



CLI: Tooling

Google Summer of Code 2020

Nitin Kumar

Github - [@snitin315](#)

LinkedIn - <https://www.linkedin.com/in/snitin315/>

New Delhi, India

About Me

I am currently in the 2nd year of my Bachelor in Technology Degree from Maharaja Agrasen Institute of Technology (Affiliated to GGSIPU) in Electronics and Communications Engineering. I have been tinkering with code since my first year. I love working on Web Apps, as well as the related tools & technologies which make Web Apps possible and I have spent many of my nights up hacking on such projects. I also regularly participate in a lot of hackathons and Open-Source contests.

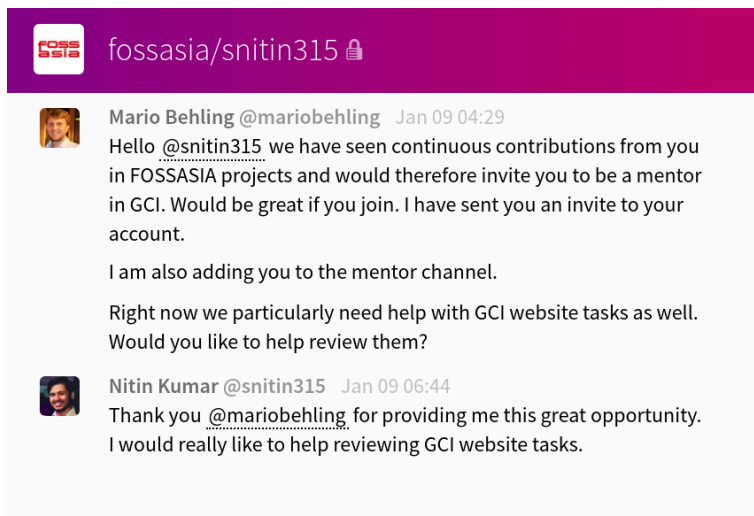
Most recently I was part of the [Codeheat](#) contest by [FOSSASIA](#) as a **student developer** and part of the **10th** edition of **Google Code-In** Contest as a **Mentor** under FOSSASIA organization.

These contests taught me a lot about **Teamwork, Problem Solving, Ability to find and fix bugs, Review code of other teammates, etc.**

I learned how to provide **efficient and high-quality code** for global projects used by thousands of people across the globe.

I was a [Finalist Winner of the Codeheat Contest](#) organized each year by FOSSASIA (One of the largest Open Source organizations in Asia), working on the Open Event Webapp project.

Seeing my contributions in FOSSASIA organization, Mario Behling (Co-Founder of FOSSASIA) [Invited me to be a mentor](#) in Google-Code-In 2019-20. Hence, I spent my winters mentoring pre-university students throughout the development of projects.



Overview

I wish to contribute to the development of [webpack-cli](#). I have been engaged in development since the last week of February 2020 and have successfully merged Pull Requests for adding features, finding & fixing bugs, writing tests and enhancing the documentation as well. I want to take that work forward and also contribute to the related webpack repositories.

There are a number of existing problems to be resolved, as well as multiple planned features that I plan to work on.

Work Experience (Open Source)

→ WEBPACK-CLI - @snitin315 (Webpack Contributions)

- ◆ I am contributing to the development of webpack-cli since the last week of February 2020. My contributions Include - Fixing bugs, adding features, writing tests, improving the documentation, triaging issues and reviewing code.
- ◆ **Major Contributions (Upto March 2020) -** Added `--mode` argument validation, `--no-mode` flag, `--no-hot` flag with corresponding tests.
 - My Commits
 - Pull Requests
 - Issues Created

→ FOSSASIA - @snitin315 (FOSSASIA Contributions)

- ◆ Contributing to the development of [open-event](#) & [susi.ai](#) web app Since November 2019, My contributions Include Code commits, Creating issues, fixing bugs, sharing daily scrums and reviewing code.
- ◆ **Major Contributions -**
 - My Commits
 - Pull Requests
 - Issues Created

→ Google Code-In Mentor-

- ◆ I spent my winters mentoring pre-university students in the 10th edition of Google Code-in under [FOSSASIA](#)
- ◆ Reviewed Tasks submitted by the students and helped them throughout the development of the official <https://gci19.fossasia.org/> website.

Goals / Deliverables:-

→ CURRENT ISSUES -

- ◆ Reduce down the current number of issues as much as possible.
- ◆ Some major issues I am aiming to solve are - [#789](#) , [#404](#) , [#1261](#) , [#866](#) , [#1264](#) , [#803](#) , [#1368](#) , [#1363](#) , [#575](#) , [#1240](#) .

→ Make the CLI More Emphatic -

- ◆ Validate the arguments passed by the user before passing their value to the webpack compiler so as to avoid errors.
- ◆ Provide a more colorful experience to the user
- ◆ **Solution Approach -**

- I will start by improving the use of **chalk** for a more colorful experience. Most of the promise rejections and logger warnings are colorless at the moment which makes it difficult to understand with the output format. Anyone can easily miss the warnings emitted by CLI. Right now the current behavior is like below -

```
root@nitin:~/Desktop/test/webpack-cli# webpack-cli
Hash: 1a66634a3fd6c1d952a0
Version: webpack 4.42.1
Time: 50ms
Built at: 03/31/2020 11:04:57 AM
Asset      Size  Chunks  Chunk Names
main.js    948 bytes  0 [emitted]  main
Entrypoint main = main.js
[0] ./src/index.js 19 bytes {0} [built]
[0] ./src/index.js 19 bytes {0} [built]
WARNING in configuration
The 'mode' option has not been set, webpack will fallback to 'production' for this value. Set 'mode' option to 'development' or 'production' to enable defaults for each environment.
You can also set it to 'none' to disable any default behavior. Learn more: https://webpack.js.org/configuration/mode/
root@nitin:~/Desktop/test/webpack-cli#
```

After correct use of **chalk**, it will be something like below -

```
root@nitin:~/Desktop/test/webpack-cli# webpack-cli
Hash: 1a66634a3fd6c1d952a0
Version: webpack 4.42.0
Time: 51ms
Built at: 03/31/2020 11:08:49 AM
Asset      Size  Chunks  Chunk Names
main.js    948 bytes  0 [emitted]  main
Entrypoint main = main.js
[0] ./src/index.js 19 bytes {0} [built]
[0] ./src/index.js 19 bytes {0} [built]
WARNING in configuration
The 'mode' option has not been set, webpack will fallback to 'production' for this value. Set 'mode' option to 'development' or 'production' to enable defaults for each environment.
You can also set it to 'none' to disable any default behavior. Learn more: https://webpack.js.org/configuration/mode/
root@nitin:~/Desktop/test/webpack-cli#
```

- Afterward, I will work on validating the arguments passed by the user before passing their value to the webpack compiler so as to avoid errors. Let's take an example of **mode**, to validate mode we need to improve the logic of **getEnvFromOptionsAndMode()** and **resolveZeroConfig()** functions in **zeroConfigGroup.js**, So that mode accepts only **development**, **production**, or **none** as value and should throw a warning else like *"You provided an invalid value for "mode" option. Using default value production."*

→ COMMANDS & ARGUMENTS -

- ◆ CLI arguments from core [webpack/webpack#10515](#) with tests.
- ◆ Allow multiple types of arguments.
 - `webpack --stats` (means `stats: true`)
 - `webpack --stats=verbose` (means `stats: verbose`)
- ◆ Allow multiple same arguments i.e, `webpack --entry=./src/one.js --entry=./src/two.js`.
- ◆ Allow negated flags like `--no-hot` , `--no-colors` , `--no-stats` etc.
- ◆ **Solution Approach -**
 - To solve the points above we need a good CLI helper and as **@webpack-cli/team** has decided to go with **commander.js**, My first step will be helping with the complete migration to **commander.js** from **command-line-args**.
 - After the migration I will first work on **negated flags**, I have already worked on this and have successfully added support of `--no-mode` ([#1276](#)) flag along with corresponding tests to CLI. I will be working on some major negated flags CLI should support like `--no-hot`, `--no-stats`, `--no-colors` and yes I will be writing tests for these at the same time as well.

→ TESTS -

- ◆ Improve smoke tests & write End-to-End Tests for the packages, commands, and arguments CLI supports.
- ◆ **Solution Approach -**
 - Right now we have low test coverage for the project, we don't have tests for some major commands like **init**, **migrate**, **loader** and **plugins**. I have worked on adding tests in the project and successfully added some tests for **--mode**, **--no-mode**, **writing-a-plugin**, etc.
 - I will be writing tests for **init**, **migrate**, **loader**, **plugin**, **utils**, **import-local**, etc
 - Other than that I will be writing tests for new features and fixes I will be adding within the corresponding PRs.

→ CODE REFACTOR -



- ◆ Right now the code is written with a mixture of TypeScript & JavaScript, the goal is to move towards TypeScript.
- ◆ There are various functions which are present in code but never used, the goal is to either remove them or improve them (e.g. [resolveNegatedFlags](#), etc.).
- ◆ **Solution Approach -**
 - At the moment the complete **webpack-cli** package is written in JS, I will start with migration **utils then groups and then the remaining files** of the package.
 - After migrating the **webpack-cli** package I will be migrating tests that are in JS. The end result will be something like -



→ Blog Posts

- ◆ A blog about the starting of a new experience in open-source with webpack.
- ◆ Write a blog wrapping up my whole experience during Google Summer of Code with WEBPACK.

Availability-

I would be able to devote **approx 50 hours to GSOC** every week. From **June 1 to June 10** I will have my end-sessional examinations, and time devoted would be around **1 hour** per day for that time. (**The dates are subjected to as INDIA has been locked Down for 3 weeks due to COVID-19 pandemic.**)

I have no obligations after June 10 and will be devoting all my time to GSOC.

Milestones-

→ BONDING TIME

- ◆ Fix existing bugs, help to merge pending PRs, and close issues and create milestones on Github projects.
- ◆ Discuss with mentors about roadmap & finalize the plan of action.

→ PHASE 1 -

- ◆ **FEATURE** - Add `--stdin` flag ([#789](#)).
- ◆ **FEATURE** - Allow Negated Flags (`--no-hot` , `--no-stats` , `--no-colors`).

- ◆ **FEATURE** - Allow multiple types of arguments.
 - `webpack --stats` (means **stats: true**)
 - `webpack --stats=verbose` (means **stats: verbose**)
 - `webpack --entry=./src/one.js --entry=./src/two.js` (Multiple same args)
- ◆ **TESTS** - Write tests for commands **init**, **migrate** & increase the test coverage.
- ◆ **DOCS (webpack-cli)** - Explain defaults for the arguments and commands CLI Supports.
- ◆ **BLOG** - Write a blog about the starting of a new experience in open-source with webpack.

→ PHASE 2 -

- ◆ **FEATURE** - Init command's splitChunks caching group generation per provided entry ([#404](#))
- ◆ **FEATURE** - Validate the arguments supported by webpack-cli. (**--entry**, **--config**, **--merge**, **--output**, etc.)
- ◆ **ENHANCEMENT** - Provide more visual & colorful experience to the user. By using **chalk** we can make the output, warnings, errors, promise rejections, etc. better for visuals. (Refers [#819](#) & [#1261](#))
- ◆ **TESTS** - Write tests for **loader** & **plugin**.
- ◆ **REFACTOR** -
 - Refactor the code present in JavaScript to TypeScript.
 - Remove or improve the unused functions, optimize the code.

→ PHASE 3 -

- ◆ **FEATURE** - Webpack Configuration Validation ([#803](#))

- ◆ **ENHANCEMENT** - Moves **types** to **@types/jscodeshift**. ([#866](#))
- ◆ **FEATURE** - Add an alias for version flag. ([#1368](#))
- ◆ **FEATURE** - **--version** for sub-packages. ([#1363](#)).

```
undefined@turing-machine ~/OSS/webpack/webpack-test node ./node_modules/webpack-cli/cli.js migrate --version
@webpack-cli/migrate 0.1.9
webpack-cli 4.0.0-beta.1
webpack 5.0.0-beta.3
undefined@turing-machine ~/OSS/webpack/webpack-test node ./node_modules/webpack-cli/cli.js info --version
@webpack-cli/info 0.2.0
webpack-cli 4.0.0-beta.1
webpack 5.0.0-beta.3
```

- The expected behavior is something like -
- ◆ **BLOG** - Write a blog wrapping up my whole experience during Google Summer of Code with WEBPACK.

→ **FUTURE WORK -**

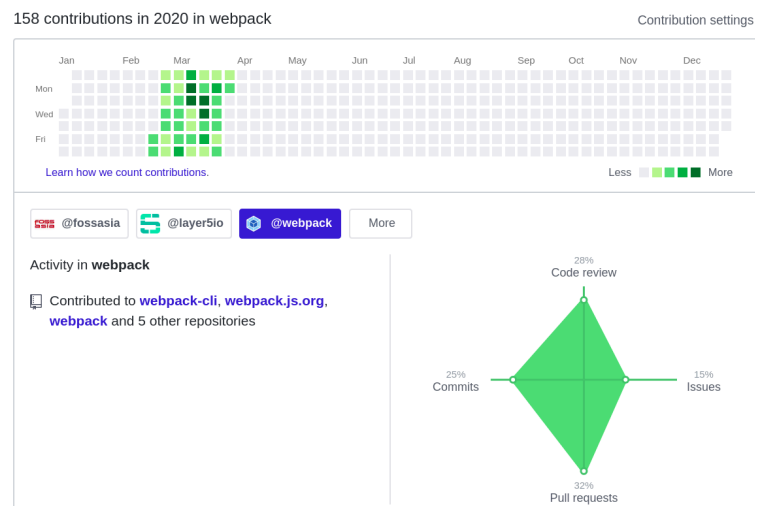
- ◆ Try to avoid using unnecessary dependencies, we should try to keep CLI small and fast.
- ◆ Section to test the CLI Locally, i.e, We could create a new package called **@webpack-cli/issue-repo** (*"name": "create-webpack-issue"*) which will be downloaded by people that wants to create a repo for an issue AND by people that are triaging the issue. ([Refer #1264](#))
- ◆ webpack interactive command.

Why me?

Other than being a self-taught developer for over 1 year, and a passionate open-source contributor, I believe that I am well suited for this project, because I am already working on the project, and I already know very well about the project vision and roadmap.

Following are some points that I would like to highlight -

→ **Consistency** - I am very consistent with my work, My contributions are regular & consistent in webpack since I have been engaged with the project.



→ **Active Contributions** - No Matter what the project is either **open-event** or **webpack-cli**, My contributions are the top contributions since I started contributing. (after maintainers).

Feb 10, 2020 – Mar 31, 2020

Contributions: Commits ▾

Contributions to next, excluding merge commits

