

Exploring the implementation of the AllContributors bot and research on other alternatives.

This is a compilation of the research I did on implementing the AllContributors Bot and alternatives to it. It begins with the description of the AllContributors Bot and how to implement it and only gets better. It describes an automation process that makes things easier after just one configuration. Then, there is a detailed research on an alternative to the AllContributors Bot which, interestingly, can be used on its own or combined with the AllContributors Bot for more efficient outcome. There is also a little bit on Git Action which is being used for the whole automation process. That's it!! I hope you have a great read and I'll be happy to get some feedback.

The AllContributors Bot

The AllContributors Bot is a GitHub bot that automates the process of adding contributors to a project and recognizing their contributions. This bot is built on top of the AllContributors specification, which provides a standardized way of recognizing and rewarding contributions to open source projects.

To add the AllContributors Bot to your repository, you need to follow these steps:

- Step 1: Create a GitHub App

To use the AllContributors Bot, you need to create a GitHub App. Go to the GitHub Developer Settings page and click on the "New GitHub App" button. Fill out the required fields such as the name and description of the app, and the callback URL.

- Step 2: Install the App in your Repository

Once you have created the GitHub App, you need to install it in your repository. To do this, go to the repository's settings page and click on the "Install App" button. Search for the AllContributors Bot and install it.

- Step 3: Configure the Bot

After you have installed the AllContributors Bot, you need to configure it. To do this, create a file called `.all-contributorsrc` in the root of your repository. This file contains the configuration for the bot. Here is an example configuration file:

```
{
  "files": [
    "README.md"
  ],
  "imageSize": 100,
  "contributorsPerLine": 7,
  "types": [
```

```
    "code",  
    "doc",  
    "test"  
  ],  
  "skipCi": true  
}
```

In this example configuration, the bot will recognize contributions in the README.md file, display contributor images at a size of 100 pixels, display up to 7 contributors per line, recognize contributions of types "code", "doc", and "test", and skip commits that have "[skip ci]" in the commit message.

- Step 4: Add Contributors

To add a contributor to your project, you need to use the all-contributors CLI tool. Install the tool by running the following command:

```
npm install -g all-contributors-cli
```

Once you have installed the tool, you can add contributors to your project by running the following command:

```
all-contributors add <username> <contributionType>
```

For example, if you want to add a contributor named "John" who contributed to the project by writing documentation, you can run the following command:

```
all-contributors add John doc
```

- Step 5: Update the Contributors List

After you have added a contributor to your project, you need to update the contributors list. To do this, run the following command:

```
all-contributors generate
```

This command will update the contributors list in the configured files (in this example, the README.md file).

- Step 6: Commit and Push

Finally, commit and push the changes to your repository. The AllContributors Bot will recognize the new contributions and update the contributors list accordingly.

In summary, adding the AllContributors Bot to your repository involves creating a GitHub App, installing it in your repository, configuring it, adding contributors using the all-contributors CLI tool, updating the contributors list, and committing and pushing the changes. With the AllContributors Bot, recognizing and rewarding contributions to your open source project becomes much easier and more streamlined.

Automating the AllContributors Bot with GitHub Actions

While this is already incredible, it is possible to automate the AllContributors Bot to update the contributors list automatically instead of manually using the all-contributors CLI.

One way to achieve this is by using GitHub Actions, which is a powerful tool for automating tasks on your repository. With GitHub Actions, you can define custom workflows that run automatically when certain events occur, such as when a pull request is merged.

To automate the process of adding contributors to the contributors list, you can create a workflow that does the following:

1. Install the all-contributors-cli tool as part of the workflow setup.
2. Configure the bot using a .all-contributorsrc file, just like you would do manually.
3. Define a custom script to automatically add contributors to the contributors list when a pull request is merged. This can be done using a GitHub Actions event that triggers when a pull request is merged, such as pull_request.closed. Within the script, you can use the all-contributors-cli tool to add the username of the person who merged the pull request and the type of contribution they made (e.g. "code", "documentation", etc.) to the contributors list.
4. Use the all-contributors-cli tool to update the contributors list and commit the changes to the repository.

Here's an example workflow that automatically adds contributors to the contributors list when a pull request is merged:

```
name: Update Contributors

on:
  pull_request:
    types: [closed]

jobs:
  update-contributors:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Install all-contributors-cli
```

```
run: npm install -g all-contributors-cli
- name: Add contributor
  run: all-contributors add ${{ github.actor }} code
- name: Generate contributors list
  run: all-contributors generate
- name: Commit changes
  run: |
    git config --global user.name 'GitHub Action'
    git config --global user.email 'noreply@github.com'
    git add .
    git commit -m "Update contributors list [skip ci]"
    git push
```

This workflow runs when a pull request is closed (i.e. merged). It installs the all-contributors-cli tool, adds the person who merged the pull request to the contributors list as a code contributor, generates the updated contributors list, commits the changes, and pushes the changes back to the repository.

By automating the process of adding contributors to the contributors list, you can save time and make sure that your contributors list is always up-to-date without needing to manually update it each time.

An alternative to ALLContributors Bot

GitHub's Contributors List is a built-in feature that allows maintainers to automatically generate a list of contributors to their repository. The contributors list can be accessed by visiting the repository's Insights tab and selecting the "Contributors" option.

The contributors list is generated based on the number of commits that each contributor has made to the repository. This means that contributors who have made more commits will appear higher on the list.

The contributors list is a great way for maintainers to recognize and showcase the contributions of individual contributors to their project. It can also be used to identify the most active contributors, and to encourage new contributors to get involved in the project.

In addition to the built-in contributors list, there are also third-party tools and integrations available that can enhance the functionality of the list. The All Contributors bot, can be used to automatically add contributors to the contributors list and provide additional information about their contributions.

Overall, the contributors list is a valuable tool for maintaining a healthy and engaged open source community on GitHub, and is a great way to recognize and showcase the contributions of individual contributors.

The GitHub's Contributors List is automatically generated based on the number of commits that each contributor has made to the repository. This means that maintainers do not need to manually add or update the contributors list - it is updated automatically as new commits are made to the repository.

GitHub's algorithm for generating the contributors list takes into account both the number of commits and the number of lines changed by each contributor. This helps to ensure that the list accurately reflects the contributions of each individual contributor, and that more substantial contributions are given greater weight.

Overall, the automated nature of the contributors list makes it a convenient and reliable way for maintainers to recognize and showcase the contributions of individual contributors to their project.

A little bit about Github Actions

GitHub Actions is a powerful workflow automation tool built into the GitHub platform. It allows developers to automate many tasks related to software development, such as building, testing, and deploying code, as well as performing other actions such as sending notifications, running scripts, and interacting with external services.

With GitHub Actions, developers can create workflows that can be triggered by various events, such as a push to a repository, a pull request, or a scheduled event. Workflows are defined using a YAML file, and they consist of one or more jobs that run in parallel or sequentially. Each job can include multiple steps, which are individual commands or actions that perform specific tasks.

GitHub Actions also provides a large library of pre-built actions that developers can use in their workflows, as well as the ability to create custom actions to meet their specific needs. The tool integrates with many popular development tools and services, such as Docker, AWS, and Slack, allowing developers to build powerful, integrated workflows that streamline their development processes.

Overall, GitHub Actions is a powerful tool that can save developers time and automate many routine tasks associated with software development, making it easier to focus on building great software.

