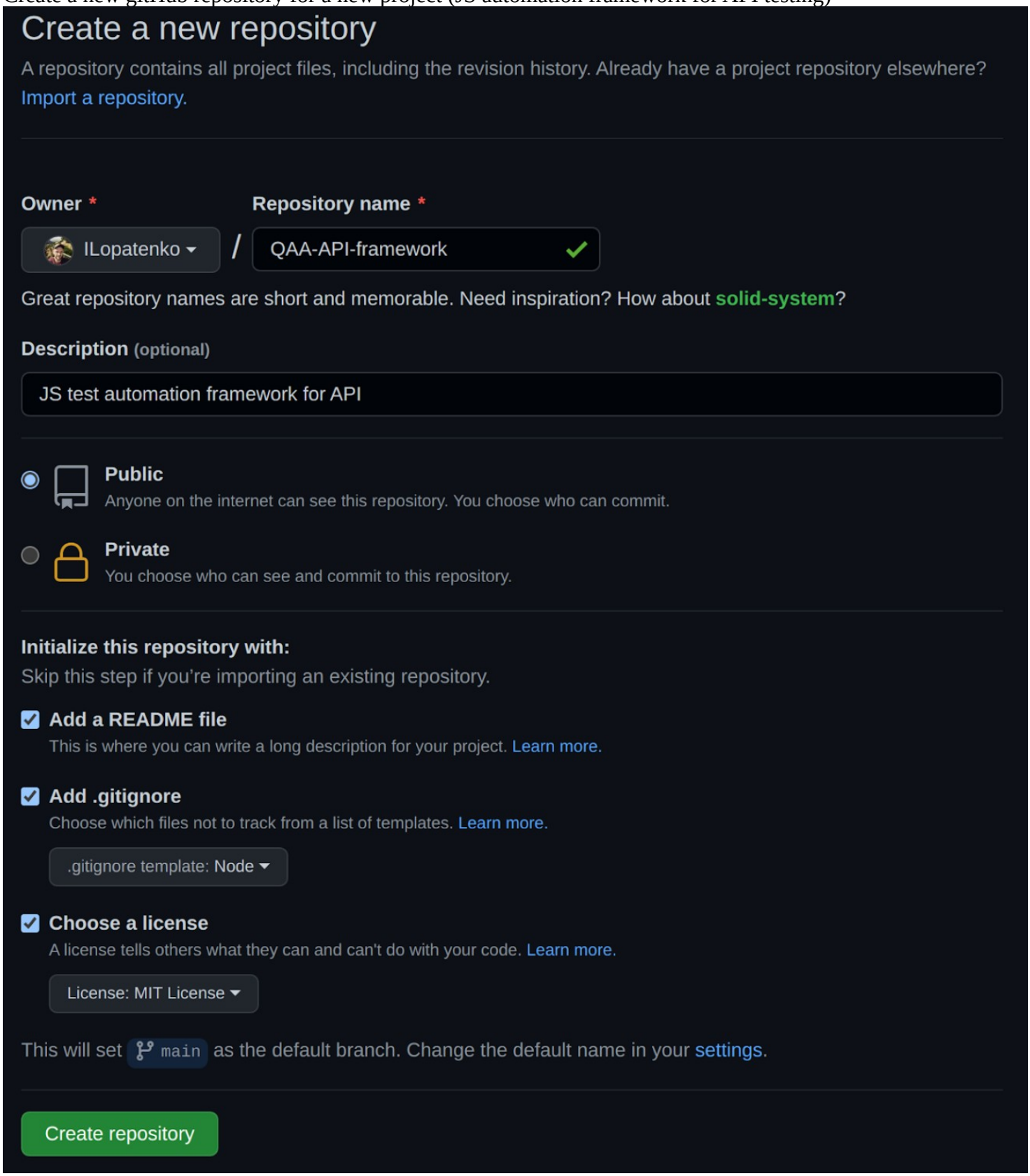
and confirm (re enter your gitHub password).

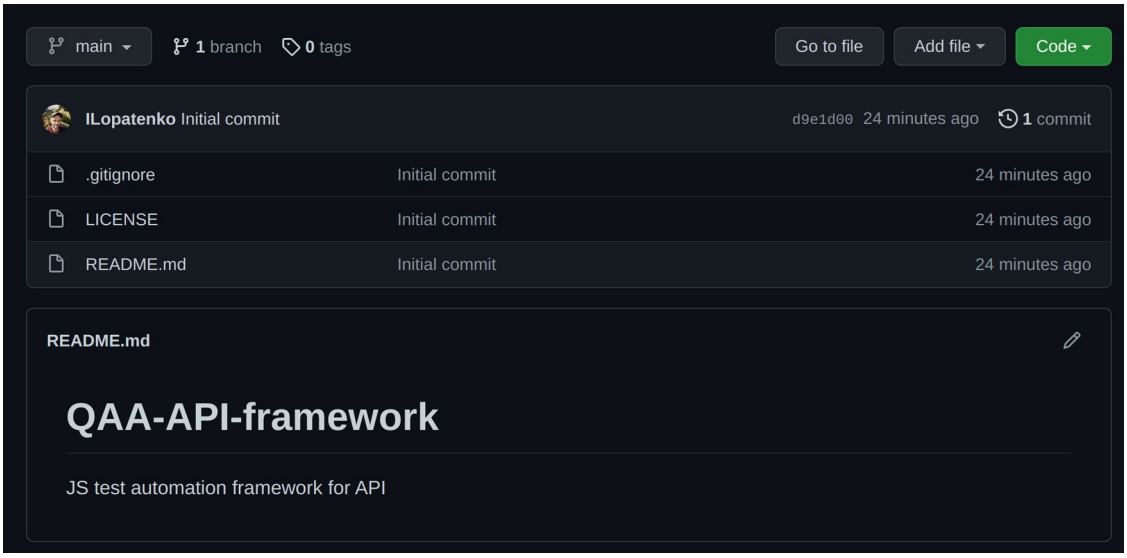


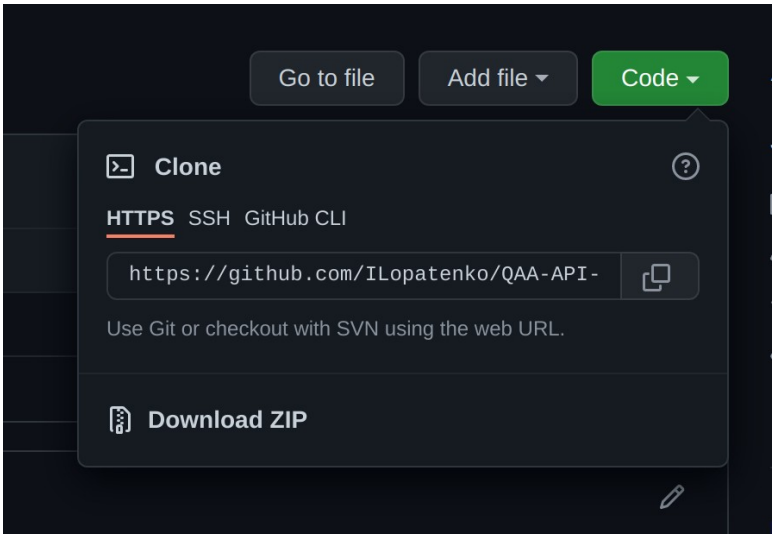


5 Test project setup

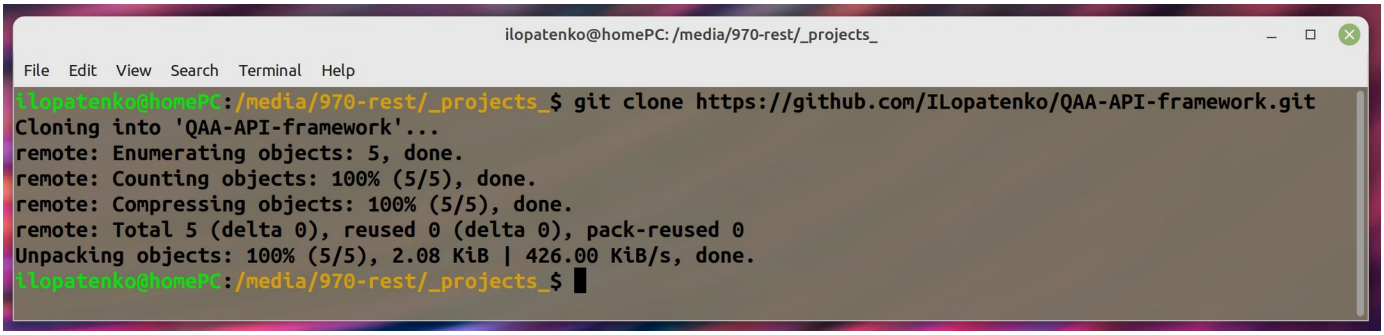
Create a new gitHub repository for a new project (JS automation framework for API testing)



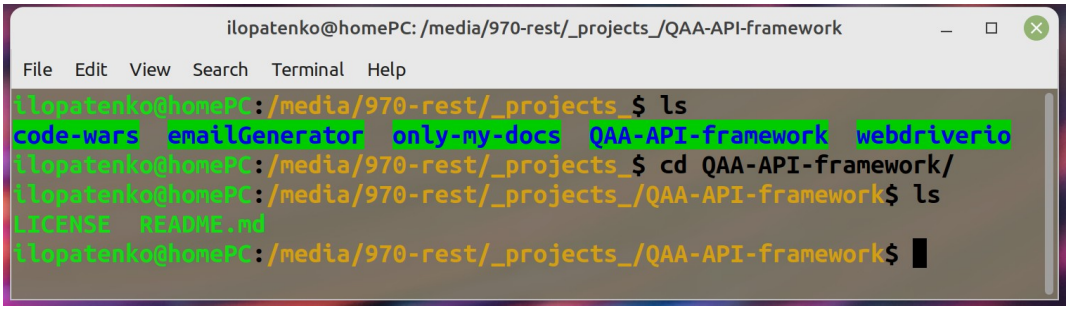




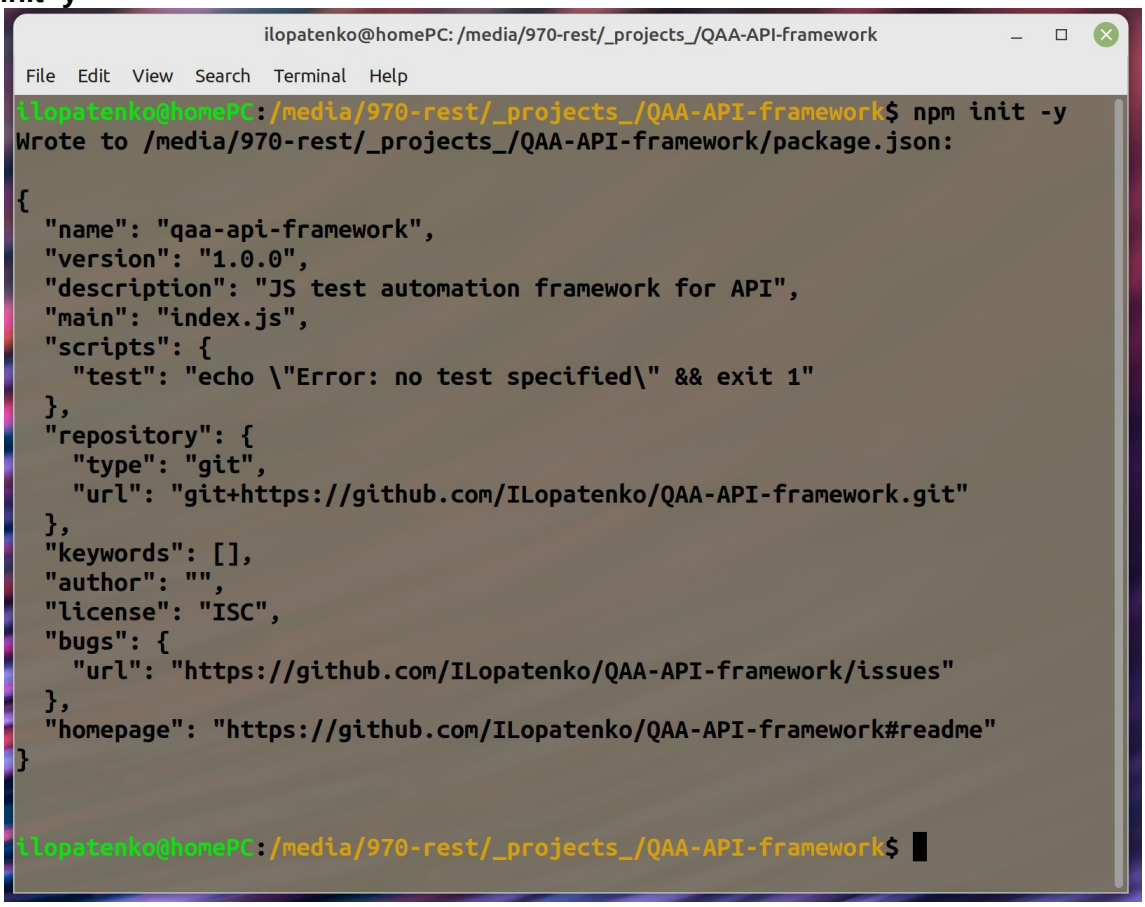
Copy your new repository's URL
Open terminal and **CD** to a directory where you want to keep your framework and clone it
CLI=> **git clone** **https://github.com/ILopatenko/QAA-API-framework.git**



CD into your project's folder



Initialize a new NPM project
CLI=> **npm init -y**



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**

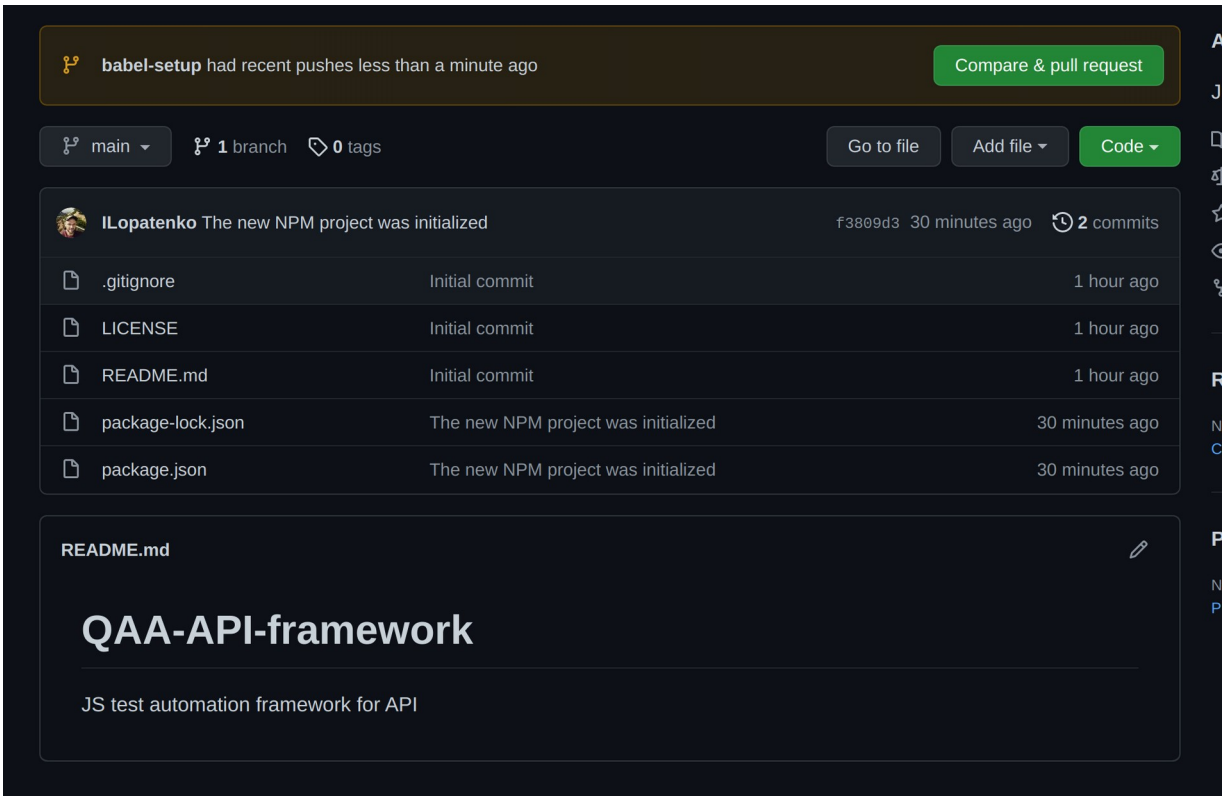
```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

  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**

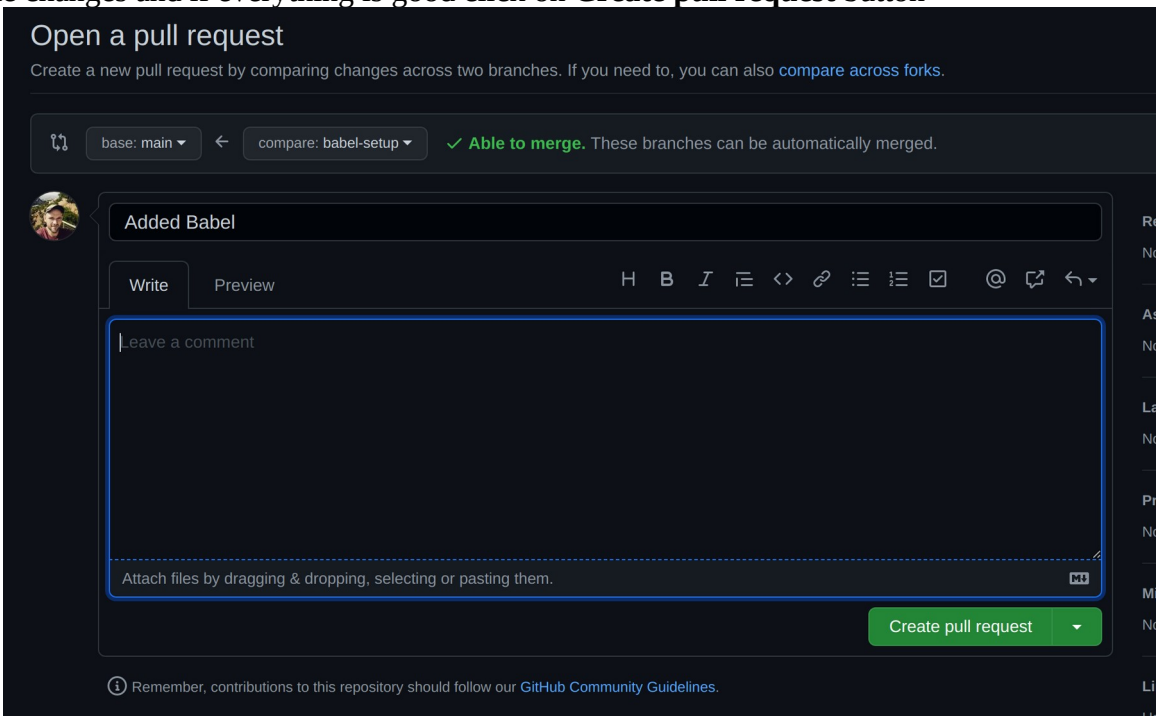
```
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$ git push --set-upstream origin babel-setup
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 44.15 KiB | 5.52 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

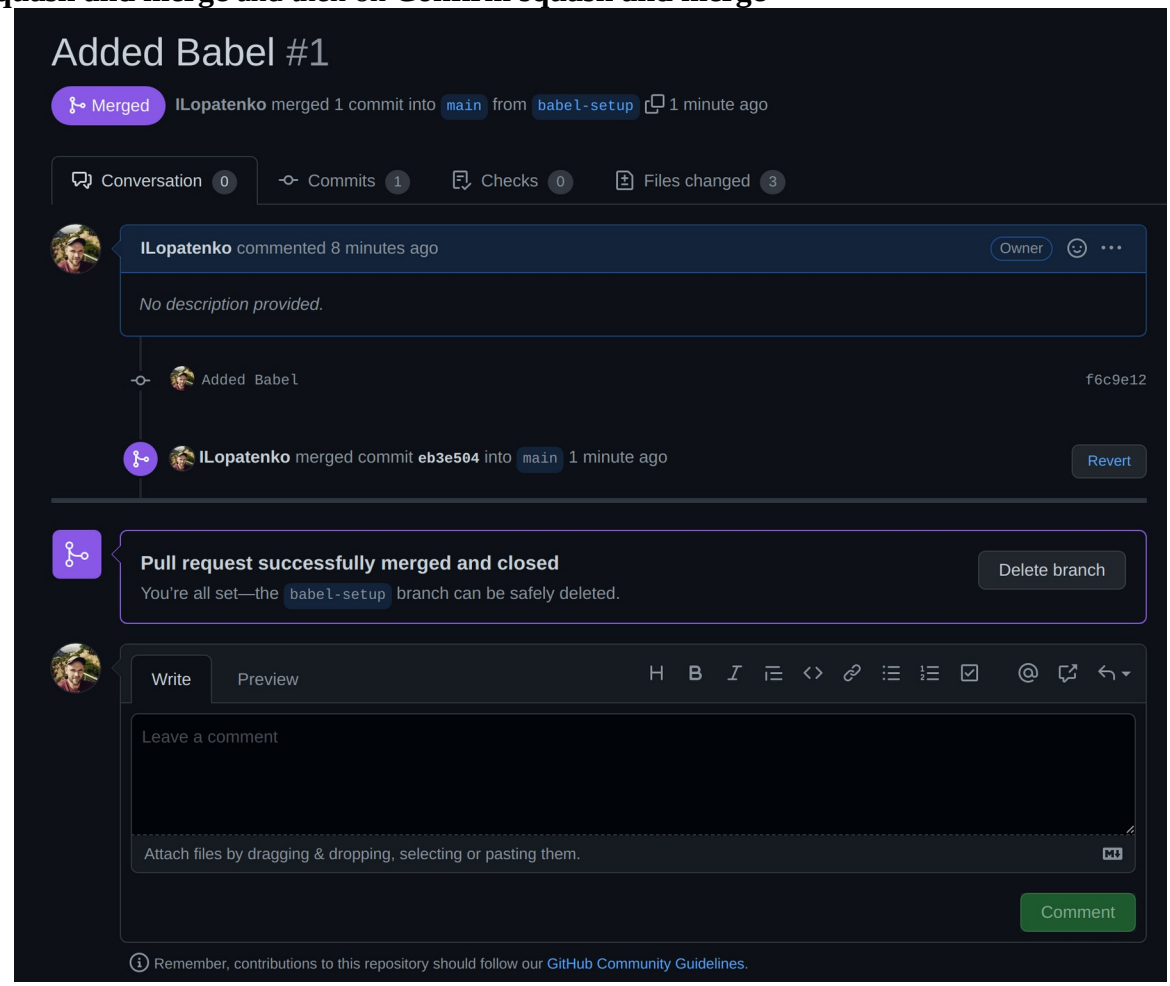


Click on **Compare & pull request** button

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



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

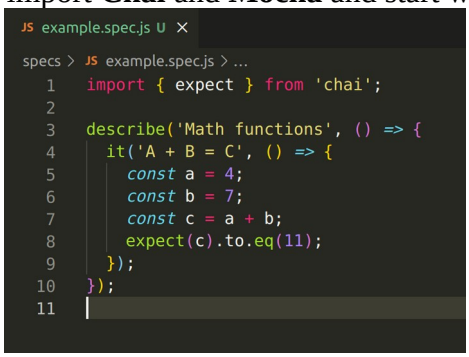
```
ilopatenko@homePC:/media/970-rest/_projects_/QAA-API-framework$ npm i -D mocha chai

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



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