Ricardo Wu

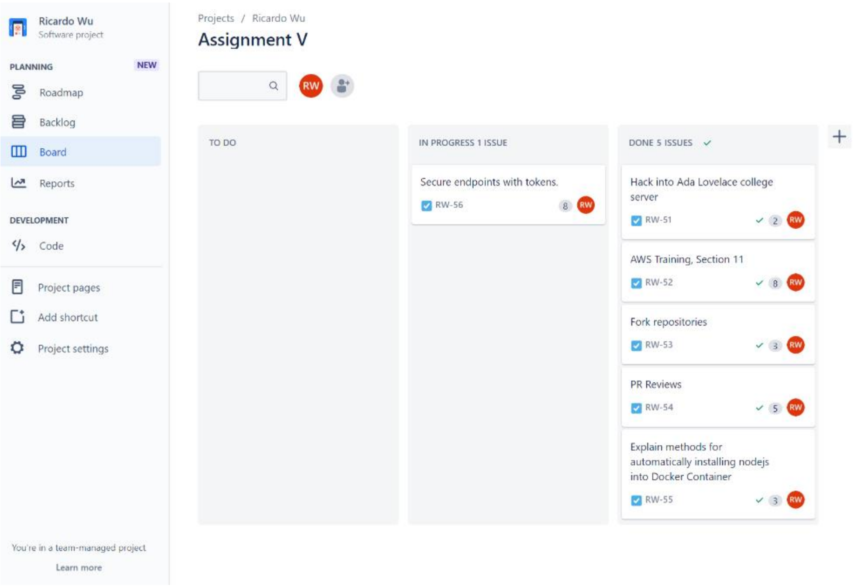
2/6/2022

CS 561

Prof. Braynen

Week 5 assignment

Scrum Board:



Assignment V

TO DO

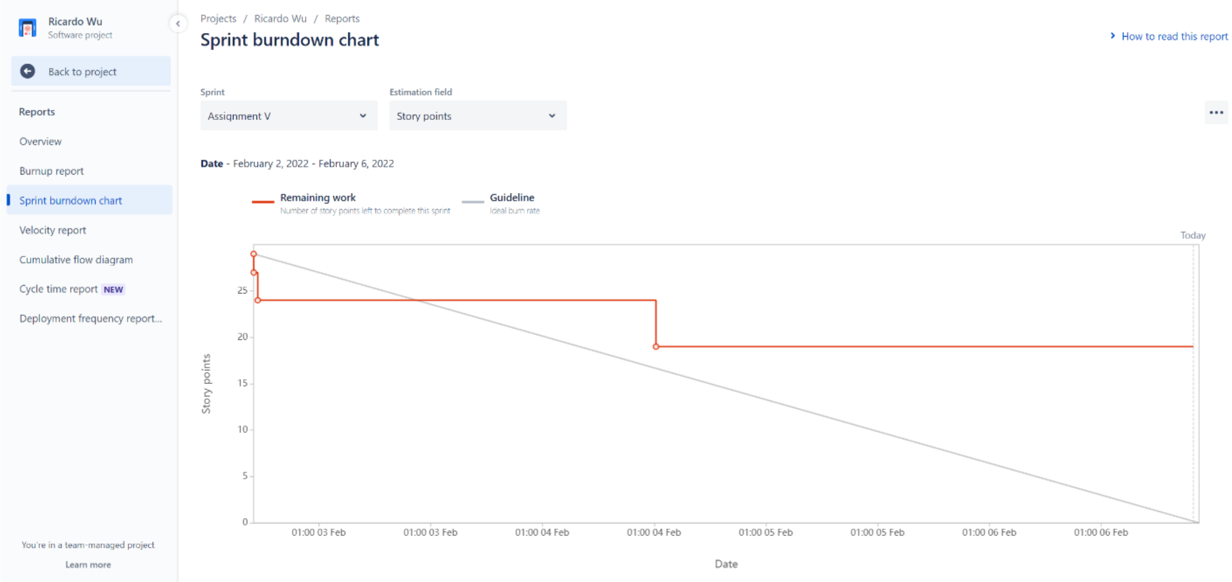
IN PROGRESS 1 ISSUE

- Secure endpoints with tokens. ☑ RW-56 8 RW

DONE 5 ISSUES ✓

- Hack into Ada Lovelace college server ☑ RW-51 ✓ 2 RW
- AWS Training, Section 11 ☑ RW-52 ✓ 8 RW
- Fork repositories ☑ RW-53 ✓ 3 RW
- PR Reviews ☑ RW-54 ✓ 5 RW
- Explain methods for automatically installing nodejs into Docker Container ☑ RW-55 ✓ 3 RW

Burndown chart:



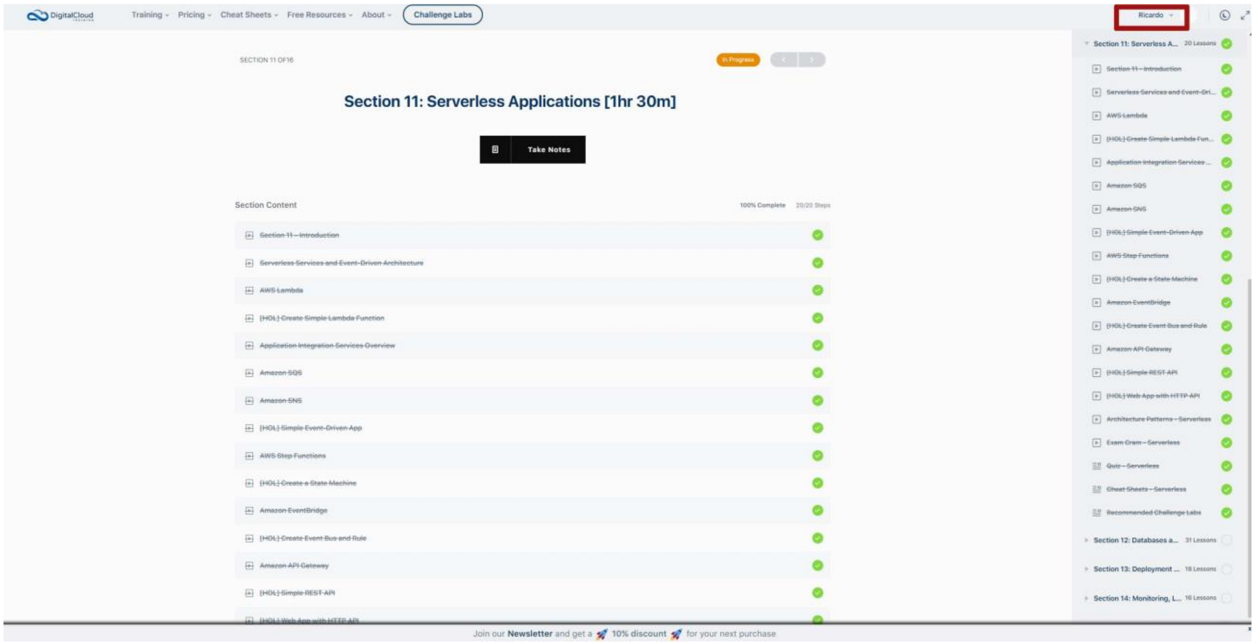
Sprint burndown chart

Sprint: Assignment V

Estimation field: Story points

Date - February 2, 2022 - February 6, 2022

AWS Training:



SECTION 11 OF16

Section 11: Serverless Applications [1hr 30m]

Section Content — 100% Complete — 20/20 Steps

- ~~Section 11 – Introduction~~ ✓
- ~~Serverless Services and Event-Driven Architecture~~ ✓
- ~~AWS Lambda~~ ✓
- ~~[HOL] Create Simple Lambda Function~~ ✓
- ~~Application Integration Services Overview~~ ✓
- ~~Amazon SQS~~ ✓
- ~~Amazon SNS~~ ✓
- ~~[HOL] Simple Event-Driven App~~ ✓
- ~~AWS Step Functions~~ ✓
- ~~[HOL] Create a State Machine~~ ✓
- ~~Amazon EventBridge~~ ✓
- ~~[HOL] Create Event Bus and Rule~~ ✓
- ~~Amazon API Gateway~~ ✓
- ~~[HOL] Simple REST API~~ ✓
- ~~[HOL] Web App with HTTP API~~ ✓

Section 11: Serverless A... 20 Lessons ✓

- ~~Section 11 – Introduction~~ ✓
- ~~Serverless Services and Event-Dri...~~ ✓
- ~~AWS Lambda~~ ✓
- ~~[HOL] Create Simple Lambda Fun...~~ ✓
- ~~Application Integration Services ...~~ ✓
- ~~Amazon SQS~~ ✓
- ~~Amazon SNS~~ ✓
- ~~[HOL] Simple Event-Driven App~~ ✓
- ~~AWS Step Functions~~ ✓
- ~~[HOL] Create a State Machine~~ ✓
- ~~Amazon EventBridge~~ ✓
- ~~[HOL] Create Event Bus and Rule~~ ✓
- ~~Amazon API Gateway~~ ✓
- ~~[HOL] Simple REST API~~ ✓
- ~~[HOL] Web App with HTTP API~~ ✓
- ~~Architecture Patterns – Serverless~~ ✓
- ~~Exam Cram – Serverless~~ ✓
- ~~Quiz – Serverless~~ ✓
- ~~Cheat Sheets – Serverless~~ ✓
- ~~Recommended Challenge Labs~~ ✓

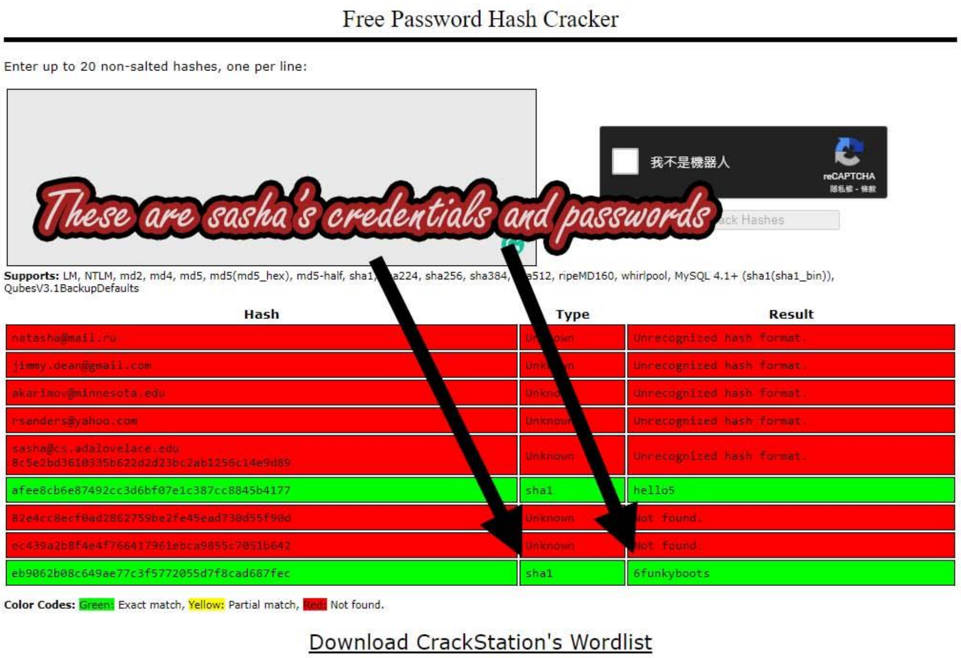
Section 12: Databases a... 31 Lessons ○

Section 13: Deployment ... 18 Lessons ○

Section 14: Monitoring, L... 16 Lessons ○

Join our **Newsletter** and get a 🚀 **10% discount** 🚀 for your next purchase

Ada Lovelace College server:



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

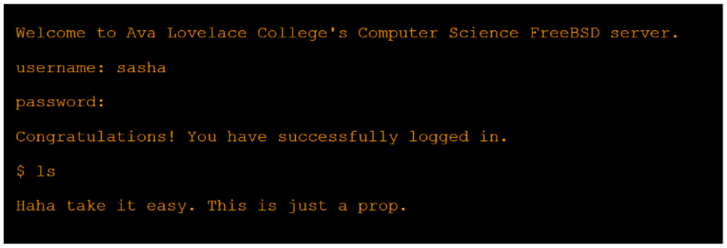
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
natasha@mail.ru	Unknown	Unrecognized hash format.
jimmy.dean@gmail.com	Unknown	Unrecognized hash format.
akarimov@minnesota.edu	Unknown	Unrecognized hash format.
rsanders@yahoo.com	Unknown	Unrecognized hash format.
sasha@cs.adalovelace.edu 8c5e2bd3610335b622d2d23bc2ab1256c14e9d89	Unknown	Unrecognized hash format.
afee8cb6e87492cc3d6bf07e1c387cc8845b4177	sha1	hello5
82e4cc8ecf0ad2862759be2fe45ead730d55f90d	Unknown	Not found.
ec439a2b8f4e4f766417961ebca9855c7051b642	Unknown	Not found.
eb9062b08c649ae77c3f5772055d7f8cad687fec	sha1	6funkyboots

Color Codes: 🟩 Green: Exact match, 🟨 Yellow: Partial match, 🟥 Red: Not found.

Download CrackStation's Wordlist

The orange-and-black terminal window below will probably not show up on a smartphone. (Have not tried on a tablet.) So you might have to use a laptop (e.g. your macbook) or desktop for this one.



```
Welcome to Ava Lovelace College's Computer Science FreeBSD server.

username: sasha

password:

Congratulations! You have successfully logged in.

$ ls

Haha take it easy. This is just a prop.
```

5 links of merged pull requests (PRs):

a. Warm-up PR: https://github.com/WisleyDoma/weather-backend-RW/pull/2

b. v1/auth PR, with token & expr. date: https://github.com/WisleyDoma/weather-backend-RW/pull/3

c. v1/weather & v1/hello PR, with token & expr. date: https://github.com/WisleyDoma/weather-backend-RW/pull/4

d. Swift client PR: https://github.com/WisleyDoma/cs561-swift-client-RW/pull/1

e. Rust client PR: https://github.com/WisleyDoma/cs561-rust-client-RW/pull/1

Difference between Dockerfile & Docker-compose (about auto installing node.js):

Dockerfile is a .txt file for users to initialize docker. It could auto-install node.js by typing **RUN apt-get install nodejs**. However, it only builds one docker at a time and dockerfile requires users to be proficient in commands since dockerfile doesn't have image; Docker-compose is a .yml file that could launch multiple dockers with images. It's easier for novices to get into Docker. When it needs to auto-install node.js, users could type **command: bash -c "apt-get update && apt-get install nodejs"** beneath the environment segment.

I prefer to use docker-compose because it helps us to practice scenarios where server and clients set in different places. Another perk is users could download docker image from docker hub. It's more convenient for users than memorizing commands for different clients' environment.

References:

Difference between Dockerfile & Docker-compose:
https://stackoverflow.com/questions/29480099/whats-the-difference-between-docker-compose-vs-dockerfile

What is Dockerfile | How to create and build a dockerfile | Dockerfile basic commands:
https://youtu.be/LQjaJINkQXY

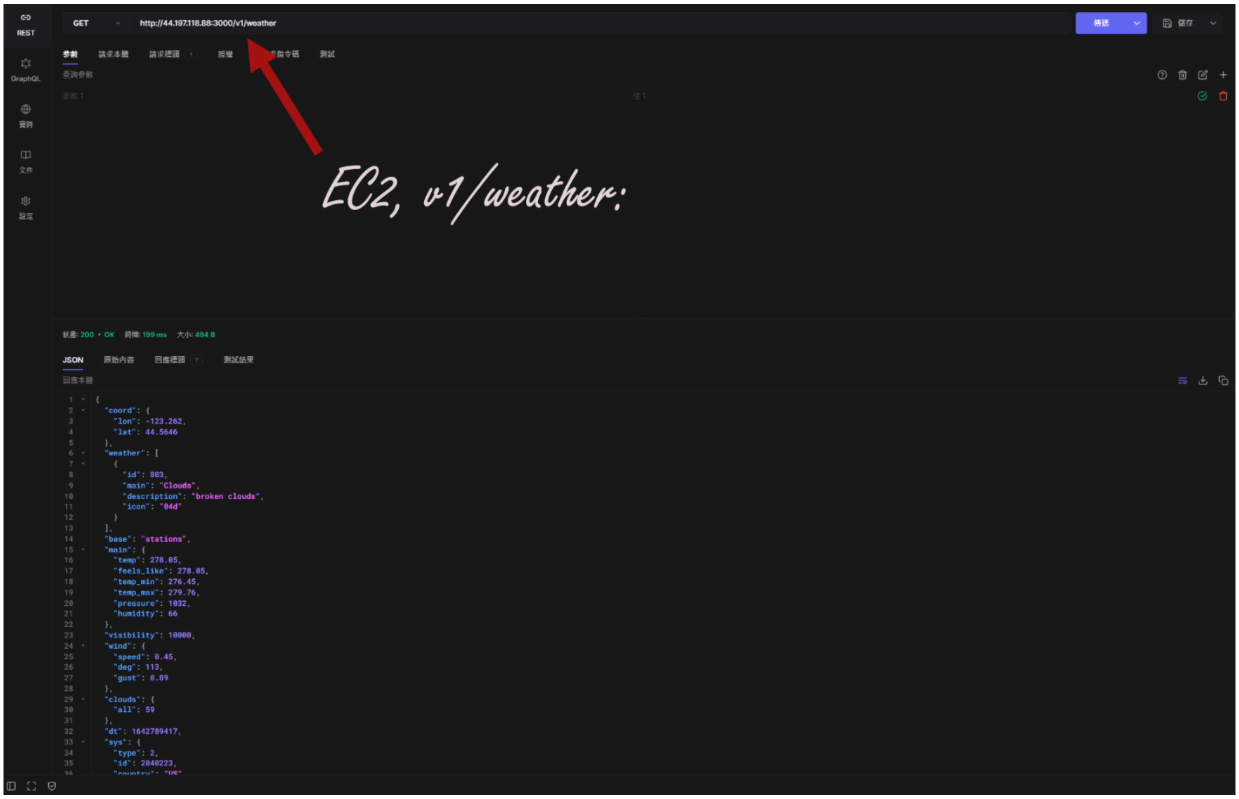
What is the difference between dockerfile and docker-compose.yml files?
https://youtu.be/BlX26GThQPQ

2 links of updated swagger doc:

a. GitHub: https://github.com/WisleyDoma/weather-backend-RW/pull/5
b. EC2 (Elastic Computing Cloud):
   ec2-44-197-118-88.compute-1.amazonaws.com
   (IP address: 44.197.118.88) – closed to prevent additional charges.

Examples:

EC2, v1/weather:

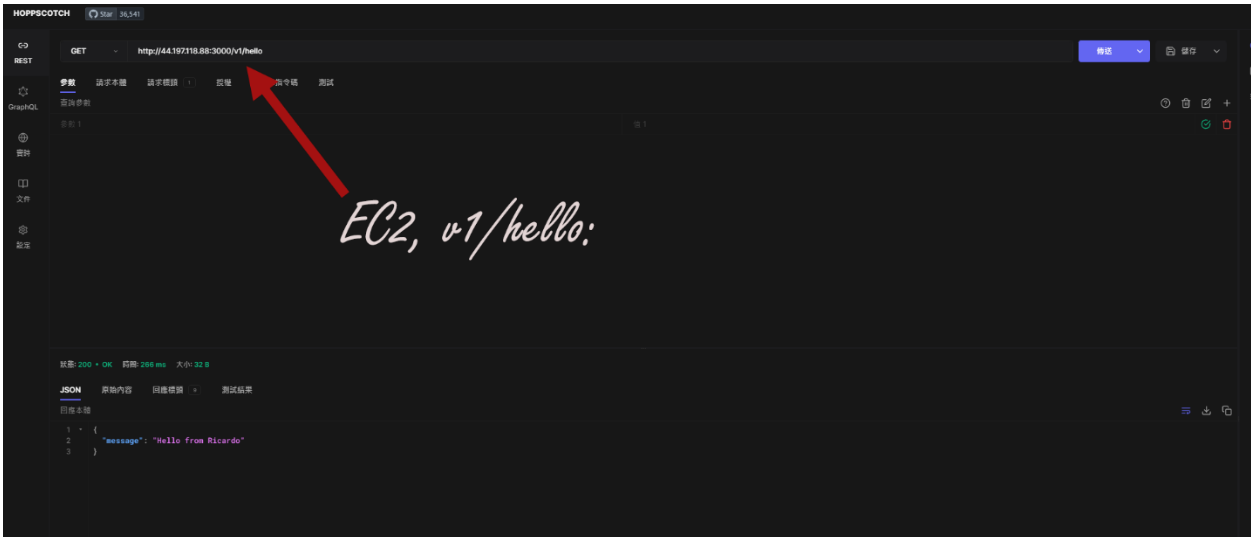


http://44.197.118.88:3000/v1/weather

狀態: 200 • OK 時間: 199 ms 大小: 494 B

```
{
  "coord": {
    "lon": -123.262,
    "lat": 44.5646
  },
  "weather": [
    {
      "id": 803,
      "main": "Clouds",
      "description": "broken clouds",
      "icon": "04d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 278.05,
    "feels_like": 278.05,
    "temp_min": 276.45,
    "temp_max": 279.76,
    "pressure": 1032,
    "humidity": 66
  },
  "visibility": 10000,
  "wind": {
    "speed": 0.45,
    "deg": 113,
    "gust": 0.89
  },
  "clouds": {
    "all": 59
  },
  "dt": 1642789417,
  "sys": {
    "type": 2,
    "id": 2040223,
    "country": "US"
```

EC2, v1/hello:



http://44.197.118.88:3000/v1/hello

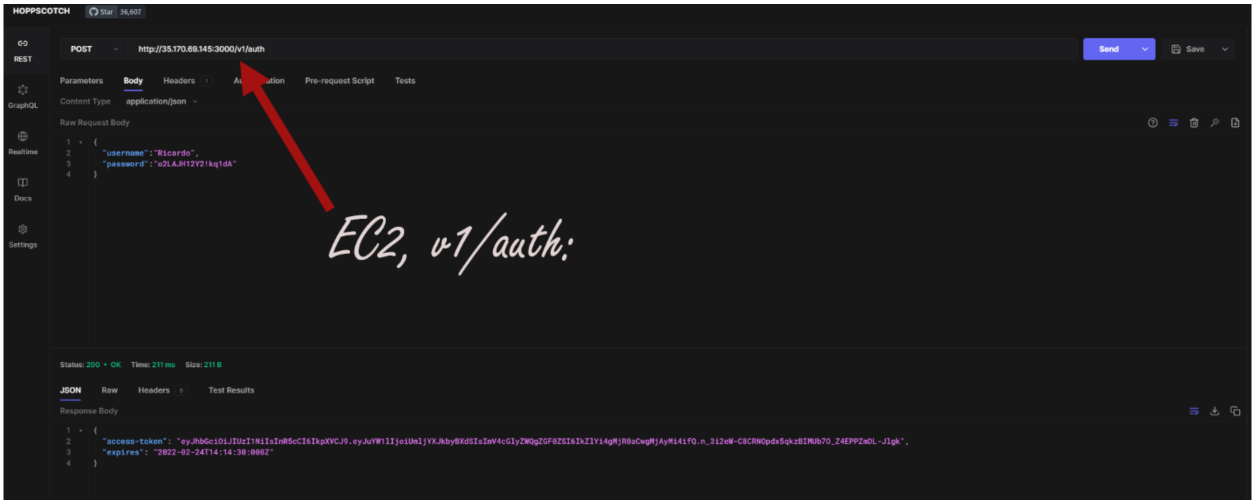
狀態: 200 • OK 時間: 266 ms 大小: 32 B

```
{
  "message": "Hello from Ricardo"
}
```

ec2-35-170-69-145.compute-1.amazonaws.com

(IP address: 35.170.69.145) – Opened to add v1/auth screenshot.

EC2, v1/auth:



POST http://35.170.69.145:3000/v1/auth

Content Type application/json

Raw Request Body

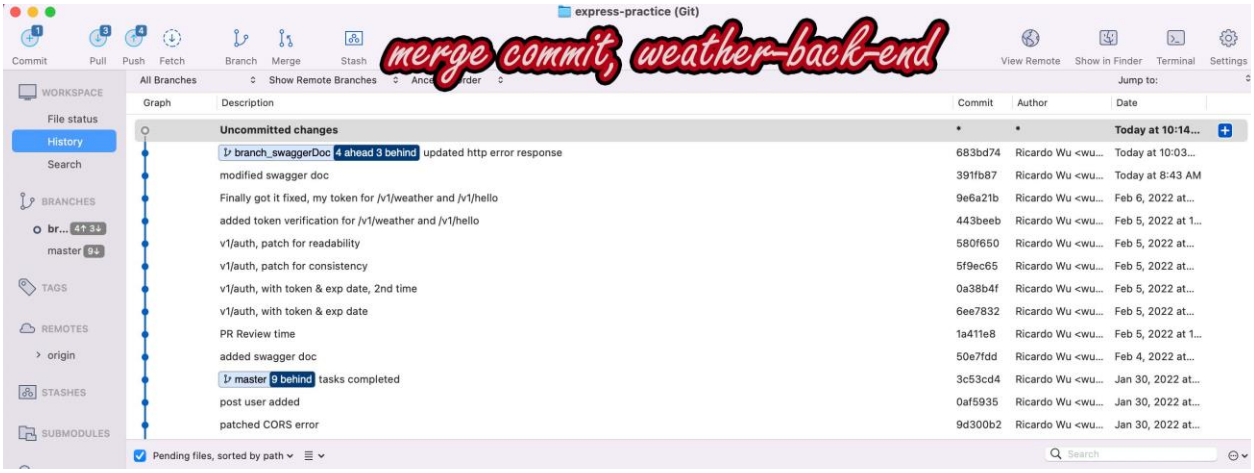
```
{
  "username":"Ricardo",
  "password":"o2LAJH12Y2!kq1dA"
}
```

Status: 200 • OK Time: 211 ms Size: 211 B

Response Body

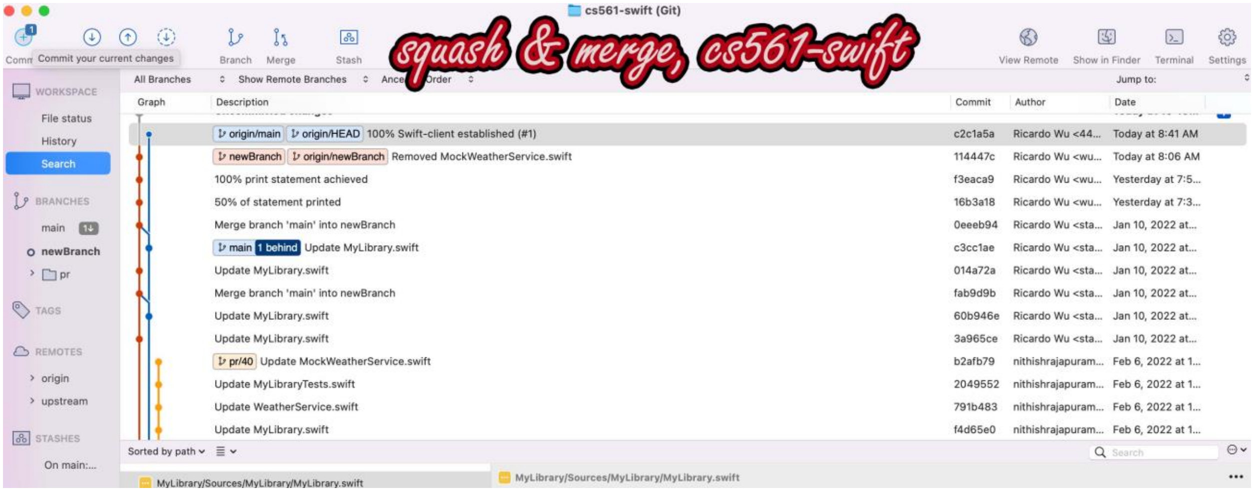
```
{
  "access-token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiUmljYXJkbyBXdSIsImV4cGlyZWQgZGF0ZSI6IkZlYi4gMjR0aCwgMjAyMi4ifQ.n_3i2eW-C8CRNOpdx5qkzBIMUb7O_Z4EPPZmDL-Jlgk",
  "expires": "2022-02-24T14:14:30:000Z"
}
```

“Create a merge to commit”, “squash and merge”, and “rebase and merge” buttons:



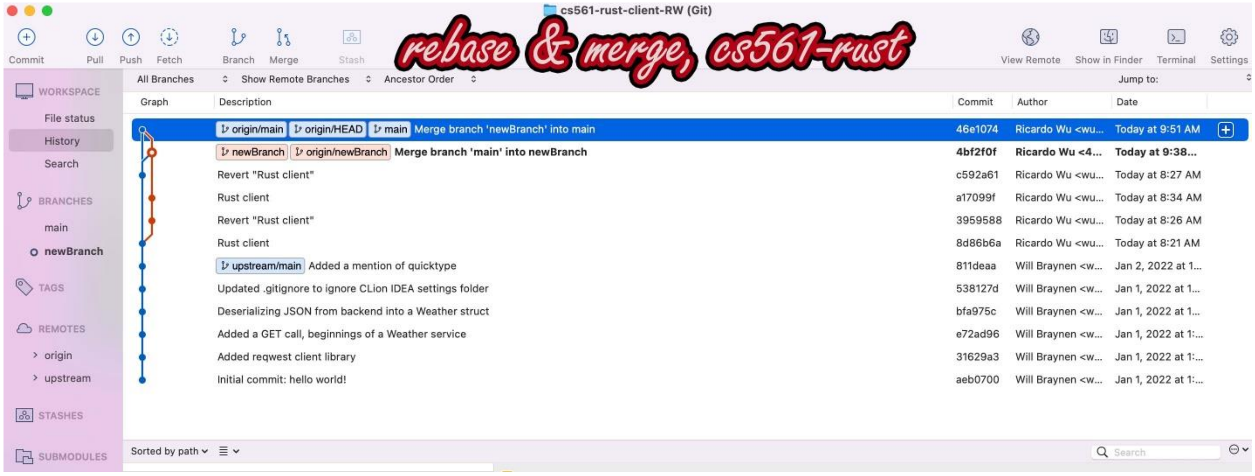
Description	Commit	Author	Date
Uncommitted changes	•	•	**Today at 10:14...**
branch_swaggerDoc 4 ahead 3 behind updated http error response	683bd74	Ricardo Wu <wu...	Today at 10:03...
modified swagger doc	391fb87	Ricardo Wu <wu...	Today at 8:43 AM
Finally got it fixed, my token for /v1/weather and /v1/hello	9e6a21b	Ricardo Wu <wu...	Feb 6, 2022 at...
added token verification for /v1/weather and /v1/hello	443beeb	Ricardo Wu <wu...	Feb 5, 2022 at 1...
v1/auth, patch for readability	580f650	Ricardo Wu <wu...	Feb 5, 2022 at...
v1/auth, patch for consistency	5f9ec65	Ricardo Wu <wu...	Feb 5, 2022 at...
v1/auth, with token & exp date, 2nd time	0a38b4f	Ricardo Wu <wu...	Feb 5, 2022 at...
v1/auth, with token & exp date	6ee7832	Ricardo Wu <wu...	Feb 5, 2022 at...
PR Review time	1a411e8	Ricardo Wu <wu...	Feb 5, 2022 at 1...
added swagger doc	50e7fdd	Ricardo Wu <wu...	Feb 4, 2022 at...
master 9 behind tasks completed	3c53cd4	Ricardo Wu <wu...	Jan 30, 2022 at...
post user added	0af5935	Ricardo Wu <wu...	Jan 30, 2022 at...
patched CORS error	9d300b2	Ricardo Wu <wu...	Jan 30, 2022 at...

The merge commit is good for new git users and small projects. It’s easy for users to examine their every commits.



Description	Commit	Author	Date
origin/main origin/HEAD 100% Swift-client established (#1)	c2c1a5a	Ricardo Wu <44...	Today at 8:41 AM
newBranch origin/newBranch Removed MockWeatherService.swift	114447c	Ricardo Wu <wu...	Today at 8:06 AM
100% print statement achieved	f3eaca9	Ricardo Wu <wu...	Yesterday at 7:5...
50% of statement printed	16b3a18	Ricardo Wu <wu...	Yesterday at 7:3...
Merge branch 'main' into newBranch	0eeeb94	Ricardo Wu <sta...	Jan 10, 2022 at...
main 1 behind Update MyLibrary.swift	c3cc1ae	Ricardo Wu <sta...	Jan 10, 2022 at...
Update MyLibrary.swift	014a72a	Ricardo Wu <sta...	Jan 10, 2022 at...
Merge branch 'main' into newBranch	fab9d9b	Ricardo Wu <sta...	Jan 10, 2022 at...
Update MyLibrary.swift	60b946e	Ricardo Wu <sta...	Jan 10, 2022 at...
Update MyLibrary.swift	3a965ce	Ricardo Wu <sta...	Jan 10, 2022 at...
pr/40 Update MockWeatherService.swift	b2afb79	nithishrajapuram...	Feb 6, 2022 at 1...
Update MyLibraryTests.swift	2049552	nithishrajapuram...	Feb 6, 2022 at 1...
Update WeatherService.swift	791b483	nithishrajapuram...	Feb 6, 2022 at 1...
Update MyLibrary.swift	f4d65e0	nithishrajapuram...	Feb 6, 2022 at 1...

The squash and merge makes each merge concise. It is because a merge contains every single commit in there, creating a better readability of merges. It is good for projects that have multiples teams (i.e. different companies) to work on.



Graph	Description	Commit	Author	Date
	origin/main origin/HEAD main Merge branch 'newBranch' into main	46e1074	Ricardo Wu <wu...	Today at 9:51 AM
	newBranch origin/newBranch **Merge branch 'main' into newBranch**	**4bf2f0f**	**Ricardo Wu <4...**	**Today at 9:38...**
	Revert "Rust client"	c592a61	Ricardo Wu <wu...	Today at 8:27 AM
	Rust client	a17099f	Ricardo Wu <wu...	Today at 8:34 AM
	Revert "Rust client"	3959588	Ricardo Wu <wu...	Today at 8:26 AM
	Rust client	8d86b6a	Ricardo Wu <wu...	Today at 8:21 AM
	upstream/main Added a mention of quicktype	811deaa	Will Braynen <w...	Jan 2, 2022 at 1...
	Updated .gitignore to ignore CLion IDEA settings folder	538127d	Will Braynen <w...	Jan 1, 2022 at 1...
	Deserializing JSON from backend into a Weather struct	bfa975c	Will Braynen <w...	Jan 1, 2022 at 1...
	Added a GET call, beginnings of a Weather service	e72ad96	Will Braynen <w...	Jan 1, 2022 at 1:...
	Added reqwest client library	31629a3	Will Braynen <w...	Jan 1, 2022 at 1:...
	Initial commit: hello world!	aeb0700	Will Braynen <w...	Jan 1, 2022 at 1:...

The rebase and merge makes previous commit and parent commit every time it rebases. It helps users to keep track of sections of project, to help them to which part is done and which part is not. It benefits for projects that has a small team (5~7 people).

Personally, I prefer the rebase and merge because it helps me to keep track of projects progress.

References:

Squashing Your Pull Requests:

https://cloudfour.com/thinks/squashing-your-pull-requests/

Introduction to Git rebase and force push:
https://docs.gitlab.com/ee/topics/git/git_rebase.html