Introduction to OpenSauced 🍕

Introduction to OpenSauced 🍕

Watch this on YouTube.

OpenSauced is a platform dedicated to empowering maintainers and teams and redefining the meaning of open source contributions. We believe that every contribution, big or small, is valuable and deserves to be recognized. Our platform provides a way to track contributions through a GitHub-powered dashboard, allowing users to showcase their contributions and gain insights into their open source projects.

Our mission is to:

- **Empower maintainers and teams**: Track and showcase your contributions and projects, connect with contributors, collaborate with your team, and gain insights into your open source projects.
- **Support contributors**: Showcase your contributions, tell your story, connect with new projects and other contributors, and gain insights into your open source journey.
- **Highlight success**: Showcase your contributions with our shareable Highlights, inspiring others to embark on their open source journey.
- **Unlock Insights**: Understand the health of your project, track contributions, and make data-driven decisions with OpenSauced Insights.

What is OpenSauced?

OpenSauced: Your Open Source Partner

OpenSauced is a platform dedicated to nurturing and expanding your open source community while recognizing every contributor's impact.

Our OpenSauced dashboard allows you to:

- **Understand open source projects:** Get a comprehensive view of open source contributors and their engagement.
- **Discover valuable insights:** Identify key contributors, emerging trends, and valuable project insights.
- **Engage effectively:** Reach out to community members to foster involvement and collaboration.
- **Generate reports:** Create insightful reports to visualize and communicate the influence of your community.
- **Collaborate with your team:** Work together with your team or other contributors for a holistic approach to open source.
- **See beyond the green squares:** Connect with various aspects of open source contribution and community activities.

Who is OpenSauced for?

OpenSauced is for maintainers, Open Source Program Offices (OSPOs), contributors, companies, community managers, and anyone interested in what's happening in the open source ecosystem.

Embarking on an OpenSauced journey enhances your growth and community engagement. Let's launch your open source success story! 🚀

OpenSauced Glossary

Welcome to the glossary, where we define concepts used on our platform to help you get the most out of OpenSauced.

Active Contributors

A contributor who has been active in the last 30 days.

Activity

Activity tracks Pull Requests (PRs) over the last 30 days:

- **High**: More than 80 PRs.
- **Medium**: Between 5 and 80 PRs.
- **Low**: Below 5 PRs.

Alumni Contributors

A contributor who has not been active on a project in the last 30 days.

Insights

The Insights feature is a one-stop solution for understanding the pulse of your open-source projects and contributions. This feature is designed to provide a comprehensive view of your project's health and your individual contributions.

Lists

The Lists feature enables you to categorize, monitor, and analyze different groups of contributors within open source projects. With the Lists feature, gain granular insights into each contributor's activity and contributions.

New Contributors

A contributor who has recently made their first contribution to a project.

PR Overview

PR Overview tells the story of the Pull Requests (PRs) submitted in the last 30 days:

- **Green line**: Open PRs.
- **Red line**: Closed PRs.
- **Purple line**: Merged PRs.
- **Grey line**: Draft PRs.

PR Velocity

PR Velocity tracks how quickly Pull Requests (PRs) are being merged in by identifying the average time to completion from the request submission.

Frequently Asked Questions

Contributions

1. How do I find good first issues?

We believe that good first issues don't exist, and the best way to find an issue to work on is to create the issue yourself.

However, sites like Good First Issues and First Timers Only can lead you to projects and issues worth working on. You can also use the `good first issue` label in the project search engine you want to contribute to. Check out this guide to learn more about finding a `good first issue` on GitHub..

Additional Resources:

- "Who's looking for open source contributors?" series on Dev.to
- Intro to Open Source Course with OpenSauced

2. My contribution does not show up on my OpenSauced profile. How do I fix it?

Create an Insights Page or add your merged pull request to a Highlight.

3. I want to provide feedback on OpenSauced.

We love feedback. Post your suggestion in our feedback repository.

Sync your GitHub Team

1. I don't see my organization on "Sync your GitHub Team".

You may not see your organization if you are not a "public" member. You can update this by going to the organization on GitHub and selecting the "People" tab. Next, search for your name. You can then change "private" to "public." For complete instructions, see this guide.

2. I get an error when I try to select my team in "Sync your GitHub Team".

If you see your organization but get an error when selecting a team, your organization likely has restricted third-party applications. You can check this if you are an owner by going to "Your organizations" on GitHub, selecting "Settings", and then "Third-party Access".

If you're an **owner** of the repository, you can select "Allow select third-party access" or specifically approve OpenSauced by following the directions in this guide.

If you are not an owner of the organization, you will need to check with your organization's owner and request access. For complete instructions, see this guide.

Highlight Your Open Source Contributions

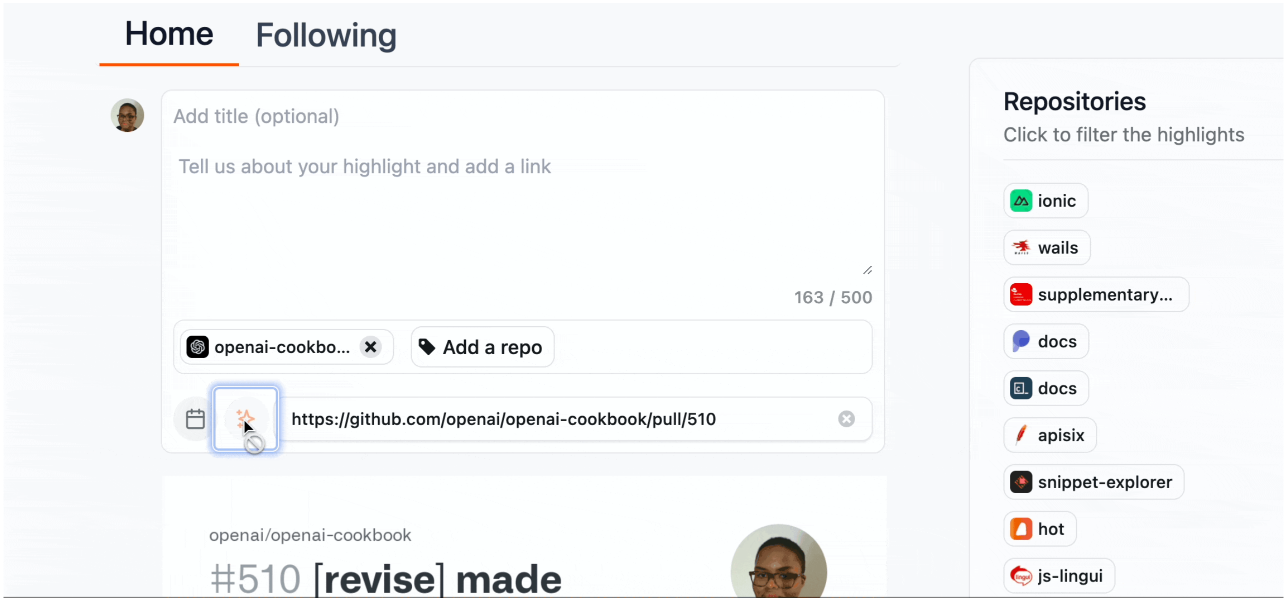
The Highlights feature is the place you can display your favorite open source contributions, share the story, and inspire others to join you in your open source journey. For maintainers, it's a great way to showcase your project and the issues that need support and attract new contributors.

There are currently three types of highlights you can add to your profile:

- Dev.to blog post
- Pull request
- Issue

How to Add a Highlight

- Go to OpenSauced's Highlights feed.
- Click the "Post a highlight to show your work!" input.
- Paste the URL to your blog post, pull request, or issue in the bottom input.
- Either use our Auto-Summarize feature or write your own summary.

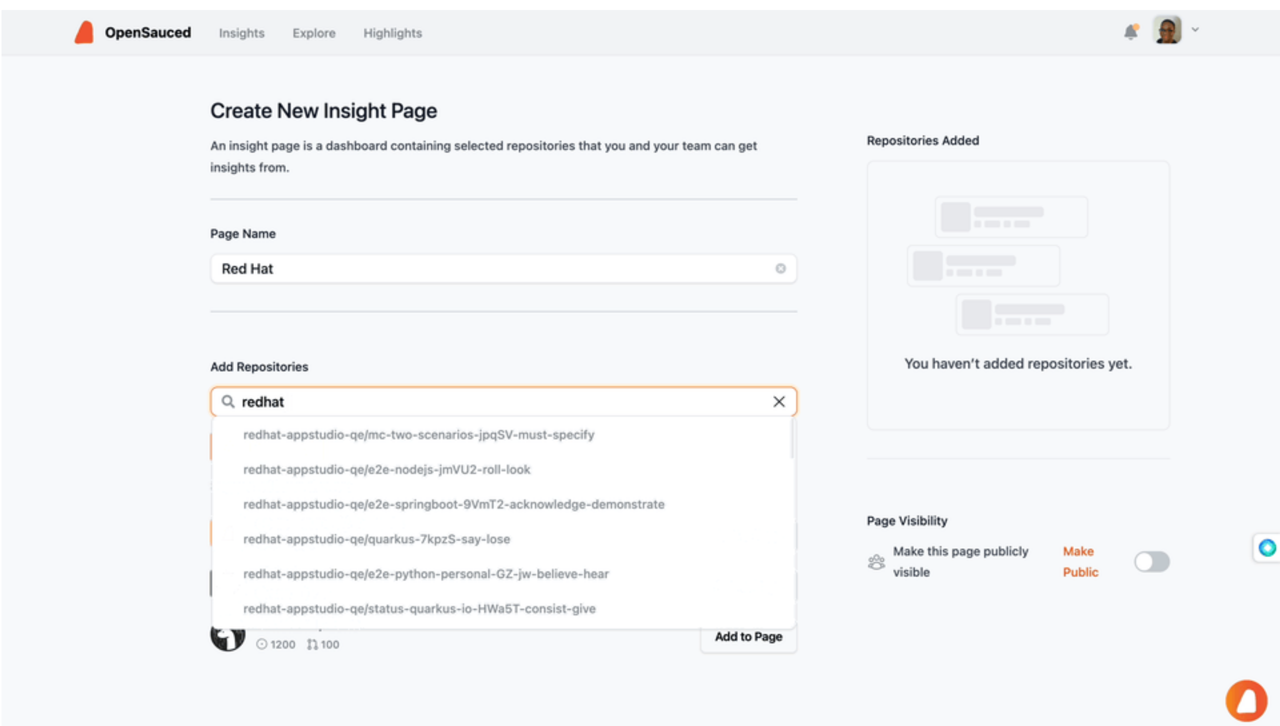


For more examples of highlights, go to the OpenSauced Highlights feed.

Eager to get started? Check out the Effectively Highlight Your Contribution section in our free "Intro to Open Source" course to learn more.

Insights into Open Source Projects

The Insights feature is your one-stop solution for understanding the pulse of your open-source projects and contributions. This feature is designed to provide a comprehensive view of your project's health and your individual contributions. From contribution trends to community health, OpenSauced Insights helps you make data-driven decisions that align with your goals. It's not just about numbers; it's about providing data to help you make decisions that can define your approach to open source and tell the story of your project or contribution journey.



How to Use Insights

Contributors

- Track your contributions and the impact you're making.
- Find new projects to contribute to.
- Discover other contributors to collaborate with.

Maintainers

- Track the health of your project.
- Identify contributors who are making an impact.
- Find new contributors to support your project.

To learn more about how to use Insights, check out our Maintainers Guide and our Understanding Insights Data pages.

Sharing Your Insights Page with Your Team

After you've created your Insights Page, share it with your team.

Sharing this page with your team ensures that everyone is well-informed about the project's status and can collaborate more effectively. It fosters a collaborative environment where team members can align their efforts, make informed decisions, and collectively work toward the organization's success.

How to Share Insights Page

There are two methods to share your Insights Page:

Method 1: Using the Share Button

This method gives view-only access to your Insights Page.

1. Click the "Insights" tab at the top bar.
2. Click the page that you want to share.
3. On the top right, click the "Share" button to copy the Insights Page URL to the clipboard.
4. Share the URL with your team members.

Method 2: Adding a Team Member

This method allows you to invite your team members to your Insights Page and offers you options to permit them to be an admin of the page, edit or view the page, or remove them from it.

1. Click the "Insights" tab at the top bar.
2. Click the page that you want to share.
3. Click the "Edit Page" button on the top right.
4. Scroll down to the "Add Team Members" section.
5. Enter the email address of the team member you want to invite.
6. Click the "Send invite" button.
   Once you click the button, your team member's email will be listed at the bottom.
7. To select and give permission to your team member, click the arrow button on the right side of their email.

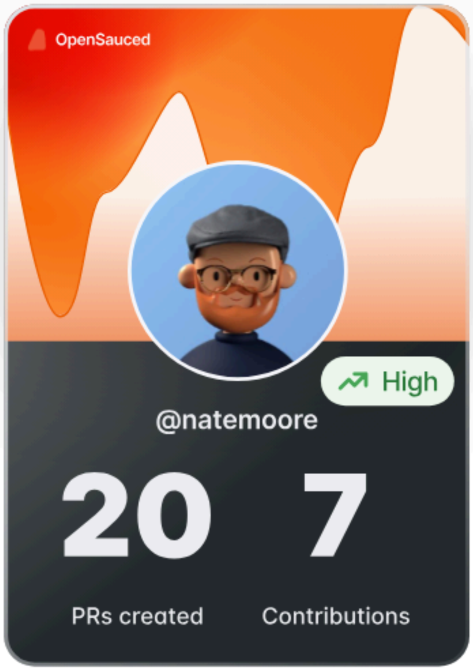
To learn more about inviting your team, check out this tutorial video.

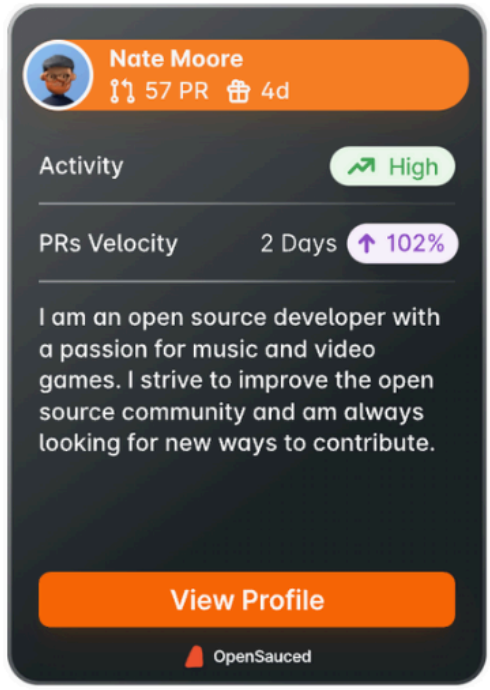
Showcase Your Open Source Stats with Your Dev Card

The Dev Card feature is what we like to call your "open source business card."

On the front of your card, you see your profile picture, username, the number of pull requests you have created, the number of repositories you contributed to, and a graph icon that describes your activity rate.

You'll see the back side of your card when you click it. It shows more information, including your profile's picture and headline, the number of pull requests you have created, the date of your first contribution on GitHub, activity rate, PR velocity rate, and a `View Profile` button that takes you to your dashboard.

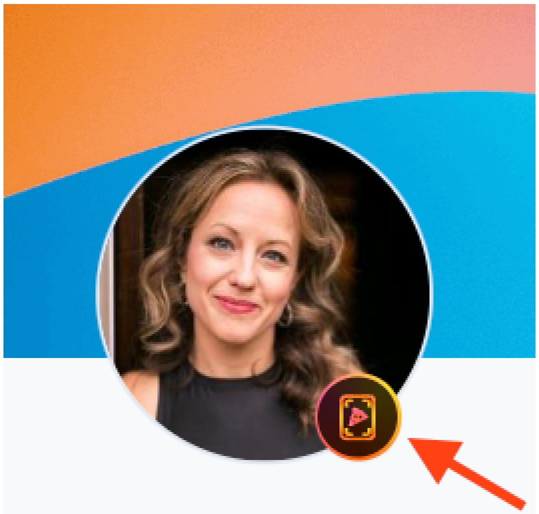




Whether you are just beginning your open source journey or a maintainer looking for more contributors and sponsors, you can share your Dev Card as a snapshot of your contributions, skills, and engagement in the open source community.

Eager to get your Dev Card? Here's how you can get yours today:

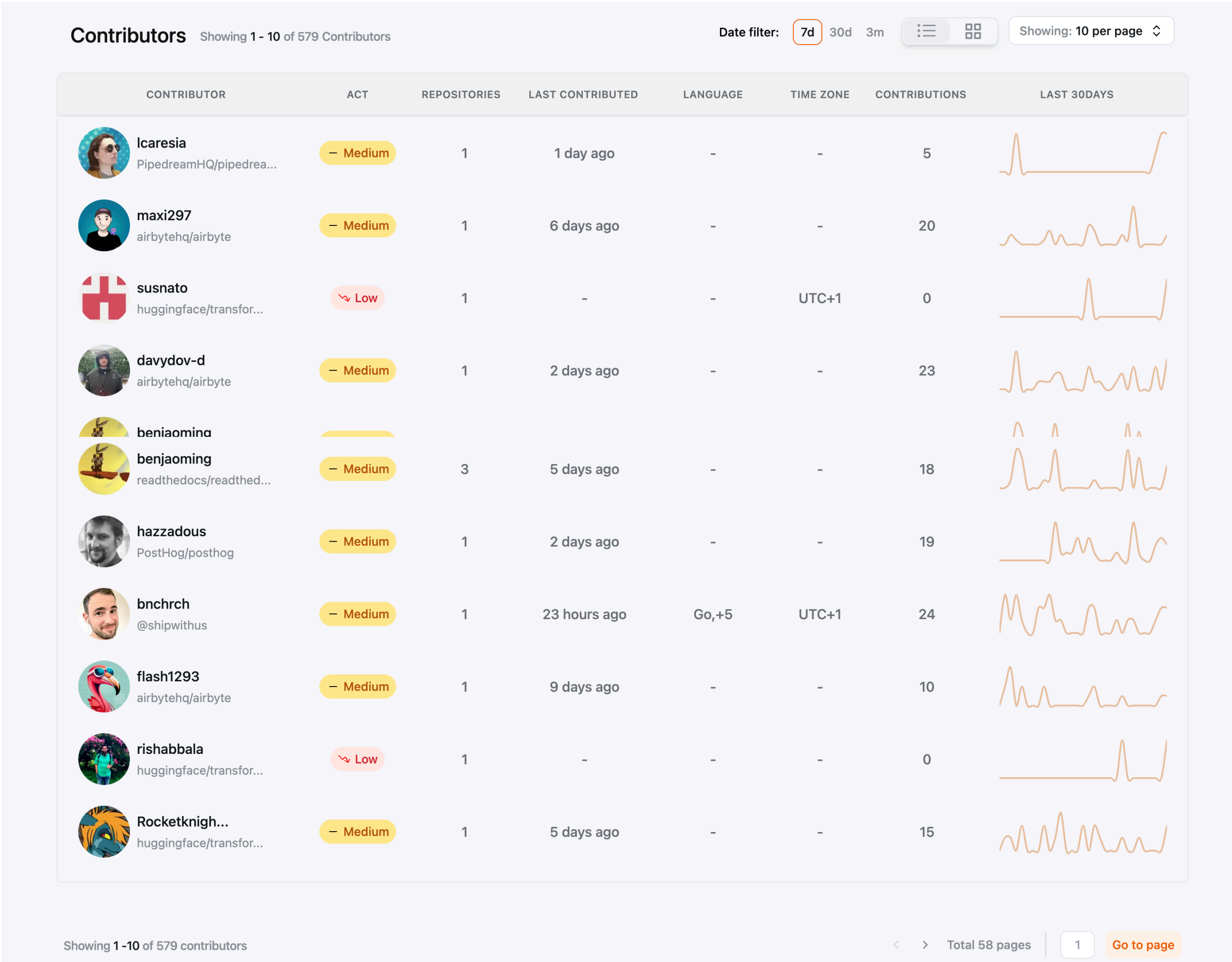
1. Log in to your OpenSauced account.
2. Navigate to your profile.
3. Click the Dev Card icon on your profile image.



Organize and Monitor Your Contributors

The Lists feature enables you to categorize, monitor, and analyze different groups of contributors within open source projects. With the Lists feature, gain granular insights into each contributor's activity and contributions.

Monitor individual commit histories, track who is actively engaged, identify alumni or new contributors, and even compare the performance of contributors against each other. This encompasses a wide range of metrics such as code commits, created and reviewed Pull Requests (PRs), as well as issue creation and commenting.


Contributors Showing 1 - 10 of 579 Contributors

Date filter: 7d 30d 3m Showing: 10 per page

CONTRIBUTOR	ACT	REPOSITORIES	LAST CONTRIBUTED	LANGUAGE	TIME ZONE	CONTRIBUTIONS	LAST 30DAYS
lcaresia PipedreamHQ/pipedrea...	Medium	1	1 day ago	-	-	5	
maxi297 airbytehq/airbyte	Medium	1	6 days ago	-	-	20	
susnato huggingface/transfor...	Low	1	-	-	UTC+1	0	
davydov-d airbytehq/airbyte	Medium	1	2 days ago	-	-	23	
benjaoming readthedocs/readthed...	Medium	3	5 days ago	-	-	18	
hazzadous PostHog/posthog	Medium	1	2 days ago	-	-	19	
bnchrch @shipwithus	Medium	1	23 hours ago	Go,+5	UTC+1	24	
flash1293 airbytehq/airbyte	Medium	1	9 days ago	-	-	10	
rishabbala huggingface/transfor...	Low	1	-	-	-	0	
Rocketknigh... huggingface/transfor...	Medium	1	5 days ago	-	-	15	

Showing 1 -10 of 579 contributors

Total 58 pages 1 Go to page

How to Create a List

After logging in, navigate to your Lists Page.

There are three ways to add a list:

1. **Explore Contributors**: Use our explore tool to find contributors and create your list.
2. **Sync your GitHub Team**: Connect to your GitHub to create a list from a team in your organization.
3. **Import your GitHub Following**: Connect to your GitHub to create a list

with all the contributors you follow.

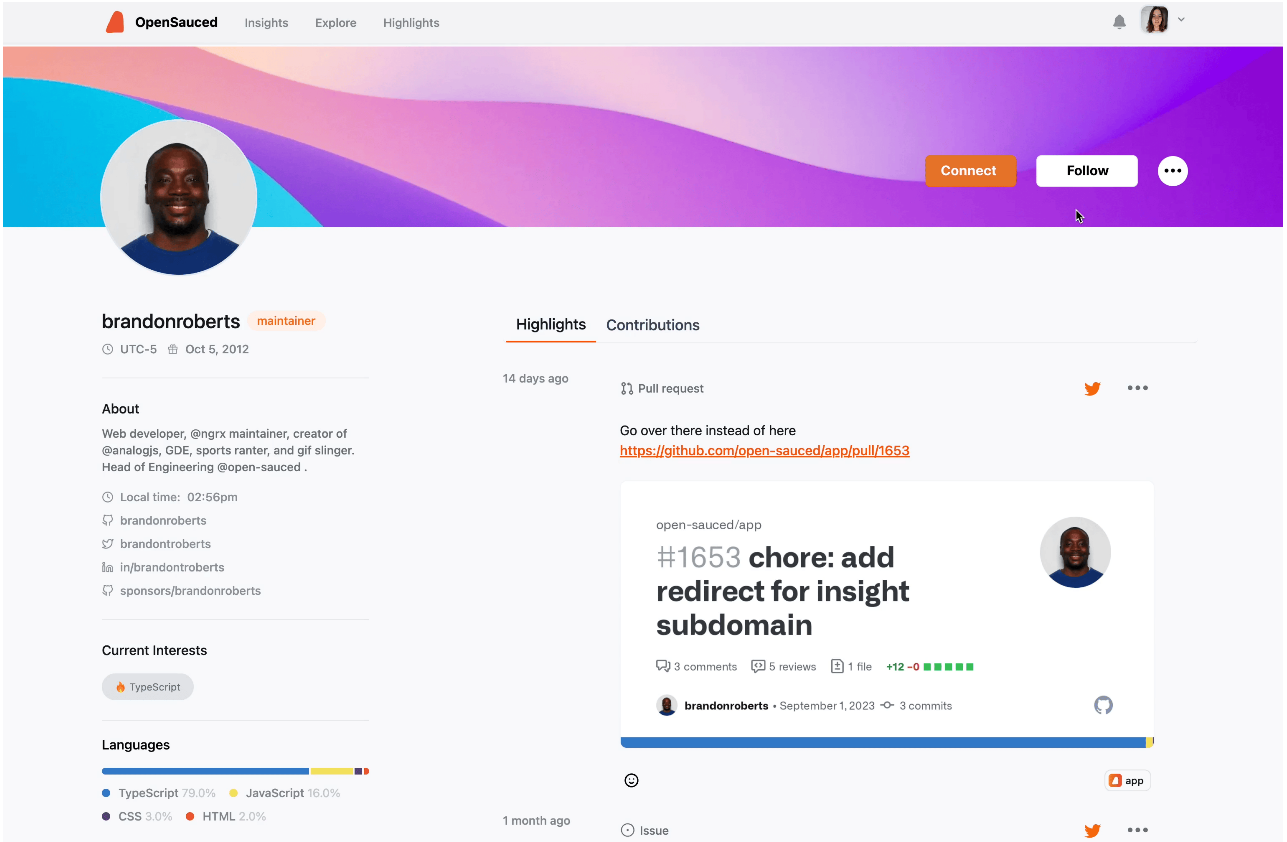
Once your list is created, you'll be able to see an overview of all the contributors in your list, a graph of their activity, and a detailed breakdown of the contributors.

If you're having trouble syncing your Organization or Team, check out our Sync Your GitHub Team in our FAQs.

Grow Your Network with Our Connections Feature!

What are Connections?

Connections on OpenSauced serve as a way to build and nurture your professional network within the open source ecosystem. Imagine you're searching for experts in Artificial Intelligence using our Lists feature, and you come across several promising developers. What's next? With Connections, you can now send a request to connect, bringing these experts into your professional circle for easy communication and collaboration.



How Do Connections Work?

- **Sending connection requests**: Once you identify users you'd like to connect with, you can send them a connection request.
- **Acceptance**: Upon acceptance of your connection request, both parties become 1st-degree connections.
- **Following highlights**: After becoming 1st-degree connections, you can follow each other's highlights, keeping up-to-date with accomplishments, contributions, and other notable activities.
- **Contact information**: Post-acceptance, you'll have access to contact information to foster direct communication and ongoing dialogue.

Are you interested? Navigate to OpenSauced, create your Pro account, and start building!

Maintainers Guide to OpenSauced

Introduction to OpenSauced for Maintainers

OpenSauced is a platform dedicated to empowering maintainers and redefining open source contributions. We offer a GitHub-powered dashboard for tracking contributions and gaining insights into your open source projects. You can customize your Insights Pages and Lists with the repositories you choose and the contributors you want to see.

Getting Started

To get started, you will need to create an account on OpenSauced. You can do this by visiting the app.opensauced.pizza and clicking the "Connect with GitHub" button. You will be prompted to sign in with your GitHub account. Once you have signed in, you will be redirected to the OpenSauced dashboard.

Insights: Connecting Your Repositories

OpenSauced Insights provides maintainers with a visual representation of their project's health and contributor engagement. It offers real-time data and analytics to understand contribution patterns, identify key contributors, and measure the project's growth. Insights can help you assess team activity, understand when bottlenecks occur, identify where resource allocation is needed, plan and identify key contributors, understand alumni contributors, or spot emerging trends.

Insights features allow you to:

- track the contributions of a specific team, a specific repository, or a group of repositories,
- view the number of pull requests that are open, closed, in draft, and merged,
- view the average number of days it takes for a pull request to be merged (PR velocity),
- identify the top contributors to your project,
- identify the current activity level and trends in activity of your contributors.

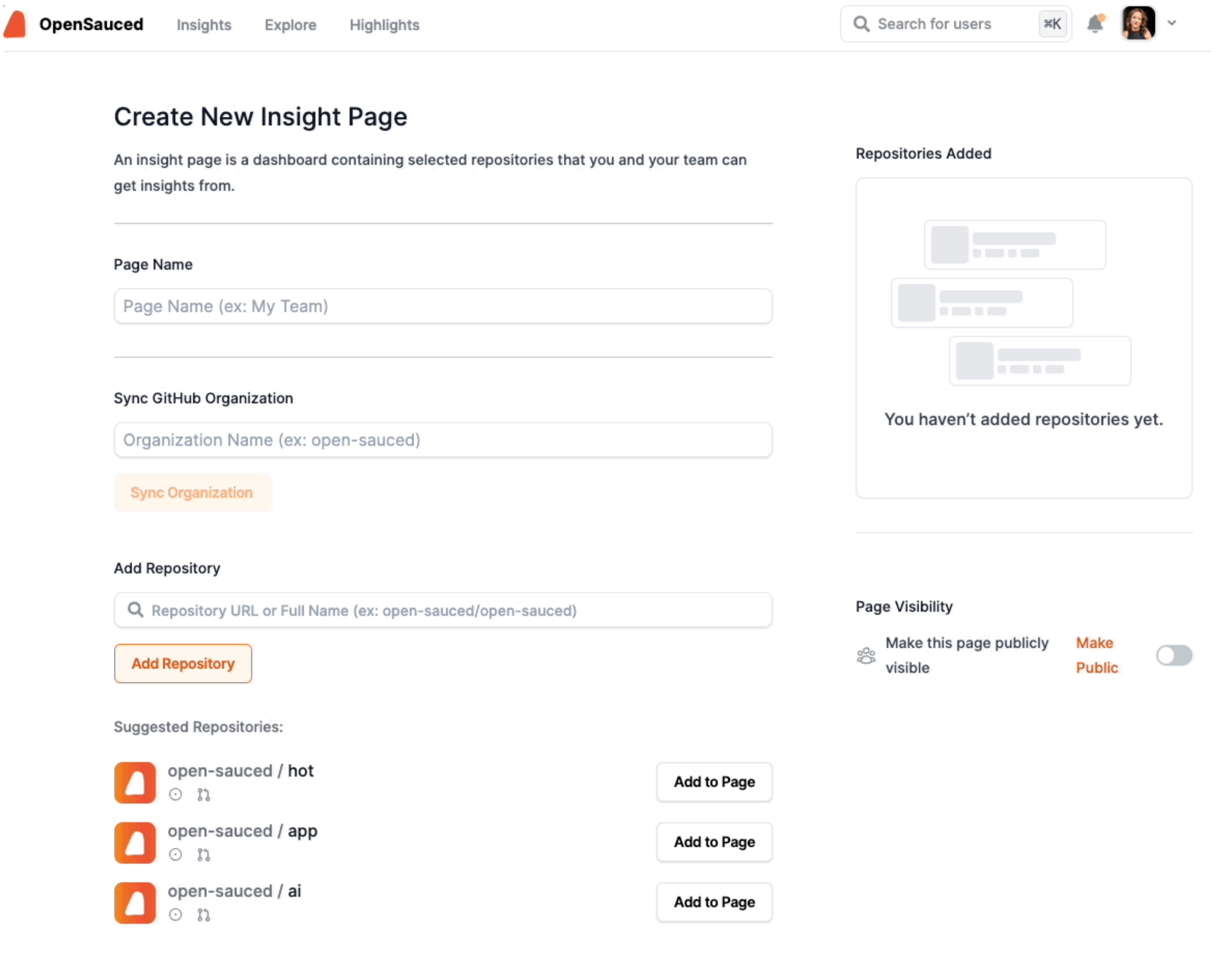
Creating a New Insight Page

To connect your repositories, click the "Insights" tab on the dashboard. You will be redirected to a page where you can create your new Insight Page.

There are two ways to add repositories to your Insight Page:

1. **Sync GitHub organization**. Syncing your GitHub organization is a good idea for someone who seeks a comprehensive, unified view of your organization's open source project activities and trends, enabling streamlined management, strategic decision-making, and efficient allocation of resources across multiple repositories.
2. **Connect individual repositories**. Connecting individual repositories is a good idea for someone who wants to track the contributions of a specific repository or group of repositories. For example, you might want to track

projects you maintain that are not under the same organization, or you may want to track projects that align with your team's objectives.



Using Your Insights Page

Once you have connected your repositories, you will be redirected to your Insight Page. Here, you will see a dashboard with an overview of the repositories and the contributors who have contributed to them.

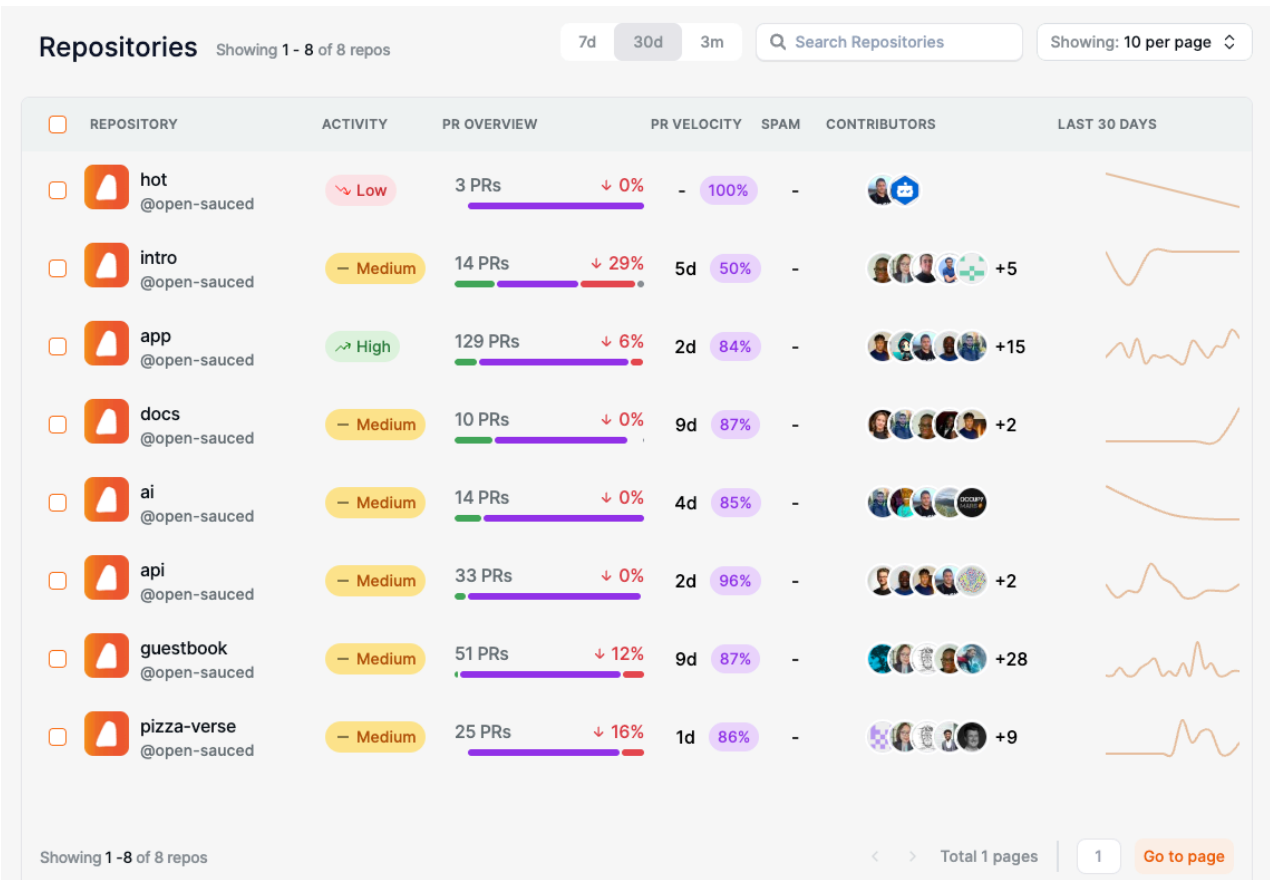
There are three tabs that provide you with more information on these repositories:

Reports

The Reports tab allows paid users to filter for more information about their repositories over the last 30 days and to generate a Downloadable CSV. Filters include: Top 100 Repos, Minimum 5 Contributors, Recent, Most Active, and Spammed.

Repositories

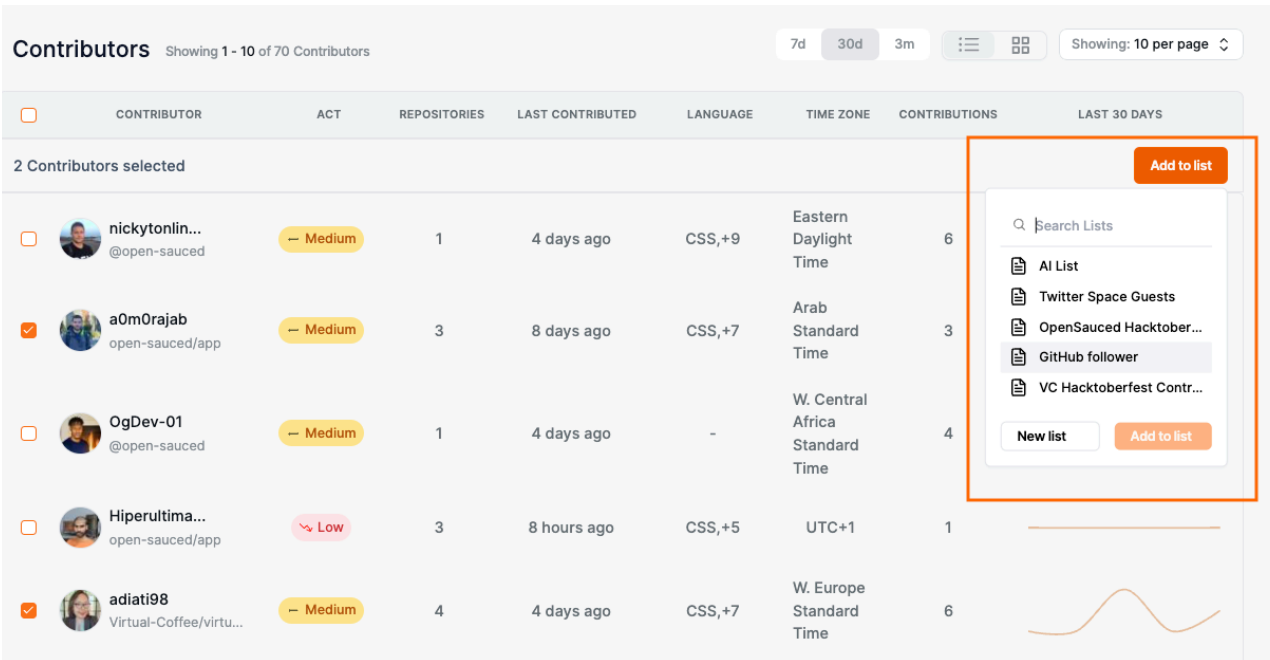
The Repositories tab allows you to view more detailed information on each repository, including Activity, PR Overview, PR Velocity, Spam, Contributors, and activity over the last 30 days. To learn more about these features, see Understanding Insights Data.



Contributors

The Contributors tab allows you to view more detailed information on each contributor, including Activity, Repositories, the date of their last contribution, Time Zone, and the number of contributions.

To learn more about your contributors, you can select them and add them to a list.



There are two ways to add contributors to a list:

1. Create a new list with selected contributors.
2. Add selected contributors to an existing list.

Lists: Connecting with Contributors

OpenSauced Lists feature allows you to categorize, monitor, and analyze various groups of contributors. You can use Lists to track the contributions of a specific team, a specific repository, or a group of repositories. This feature offers granular insights into each contributor's activities and contributions, helping maintainers to track individual contribution histories, identify active engagement, recognize new or alumni contributors, and compare the performance of different contributors.

The Lists features allow you to:

- track the commits and PRs of the contributors on your list,
- filter your contributors by their activity level, including most active, new, and alumni,
- view the repositories that your contributors have contributed to, the top programming languages they've used, and their time zones.

Why Create a List?

There are many reasons to create a list, but here's some inspiration to get you started:

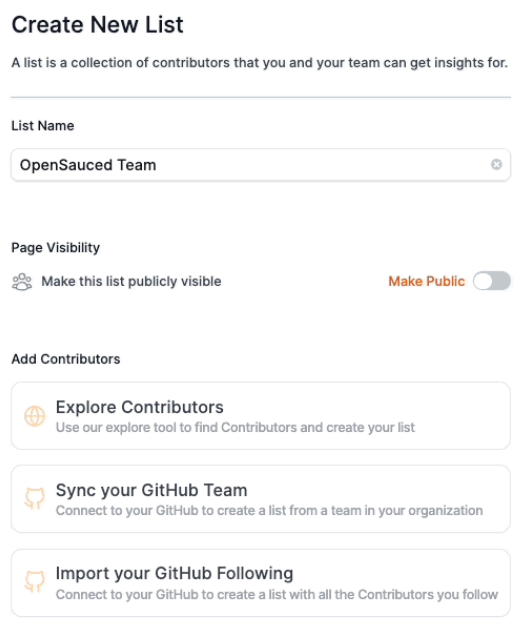
- **Interacting contributors**: Keep tabs on contributors actively interacting with your repository.
- **Organization's developers**: Monitor the contributions and performance of your internal team.
- **VIP contributors**: Highlight and track at least five developers who are key

maintainers or significant contributors.

- **Sales leads**: Identify contributors whose engagement could potentially be turned into sales opportunities.
- **Potential hires**: Observe contributors who exhibit skill and commitment, making them prime candidates for hiring.
- **Alumni contributors**: Keep an eye on contributors who have decreased their engagement or left the project.

Creating a New List

1. Click the "Lists" tab on the Insights hub to create a new list. You will be redirected to a page where you can create your new list.
2. Click on the "New List" button to start a new list.
3. Give your list a name.
4. Choose your page visibility. You can choose to make your list public or private. By default, it is set to private.
5. Add contributors to your list by searching for their GitHub username, syncing your GitHub Team, or importing your GitHub Following.



Using Your List

Once you have created your list, you will be redirected to your List Page. Here, you will see a dashboard with an overview of the contributors.

There are three tabs that provide you with more information on these repositories:

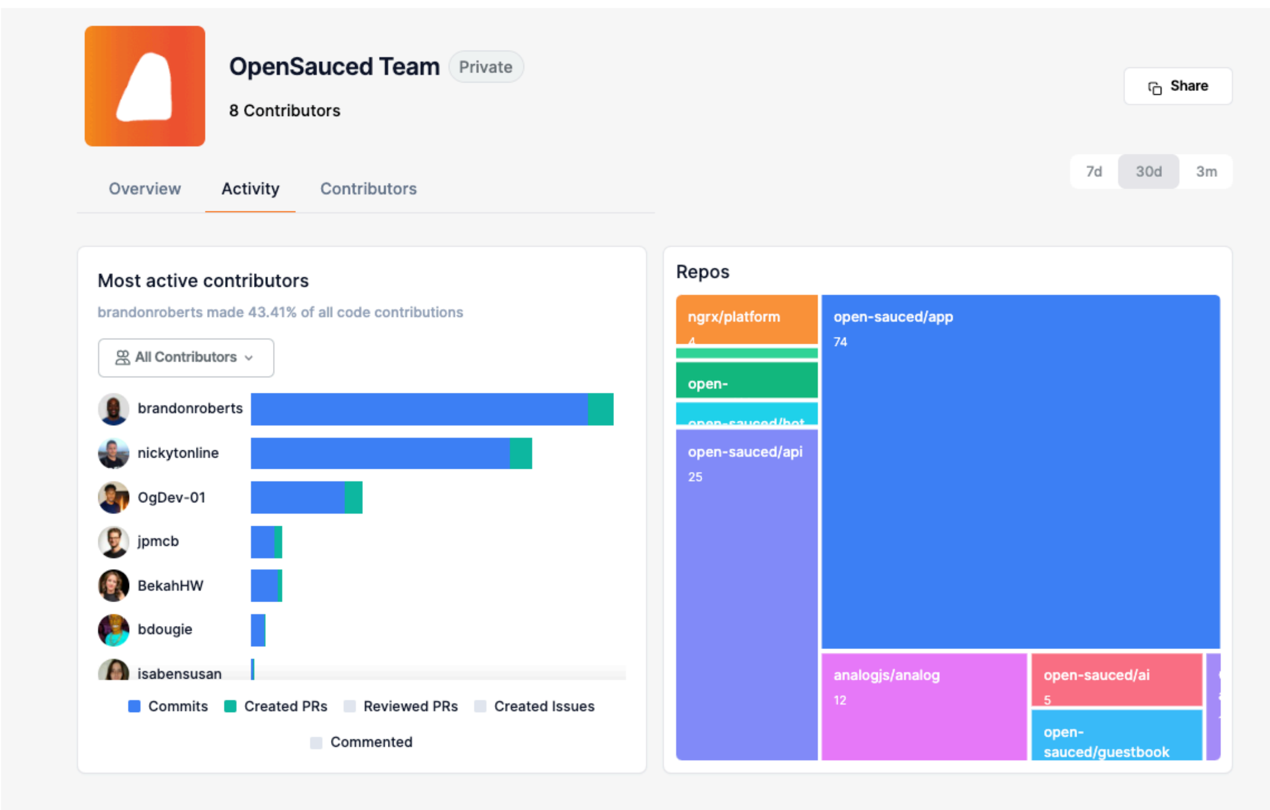
Overview

The Overview tab gives you a high-level view of the contributors in your list, including the total number of commits in the last 30 days and the types of contributors on your list: Active, New, and Alumni.

Activity

The Activity tab gives you a graph view with more detailed information on each contributor, including the type of activity, repositories they've contributed to, and how they compare to each other.

You can filter your list by All Contributors, Active Contributors, New Contributors, and Alumni Contributors.



Contributors

The Contributors tab gives you the list of the contributors and includes information like activity level, last repository contributed to, the date of their last contribution, time zone, programming languages used, and number of contributions.

To learn more about these features, see Understanding Lists Data.

Highlights: Connecting and Recognizing Contributions

Leveraging OpenSauced's Highlights feature is a great way for maintainers to recognize and showcase the valuable contributions made to your open source projects, to curate and share key achievements, contributions, and milestones, and bring visibility to the hard work of your community.

Highlights can be used to:

- publicize issues on your repository that you'd like to match with contributors,
- share Dev.to blog posts you've written,
- celebrate PRs for your project.

Creating a New Highlight

There are two ways to create a new highlight:

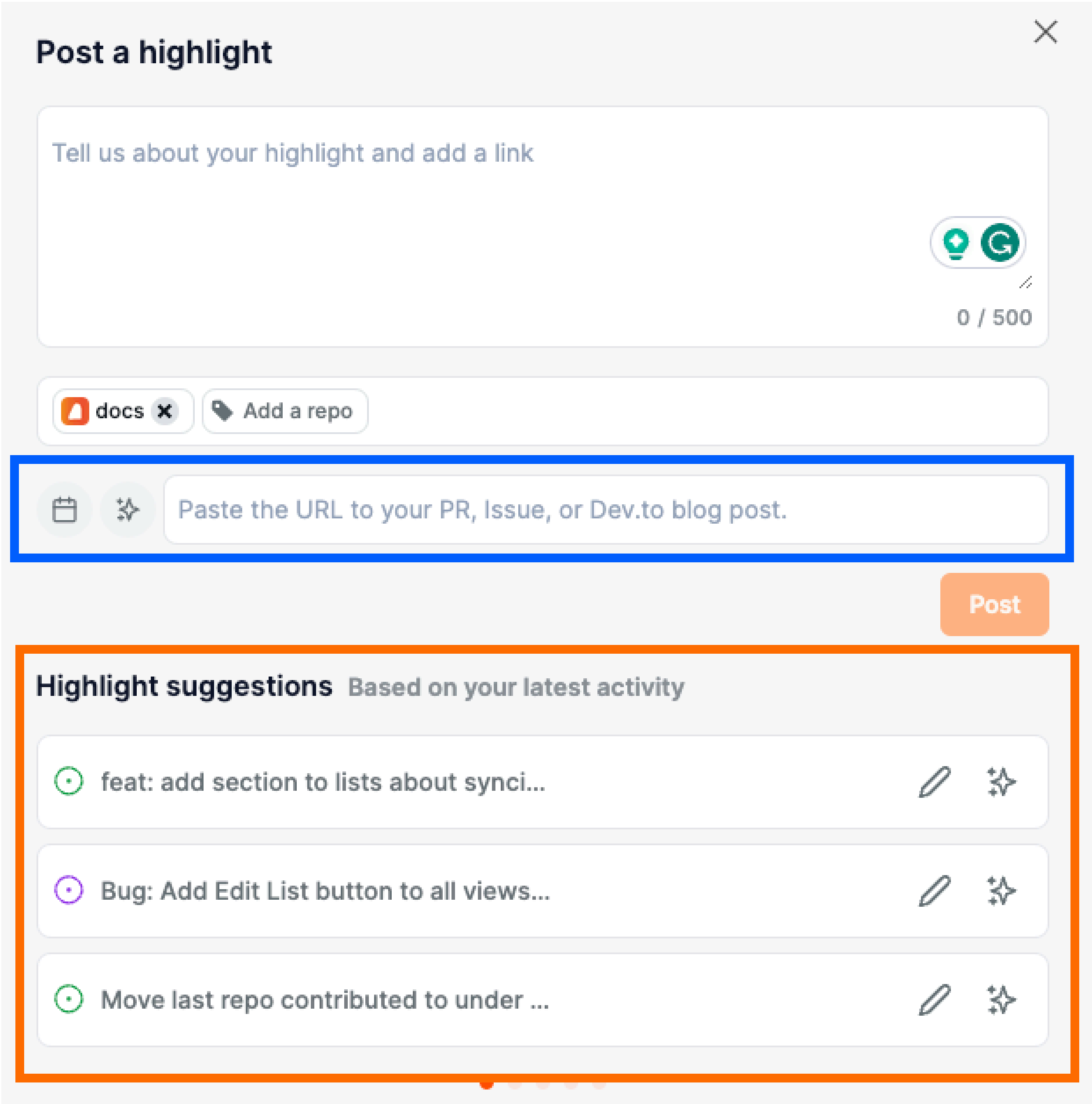
1. After clicking the Highlights tab in the top navigation, click the input that says "Post a highlight to show your work!"

2. From your profile, click the input that says "Post a highlight to show your work!"

After you've clicked the input, you'll see a form that allows you to create a new highlight.

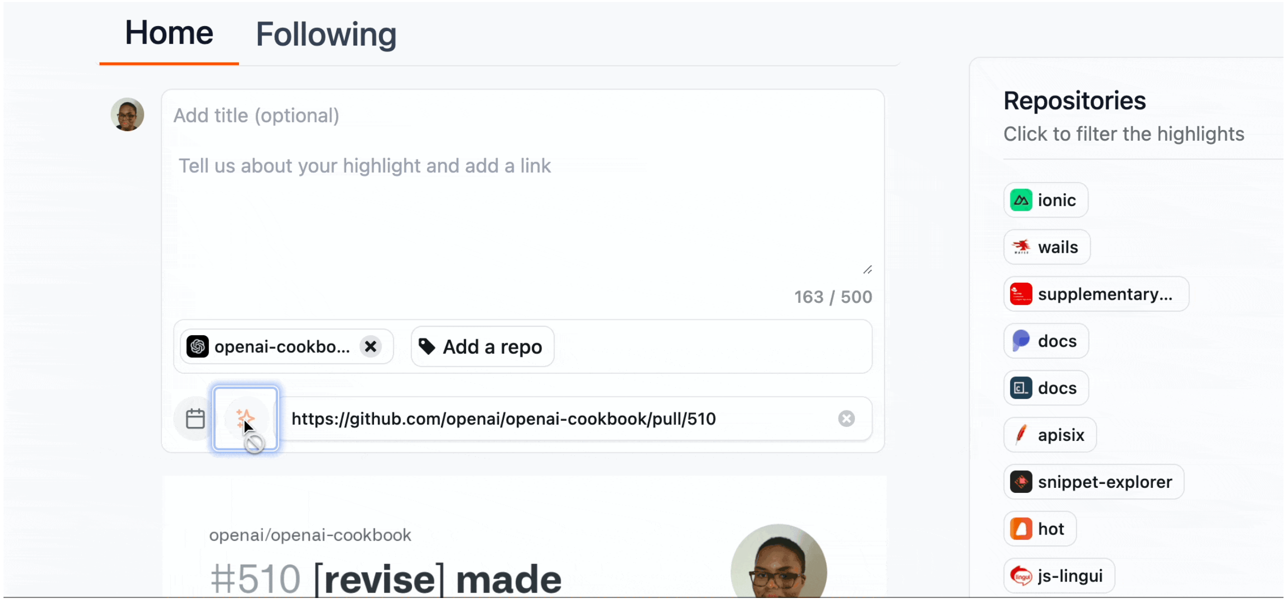
There are two ways to create a highlight:

1. Paste the URL to your PR, Issue, or Dev.to blog post (shown in blue below).
2. Choose from a list of suggestions (shown in orange below).



If you're adding a blog post, you'll need to associate it to a repository by clicking the "Add a repo" button.

Finally, it's time to add the details of your highlight. You can add a description yourself or use the auto-summarize function to generate a description.



Writing an Effective Highlight

Writing an effective highlight is key to attracting contributors to your project. Here are some tips to help you write an effective highlight:

Highlighting Issues

1. **Focus on the problem solved**: Begin by briefly describing the problem or challenge the issue addresses. This sets the context for why it was important.
2. **Acknowledge contributor(s)**: Mention the contributor(s) who reported or worked on the issue, recognizing their effort.
3. **Impact statement**: Explain how resolving this issue benefits the project, such as improving functionality, user experience, or security.

Highlighting PRs

1. **Summarize the changes**: Start with a clear summary of what the PR achieves. This could be a new feature, bug fix, or enhancement.
2. **Credit the contributors**: Acknowledge everyone who contributed to the PR, including reviewers.

3. **Highlight the value**: Explain the value of these changes to the project or its users. For instance, how it improves performance, adds a requested feature, or simplifies the codebase.

Highlighting Dev.to Blog Posts

1. **Capture the essence**: Begin with an interesting blog post summary, capturing the main theme or message.
2. **Discuss the relevance**: Explain why this blog post is important for your open source project or community. Is it about a major update, a community event, or sharing best practices?
3. **Engage with a call-to-action**: Encourage readers to engage with the post, whether by reading, commenting, or sharing their thoughts on the topic.

Understanding Insights Data

Once your repositories are connected, it's time to understand the data provided in your Insight Page. Understanding this data is key to effectively managing your projects and contributors.

Understanding Insights Repository Data

Analyzing Contribution Trends

Discover patterns in contributions over time. Look for spikes or drops in activity and correlate these with specific events or milestones. This can help you understand what drives engagement in your projects.

Monitoring Pull Request (PR) Activity

Knowing the status of PRs (open, closed, draft, merged) helps in understanding the workflow efficiency and backlog. High volumes of open or draft PRs might indicate a need for more resources or a revision of your review process. Conversely, a steady flow of merged PRs can indicate a healthy, active development process.

Measuring and Improving PR Velocity

Track how quickly PRs are being merged. The average time taken for PRs to merge is a critical metric for understanding the efficiency of the code review and deployment process. Longer PR merge times might signal issues in the review process, resource allocation, or could indicate complex code changes that require more thorough checks. Use this data to set benchmarks and identify opportunities for streamlining your review and integration processes.

Understanding Insights Contributor Data

Identifying and Engaging with Key Contributors

Recognizing key contributors helps in understanding the project's most active and influential members. Consider reaching out to them for deeper collaboration or acknowledging their efforts to keep them motivated and engaged.

Tracking and Responding to Contributor Activity Levels

Tracking how active contributors are and the trends in their activity levels is crucial for project health. Understanding the activity patterns of contributors helps in assessing their engagement and reliability. Regular contributors are often more familiar with the project and can be more dependable for critical tasks.

Notice the ebbs and flows in individual contributors’ activity. A decline in a normally active contributor’s engagement could indicate burnout or dissatisfaction, while a surge in activity might be tied to recent changes or enhancements in the project. Tracking activity levels can help you manage team burnout, understand resource needs, and spot potential issues before they escalate.

Activity Levels Across Repositories

This indicates the breadth of a contributor's involvement in open source projects. A contributor active across multiple repositories might bring diverse experiences and ideas but may also have divided attention. Alternatively, a contributor active in a single repository might be more focused and familiar with the project.

Understanding Activity Levels Over Time

Past activity levels can be a predictor of future contributions. Regular past contributions might suggest continued involvement. However, a contributor who has been inactive for a while might be less likely to contribute in the future.

Trends in activity levels and contributions can be indicators of the overall health of the project community. A decline in activity might signal issues that need addressing to keep the community vibrant and engaged.

Using Programming Language Data

Awareness of the programming languages a contributor is comfortable with enables maintainers to assign tasks more effectively, ensuring that contributors work in areas where they are most skilled and interested. This can help in identifying contributors for specific tasks or projects.

Creating Opportunities

Understanding who contributes to what open source projects and when they do can help maintainers manage team dynamics. With this information, maintainers can make decisions to ensure workloads are balanced and contributors are working in areas that match their skills and interests. Identifying less experienced contributors who are active and showing potential can lead to mentorship opportunities, helping to grow the project’s contributor base.

Understanding Lists Data

Once you've created your list of contributors, it's time to understand the data provided. Understanding this data is key to effectively managing your projects and contributors.

Understanding Lists Contributor Data

Interpreting Total Commits in the Last 30 Days

Understanding the commits over the last 30 days helps in gauging the overall activity and productivity of the group.

High commit volume might indicate a highly active and engaged team. A sudden drop could signal potential issues or a need for additional support or motivation.

Monitoring Active Contributors

Identifying active contributors helps in recognizing reliable team members and planning future tasks.

Trends in active contributor numbers can indicate the health of the project community and the effectiveness of engagement strategies. Notice the ebbs and flows in individual contributors’ activity.

A decline in a normally active contributor’s engagement could indicate burnout or dissatisfaction, while a surge in activity might be tied to recent changes or enhancements in the project. Tracking activity levels can help you manage team burnout, understand resource needs, and spot potential issues before they escalate.

Understanding Types of Contributors

New Contributors

Tracking new contributors is key to understanding the growth and influx of new talent.

An influx of new contributors might be a positive sign of project popularity, successful outreach efforts, or the community health of your project. However, it could also indicate a need for more support and guidance for new contributors.

Alumni Contributors

Alumni contributors are contributors who haven't been active in the last 30 days. Identifying contributors who have stopped contributing helps in assessing potential issues or dissatisfaction.

High churn rates could signal deeper problems within the project environment or community management.

Most Active Contributors

Identifying key contributors and potential leaders is important for project health and growth. It can help with delegation of responsibilities and recognition of top performers. It can also identify potential mentors for new contributors and help in planning for succession.

If you see that a few active contributors are doing most of the work, it might be a sign that the project needs more support or that the workload needs to be better distributed.

Activity Across Other Repositories

Understanding the range of repositories to which contributors in a list have recently contributed shows the broader interests and skills of the community. It reveals contributors' engagement levels and potential specializations, which allows for effective task allocation and identifying mentorship or leadership candidates.

Additionally, it opens paths for potential cross-project collaborations and strategic project planning, aligning the project’s goals with trends and interests in the open source ecosystem.

Understanding the recent history of each contributor's engagement assists in tracking individual contribution patterns and identifying potentially cooling interests.

Welcome to the OpenSauced Community

Welcome to the OpenSauced community! At OpenSauced, we strive to bring collaboration and inspiration to every open source contributor and help build a global community of open source developers, empowering you to grow, innovate, and achieve greatness in the open.

Community Guidelines

We pledge to create a welcoming and inclusive community for everyone at OpenSauced. We have a few guidelines to help us achieve this goal:

- Be respectful and kind to others in the community.
- Be patient with others and help them learn.
- Be open to feedback and constructive criticism.
- Adhere to the Code of Conduct.

Keeping Up with OpenSauced

The OpenSauced community is a group of open source enthusiasts who are passionate about making open source more accessible to everyone. Here are some ways you can keep up with what we're doing:

- Follow us on X @SaucedOpen for announcements and our frequent X Spaces.
- Join our Discord and hang out with us during our weekly office hours.
- Subscribe to our YouTube channel for the latest updates and video content for OpenSauced.
- 📰 Subscribe to our newsletter for all things OpenSauced and open source.
- 🗓️ Check out the calendar for our next weekly community chat.

How to Get Involved

You can get involved in the OpenSauced community in a few ways:

- Share your contributions! We love to see what you're working on. Highlight your contributions on OpenSauced.
- Open an issue or ask to be assigned to an existing issue on any of our OpenSauced repositories.
- Share what you're working on, ask questions, or mentor new contributors in our Discord.
- Provide feedback on our GitHub Discussions.

Resources on Getting Started with Open Source

Getting started with open source can be challenging, so we've put together some resources to help you get started.

- 📝 Check out our blog on Dev.to, where we provide resources for open source contributors.
- 📖 Take our Intro to Open Source Course to help you start with open

source.

- 📅 Join our #100DaysOfOSS Challenge to help you grow your skills and gain real-world experience in open source.

#100DaysOfOSS: Growing Skills and Real-World Experience

Inspired by the great work of the #100DaysOfCode challenge, we're starting #100DaysOfOSS.

With this challenge, OpenSauced hopes to help contributors enhance their skills, expand their abilities, gain practical experience over 100 days, support maintainers, onboard more contributors into open source, and expand the open source community. With a focus on open source software (OSS), we encourage contributors of all technical backgrounds to immerse themselves in the world of collaborative development and engage with a supportive community.

How to Participate

The beauty of this challenge is that you're not required to code. The main purpose is to grow in your understanding of open source software (OSS), contribute in ways that are meaningful to you, and further develop the skills and knowledge you're interested in pursuing. It's all about personal growth and making a positive impact on the OSS community. Because this challenge is focused on growth, you can participate in any way that helps you achieve your goals, including taking days off when you need it.

There are numerous ways to participate in the #100DaysOfOSS challenge, including:

Content Creation

Love creating content? Here's how you can use this skill for the challenge:

- Give a talk or presentation on OSS.
- Participate in or even start an X Space.
- Write a blog post or create a video about OSS.

Project Management

Want to improve your project management skills? Need to get better at communicating or giving feedback to others? Here's how you can use this skill for the challenge:

- Review pull requests.
- Triage issues.
- Start meaningful discussions.
- Write issues to identify bugs or suggest new features.

Community Engagement and Leadership

Community experience is involved in many different tech jobs, including support, success, Developer Relations, Developer Experience, and more. Community experience can be a valuable skill to develop. Some ways you can do this:

- Engage in the community by sharing your insights and knowledge.
- Support contributors working on OSS projects, providing guidance and support.

- Create a study group or accountability group where you work on your goals.

Technical Skills

Now is a great time to show your technical growth. Here are some paths you can take:

- Update or write documentation to improve clarity and usability.
- Maintain a project: review pull requests, triage issues, and respond to questions.
- Submit pull requests to contribute code changes.

Tracking Your Progress

To keep track of your progress, post on social media, your blog, or any other platform you prefer with:

- progress you made,
- the day of the challenge indicated by 'D',
- the hashtag #100DaysOfOSS.

For example, if you're on day one, you could say, "Today, I reviewed the documentation for the OpenSauced/App repository. D1 #100daysOfOSS." Then, on day two, you would continue with D2, and so on.

Here's what you can do if you're ready to join this challenge:

- Post your commitment today on X or share on your platform of choice!

The Official Kickoff

We started our first #100DaysOfOSS on July 23rd, 2023, and went to the end of Hacktoberfest, a month-long celebration of open source contributions.

Don't worry if you didn't start then. The #100DaysOfOSS challenge is flexible; you can join whenever you're ready. Just jump in at any point and begin with day one of your personal challenge.

Support

To make the most of your #100DaysOfOSS journey, here are some additional resources and events you can explore:

1. **Weekly X Spaces:** Join our weekly X Spaces sessions, where we discuss open source topics, share insights, and connect with like-minded individuals. Follow us on X to stay updated on upcoming sessions.

2. **Community events:** Discover a wide range of events on our community docs page. Whether hack days, workshops, or office hours, these events provide excellent opportunities to learn, collaborate, and find new projects to contribute to.

3. **Weekly contribution opportunities:** If you're actively looking for open source projects to contribute to, check out the weekly post for new contribution opportunities.

4. **Weekly office hours:** Have questions or need help? Join us on Discord during our office hours or post on our `#100DaysOfOSS` channel. We're here to help you succeed!

Where to Start?

If you're ready to start your #100DaysOfOSS journey, here are some tips to help you get started:

- **Find a project:** Explore the OpenSauced website to find a project that interests you. You can also check out the weekly post for exciting contribution opportunities.
- **Take our Intro To Open Source course:** If you're new to open source, we recommend taking our Intro to Open Source course to learn more about open source and how to get started.
- **Record your journey:** You will learn a lot as you navigate this challenge, so we highly recommend using our #100DaysOfOSS journal template on GitHub or Notion to help you reflect. Remember, this is *your* journey, so feel free to customize them however you like.

The Power of the #100DaysOfOSS Challenge

The #100DaysOfOSS challenge offers a supportive community where developers can find encouragement, share experiences, and overcome roadblocks together.

Our hope is that the community will provide a safe space to discuss challenges, celebrate achievements, and exchange insights, creating an environment that helps individuals stay on track and avoid giving up.

Why Join the #100DaysOfOSS

Challenge?

1. **Skill enhancement**: By working on real-world projects, you'll gain practical experience and exposure to different projects, documentation, communities, programming languages, frameworks, and tools. You'll also learn from experienced developers, receive feedback on your code, and improve your problem-solving abilities.

2. **Collaboration and networking**: You'll have the opportunity to work alongside other contributors, collaborate on shared goals, and build professional relationships. This experience can lead to networking opportunities, mentorship, and exposure to diverse perspectives in tech.

3. **Resume and portfolio boost**: Experience in open source demonstrates your ability to work in a team, follow best practices, and contribute to larger codebases. Open source contributions are tangible evidence of your skills, commitment, and ability to grow.

4. **Learning from peers**: By examining the codebase, participating in discussions, engaging in the community, and reviewing pull requests, contributors can gain insights into different approaches to community, projects, coding styles, architecture patterns, and software development best practices, accelerating a developer's learning curve.

5. **Making a positive impact**: Your contributions benefit other contributors who rely on these projects, fostering a sense of fulfillment and giving back to the community.

Weekly Office Hours

Have questions or need help? Join us on Discord during our office hours every

Tuesday at noon ET or post on our `#100DaysOfOSS` channel. We're here to help you succeed!

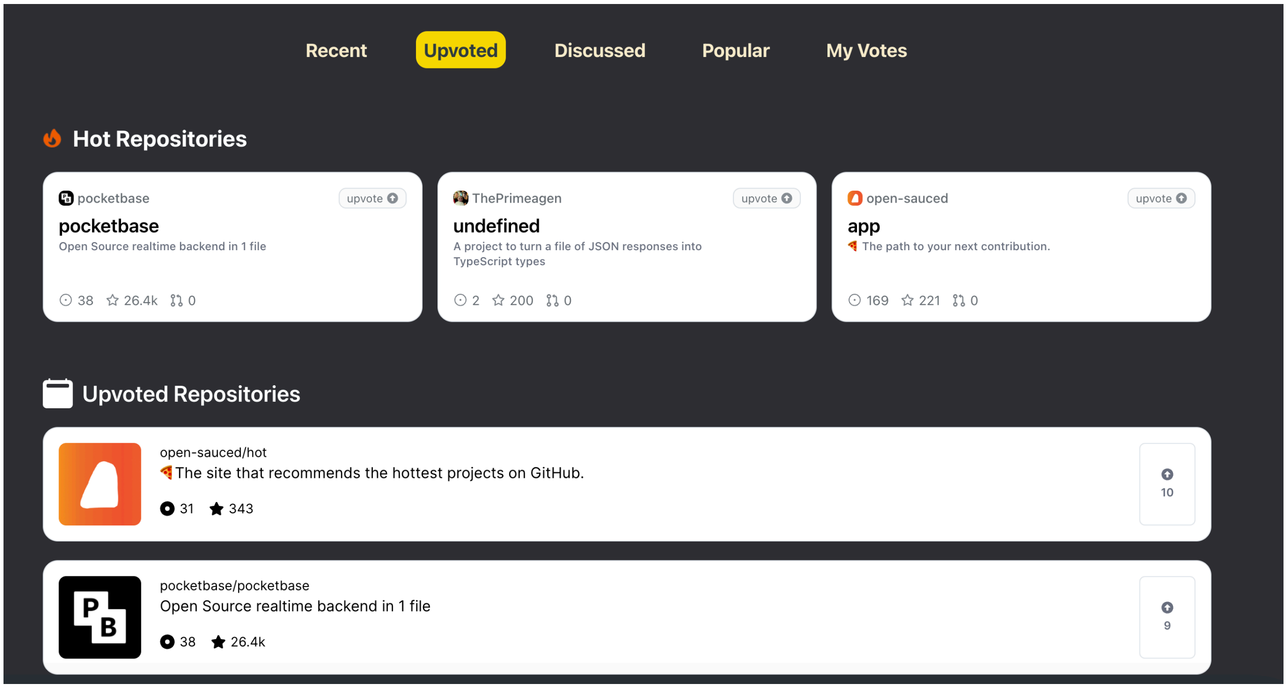
Happy contributing, and best of luck on your #100DaysOfOSS adventure!

Hot OpenSauced Project

Hot OpenSauced is a site that helps you discover the hottest open source projects to contribute to. You also have the ability to vote for your favorite open source projects.

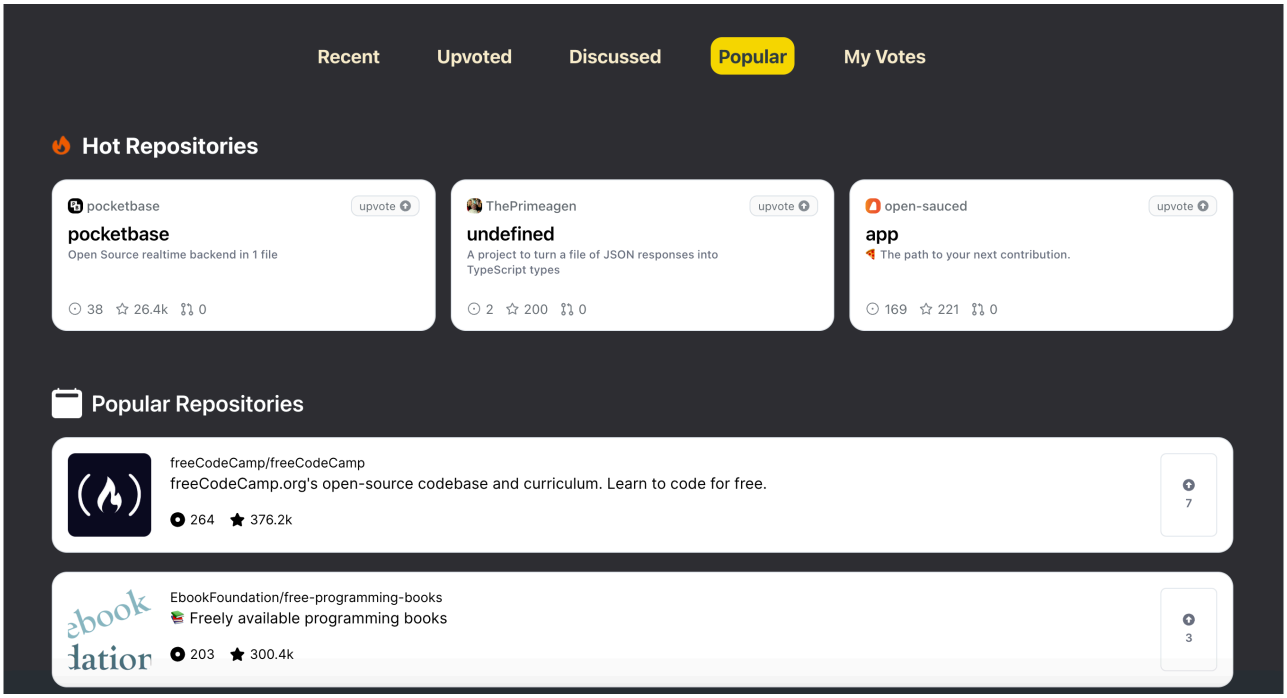
How to Find Hot Open Source Projects to Contribute to?

When you visit the Hot OpenSauced site, you will see a list of open source projects that are currently trending on GitHub. You can sign in with your GitHub account and search for the most upvoted, recent, and popular projects.



How to Vote for Your Favorite Open Source Projects?

You can vote for your favorite open source project by clicking on the upvote button. You can view all of your upvoted projects under the My Votes tab.


Recent
Upvoted
Discussed
Popular
My Votes
Hot Repositories
pocketbase
upvote
pocketbase
Open Source realtime backend in 1 file
38 26.4k 0
ThePrimeagen
upvote
undefined
A project to turn a file of JSON responses into TypeScript types
2 200 0
open-sauced
upvote
app
The path to your next contribution.
169 221 0
Popular Repositories
freeCodeCamp/freeCodeCamp
freeCodeCamp.org's open-source codebase and curriculum. Learn to code for free.
264 376.2k
7
EbookFoundation/free-programming-books
Freely available programming books
203 300.4k
3

Introduction to the Pizza CLI

The OpenSauced Pizza CLI

The OpenSauced Pizza CLI is a command-line utility designed specifically for OpenSauced. It provides a comprehensive set of functionalities that make it easier for users to gain insights and metrics and manage various tasks related to open source projects through OpenSauced.

Beyond just code commits, the pizza CLI reports on different ways individuals contribute to the open source ecosystem through community engagement, issue reporting, blog post writings, or pull request creations.

Key Features

- **Bake**: Seamlessly source git commits into OpenSauced with the metaphorical 'oven' of technology.
- **Insights**: Dive deep into data about git contributors, repositories, users, and pull requests.
- **Repo-query**: Get your questions about any GitHub repository answered.
- **Show**: Visualize the metrics of a repository, providing a holistic view of its health and activity.
- **Login**: Securely log into the application via GitHub, ensuring a streamlined and safe experience.

Installation and Instructions

📦 Download and Install

The pizza CLI provides multiple installation methods to cater to the varied preferences of its users:

- **Homebrew**: For macOS users who prefer the Homebrew package manager.

  ```
  brew install open-sauced/tap/pizza
  ```

- **NPM**: For those who are fond of the Node Package Manager.

  ```
  npm i -g pizza
  ```

- **Direct script**: A quick way to get started by directly fetching the installation script from GitHub.

  ```
  curl -fsSL https://raw.githubusercontent.com/open-sauced/pizza-cli/main/install.sh
  ```

  ⚠ **Warning**: Piping scripts directly from the web to your shell can be risky. Always ensure you trust the source. For a safer approach, download, inspect, and then execute the following commands:

  ```
  curl -fsSL https://raw.githubusercontent.com/open-sauced/pizza-
  ```

- **Manual build and install**: For those who like a hands-on approach.

  ```
  make install
  ```

  Ensure `/usr/local/bin/` is in your path with:

  ```
  export PATH="$PATH:/usr/local/bin"
  ```

  Alternatively, after building with `make build`, move the binary to your desired location:

  ```
  mv build/pizza <your-preferred-path>
  ```

🖥️ Local Development Setup

To contribute to or modify the pizza CLI, you need a specific set of tools:

- **Go Toolchain**: The foundational tool for building and managing Go applications.
- **Docker**: Required for linting and other auxiliary tooling.
- **Make**: A build automation tool.

  Development commands:

  - **Linting**: Ensure code quality with `make lint`.
  - **Testing**: Verify functionality with `make test`.
  - **Building**: Create the executable with `make build`.

Introduction to the Chrome Extension

The OpenSauced Chrome Extension

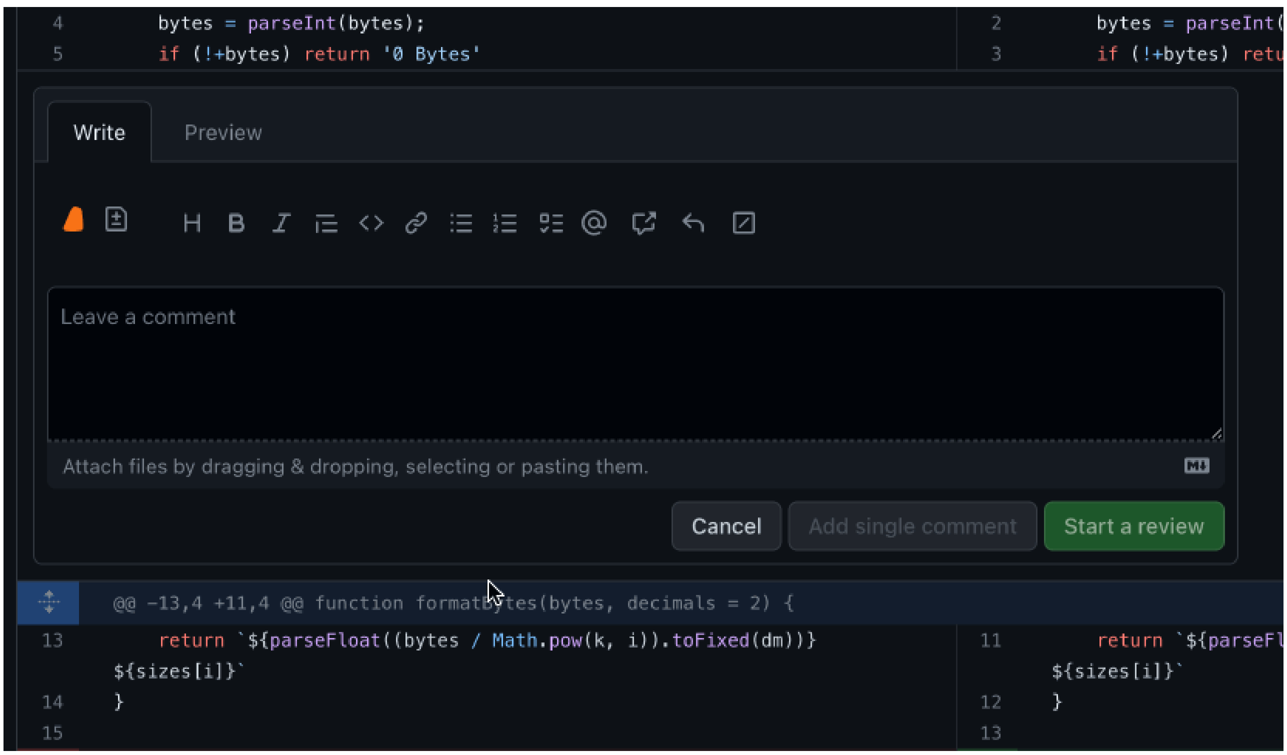
The OpenSauced Chrome extension seamlessly integrates GitHub with the OpenSauced platform. With this extension, you can easily view and discover open-source projects looking for contributions directly from GitHub, making collaboration and contribution easier than ever.

Installing the Chrome Extension

To install the OpenSauced Chrome extension, navigate to the OpenSauced.ai extension on Chrome Web Store and click the "Add to Chrome" button.

Code Explanation

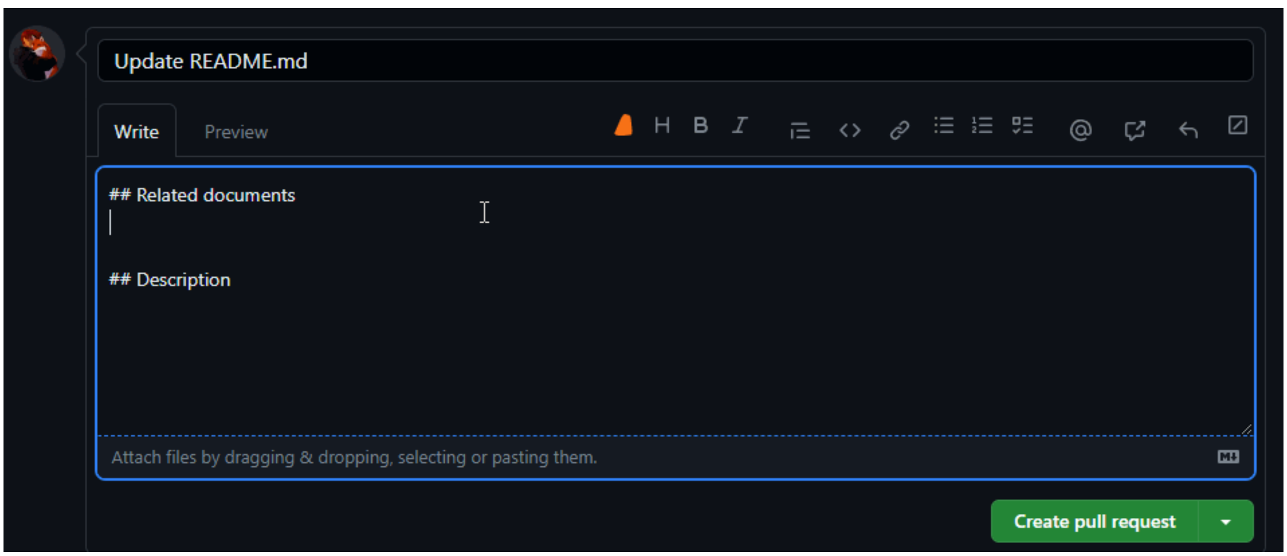
Generate an explanation for a block of code using the blue plus button that is displayed when hovering over a line. To select a block of code, hover over the starting line, click, and drag the blue plus button until you reach the desired line.



PR Description

The OpenSauced Chrome extension will help you create a PR based on the configuration you have in the extension. It will enable you to create the summary based on either commit messages, the difference between the files, or both. You can even set the message's length and tone using the same settings.

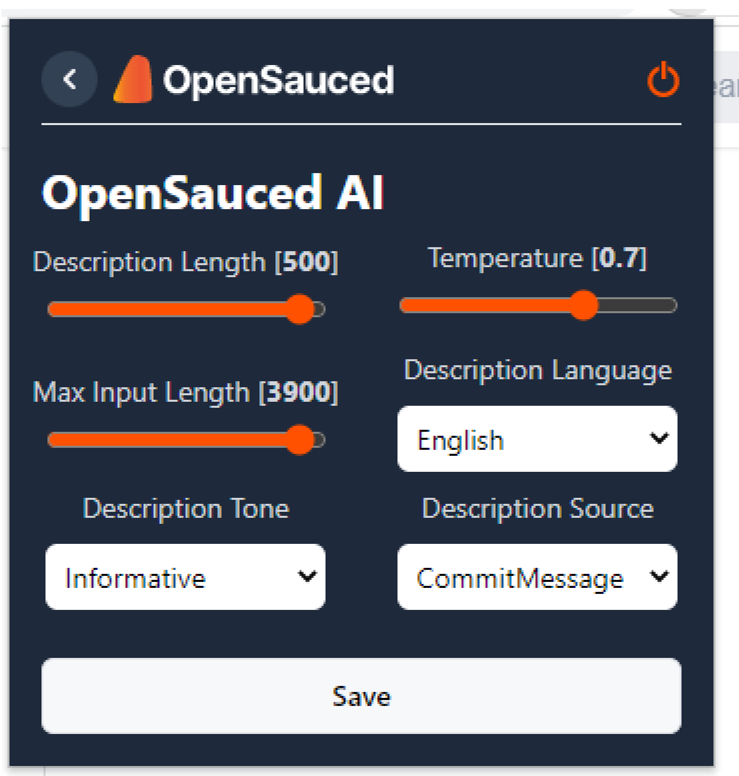
***Note**: This feature is not available on private repositories.*

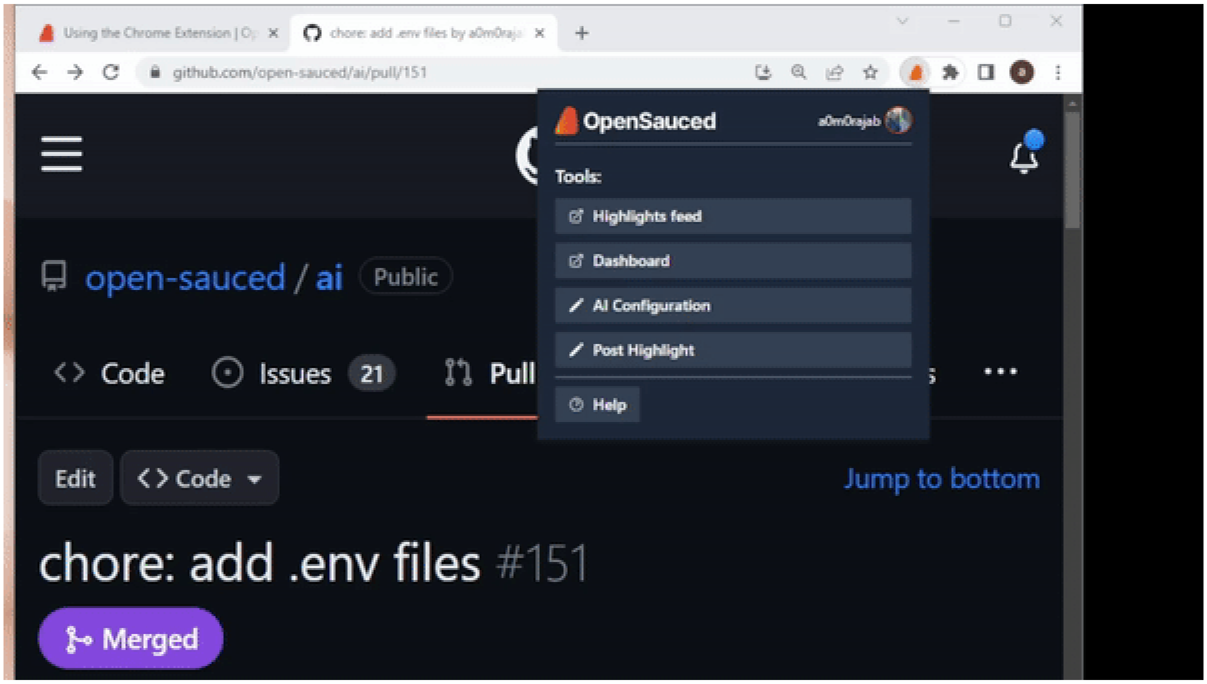


You can use these settings to set the strictness and the tone of the generated AI. Below is the explanation of the settings:

- **Description Length**: The output length.
- **Temperature**: The similarity between the input text and the output, higher temperature mean more randomness, lower temperature means more strict to the input.

- **Max Input Length.**
- **Description Language:** Right now, this feature supports: English, Spanish, French, German, Italian, Portuguese, Dutch, Russian, Chinese, Korean.
- **Description Tone:** The tone of the output: Exciting, Persuasive, Informative, Humorous, Formal.
- **Description Source:** The source of the description that our AI will use to get the output it can be the next: **Commit Messages** or **The diff between the files** or **both**.





Example PR, using the next options: both in the description source, 500 for output length and 3900 max input length, temperature 0.7.

Exciting: This PR fixes the follow call and adds a `FollowUser` function to the `contributor-highlight-card`. The code has been linted and merged from the `beta` branch. `FollowUser` function allows users to follow/unfollow contributors. Copy link and follow/unfollow options are now available in the dropdown menu.

Persuasive: This PR fixes the follow call and linting issues in the `ContributorHighlightCard` component. It also merges the `beta` branch into `fix-follow-check`. A new `FollowUser` function has been added to handle follow/unfollow functionality. This PR ensures a smoother user experience.

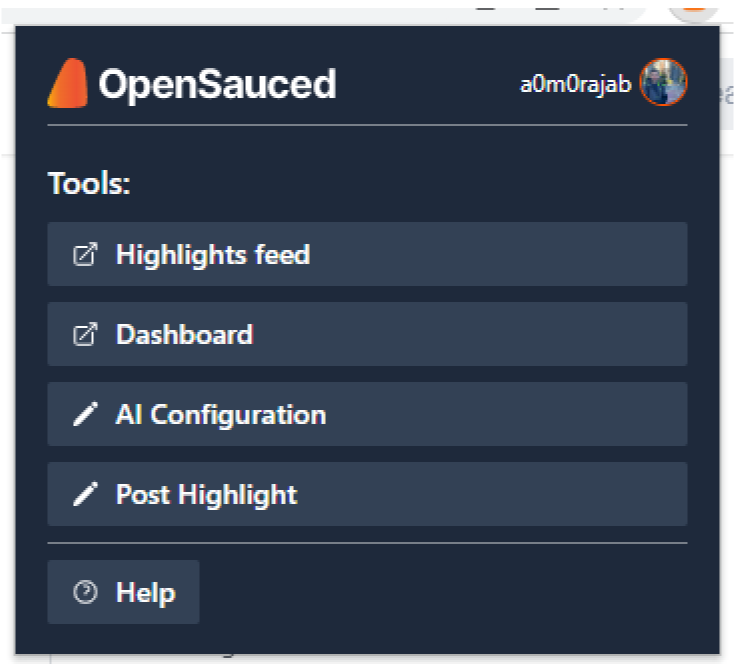
Informative: This PR fixes the follow call and linting issues in the `ContributorHighlightCard` component. A new function `FollowUser` has been added to handle the follow/unfollow functionality. The `useFollowUser` hook has been moved inside this function. The `FollowUser` function is now used in place of the previous follow/unfollow code block. The code has been tested and reviewed.

Humorous: This PR fixes the follow call and also lints the file. We've also added a new `FollowUser` function to make the code more readable. Now you can follow/unfollow contributors with ease. We've also merged the `beta` branch into `fix-follow-check`.

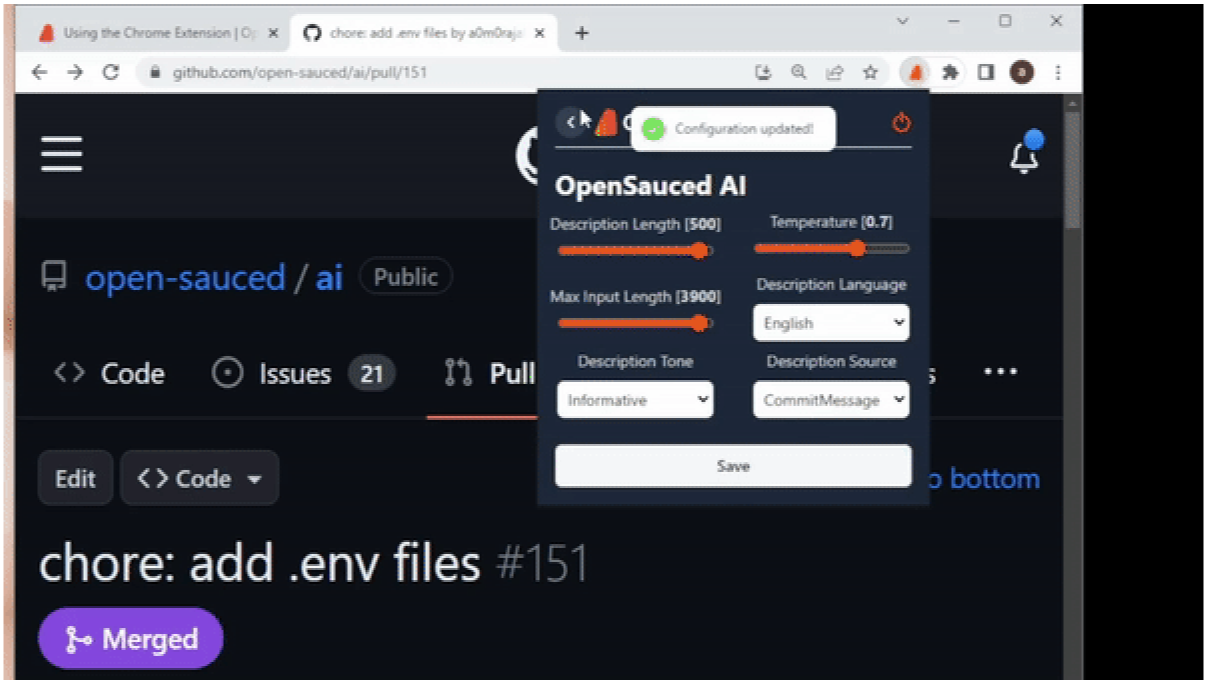
Formal: This PR fixes the follow call and linting issues in the `ContributorHighlightCard` component. A new function `FollowUser` has been added to handle the follow/unfollow functionality. The `useFollowUser` hook has been moved inside this function. The `FollowUser` function is now used in place of the previous follow/unfollow code block. This PR also merges the `beta` branch into `fix-follow-check`.

Highlight

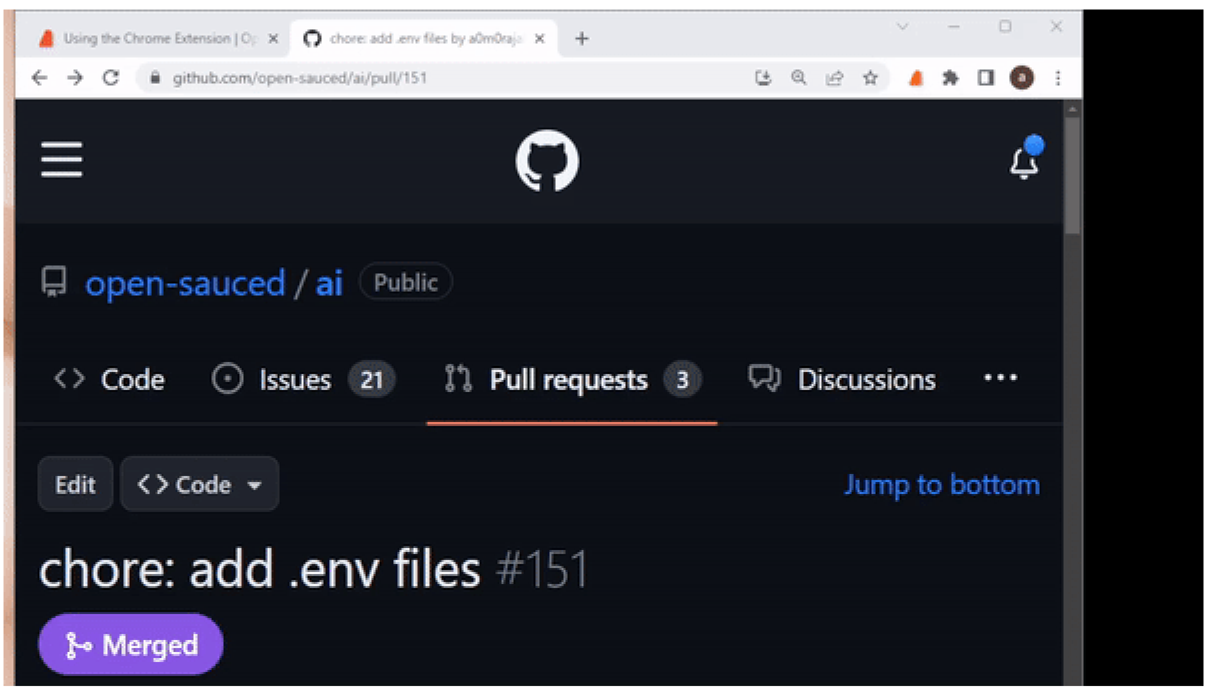
You can access the post to highlight it from the popup window in the extension. When you clicked, it will automatically, populate the PR or issue title, here you can use our AI functions to get a summary of the highlight similar to the PR summary.



If your Highlight is successfully posted, you will get a message with an option to see the Highlight on the OpenSauced website.



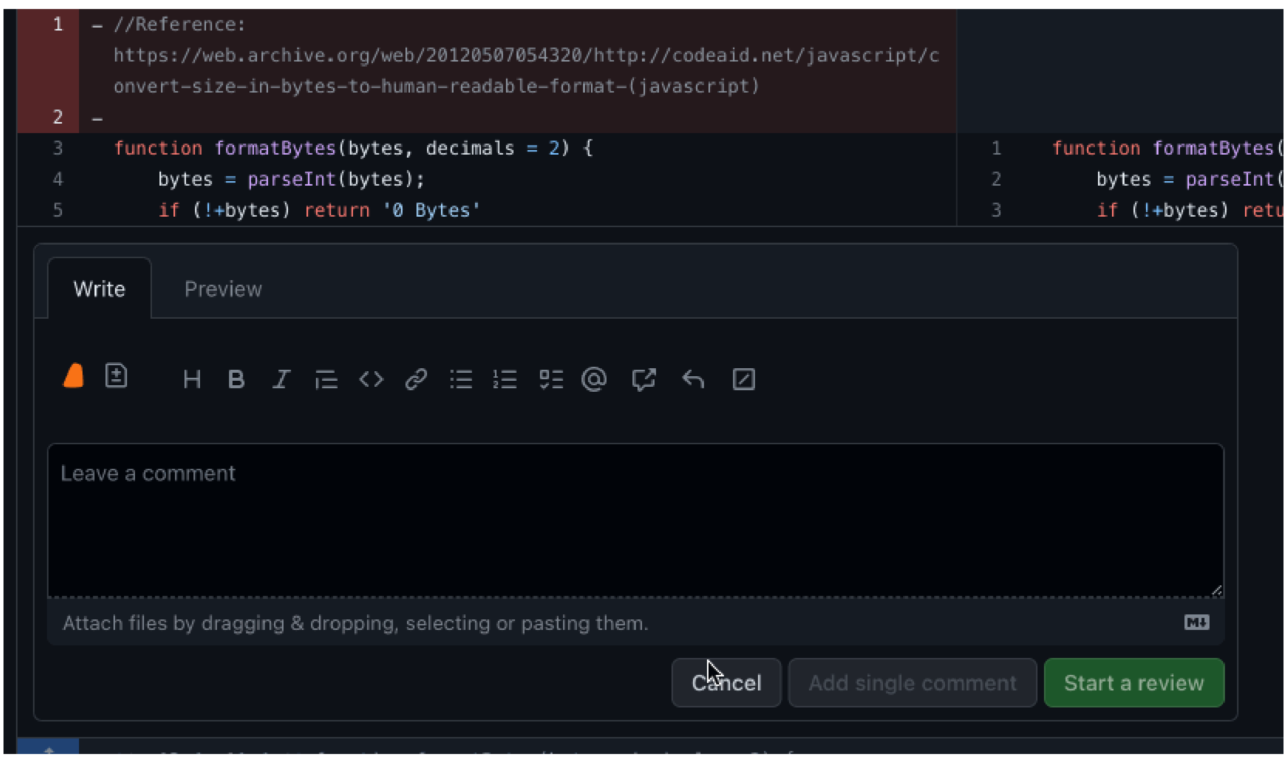
But if there is an error with the summarization, you will see it in the text area, as in the example below:



Refactoring and Testing

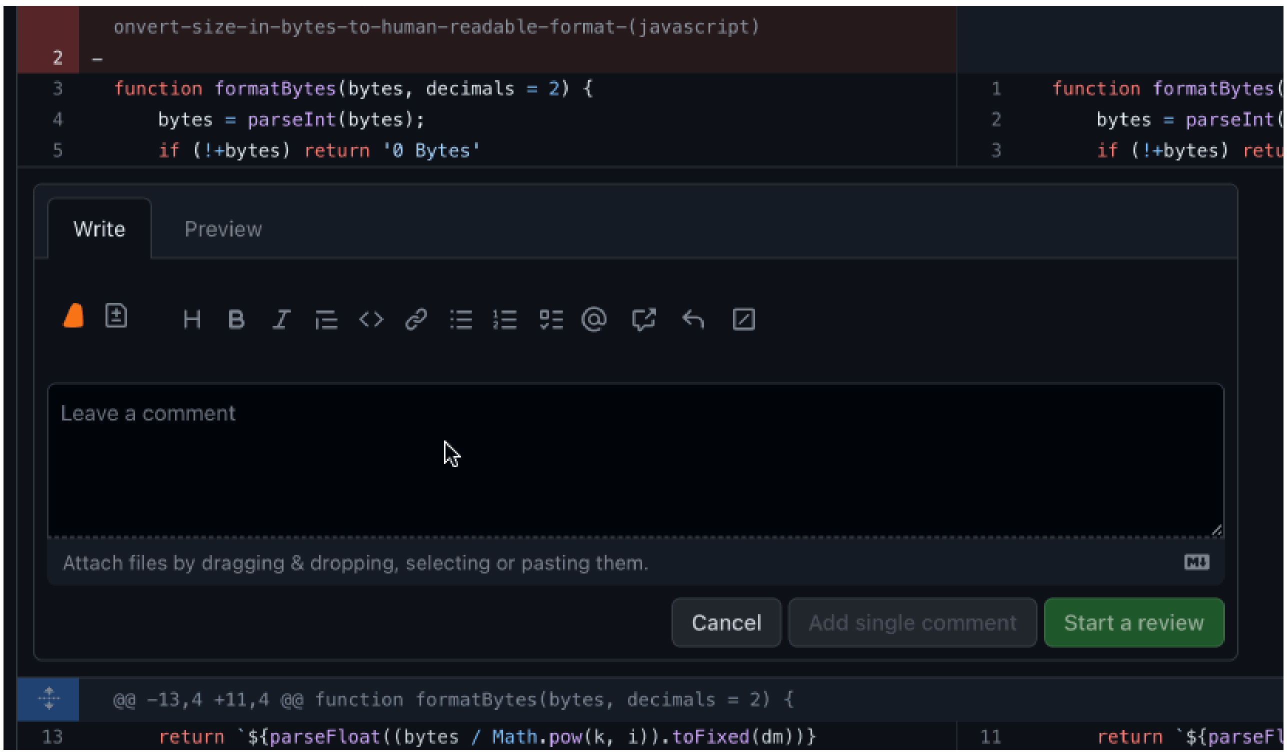
Refactoring a Block of Code

Refactor a block of code by using the blue plus button that is displayed when hovering over a line. To select a block of code, hover over the starting line, click, and drag the blue plus button until you reach the desired line.



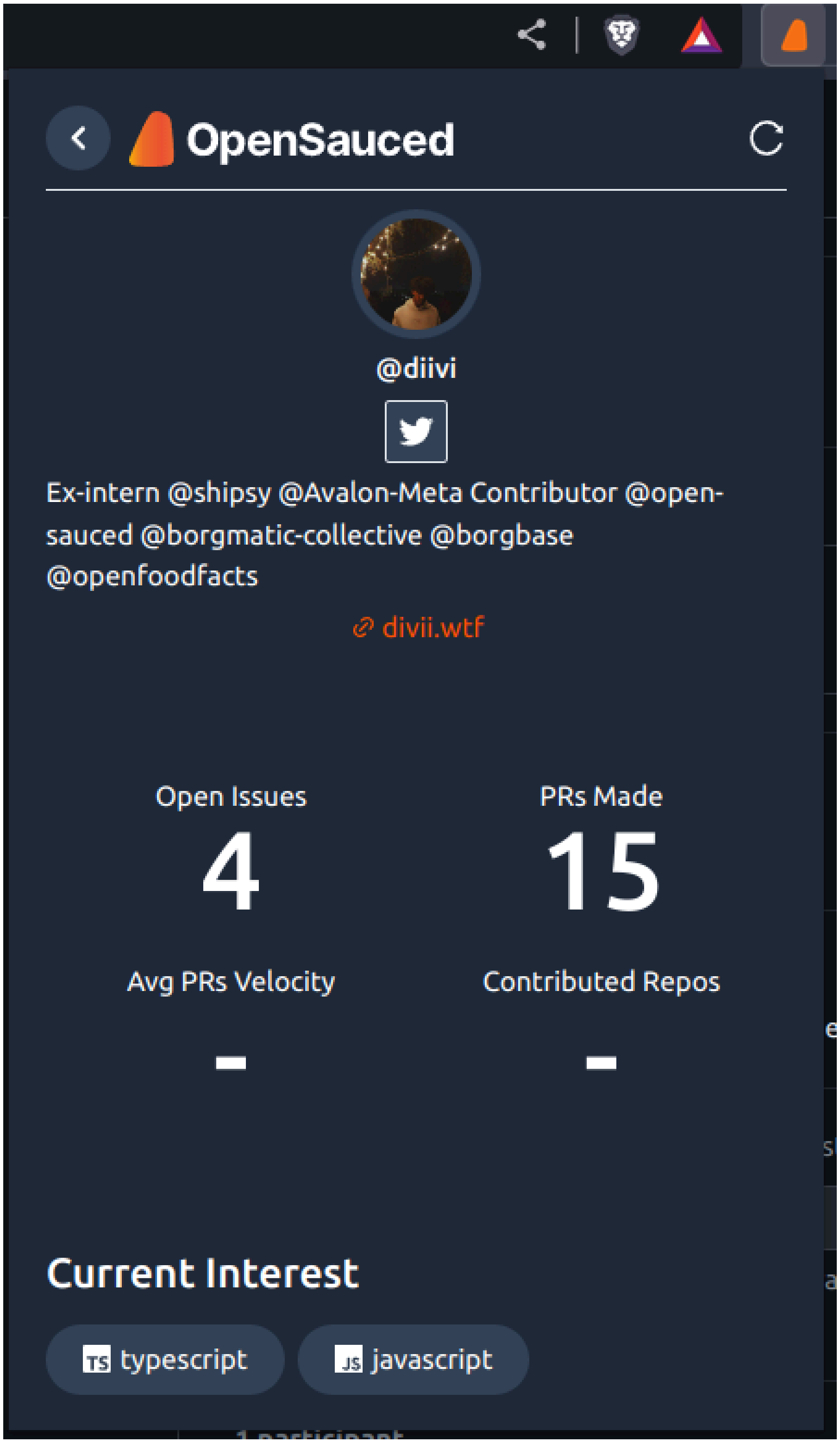
Testing a Block of Code

Generate tests for a block of code by using the blue plus button that is displayed when hovering over a line. To select a block of code, hover over the starting line, click, and drag the blue plus button until you reach the desired line.



Viewing Insights and Invitations

View stats about open issues, PRs made, average PRs velocity, and contributed repos that are in the OpenSauced database. This page can be found by clicking the profile picture at the top left of the extension.


OpenSauced
@diivi
Ex-intern @shipsy @Avalon-Meta Contributor @open-sauced @borgmatic-collective @borgbase @openfoodfacts
divii.wtf
Open Issues
4
PRs Made
15
Avg PRs Velocity
-
Contributed Repos
-
Current Interest
typescript
javascript

Invite GitHub Users to Join OpenSauced With a Single Click

Invite other users to create an OpenSauced account to keep track of open source contributions when visiting their GitHub profile.

View GitHub Users' OpenSauced Profiles and Connect With Them

View a user’s OpenSauced profile when on their GitHub profile page.

Introduction to Contributing

Contributions are always welcome, no matter how large or small. Before contributing, please read the code of conduct and follow the directions below:

Recommended Communication Style

1. Always leave screenshots for visual changes.
2. Always leave a detailed description in the pull request. Leave nothing ambiguous for the reviewer.
3. Always review your code first. Run the project locally and test it before requesting a review.
4. Always communicate in the GitHub repository. Whether in the issue or the pull request, keeping the lines of communication open and visible to everyone on the team helps everyone around you.

Issues

If you wish to work on an open issue, please comment on the issue with `.take`, and it will be assigned to you. If an issue is not assigned, it is assumed to be available for anyone to work on. Please assign yourself to an issue before working on it to avoid conflicts.

If you contribute to the project for the first time, you can search for issues with `good first issue` or `bug` labels in the repository.

In case you get stuck, feel free to ask for help in the Discord server or GitHub Discussions.

Please follow our code of conduct in all your interactions with the project and its contributors.

Commits

When contributing to OpenSauced projects, use this command to follow our conventional commits:

```
npm run push
```

The `npm run push` command is an interactive replacement for `git commit`. It enforces the conventional commit specification for writing commit messages, making it easier for developers and maintainers to understand the changes made in a particular commit.

After running this command, you can push your changes to the remote repo.

See Conventional Commits for more examples.

Pull Requests (PR)

We actively welcome your pull requests. However, you must link your work to an existing issue.

1. Fork the repo and create your branch from the default branch.
2. Name your branch to something descriptive of your work, i.e., `add-navbar` or `fix/broken-links`.
3. If you've added code that should be tested, add tests.
4. If you've changed APIs, update the documentation.
5. If you make visual changes, screenshots are required.
6. Ensure the test suite passes.
7. Make sure you address any lint warnings.
8. If you improve the existing code, please let us know in your PR description.
9. Completing the PR form is required. Make sure to fill in the PR title, description, link to an issue, and all the necessary areas.
   - The title must begin with "feat:", "fix:", or anything related to your changes.
     Tips: You can follow your chosen option when committing your changes.
   - Unsolicited code is welcomed, but an issue is required to announce your intentions.
     PRs without a linked issue will be marked invalid and closed.

> ***Note for maintainers****: All PRs need a label to assist automation. See this guide to determine which labels to use.*

PR Validation

Examples of valid PR titles:

- fix: Correct typo.
- feat: Add support for Node 12.
- refactor!: Drop support for Node 6.

***Note**: For any breaking changes, you have to use the `!` syntax.*

Work in Progress

GitHub supports draft pull requests, which will disable the merge button until the PR is marked as ready for merge.

Additional Resources

- *3 tips for getting your Pull Request reviewed on GitHub*

Getting Started

Setting Up Projects Locally

1. Fork the repository you're working on from github/open-sauced to your own GitHub account.
2. Clone the forked repository to your local machine.
3. Check the project's README to determine how to run the project locally.

Additional Information

For many of our repositories, you'll run `npm ci` to install the dependencies and set up the project locally.

From the npm official docs:

> The `npm ci` is a command that installs a project with a clean slate, skipping certain user-oriented features and being more strict than `npm install`.

You can also use the shell commands below to get started once you have forked the repository:

```
git clone https://github.com/<GITHUB_USERNAME>/<REPO_NAME>.git
cd <REPO_NAME>
npm ci
```

Make sure to replace `<GITHUB_USERNAME>` with your GitHub username.

Building

To generate a production-ready version of your code, run:

```
npm run build
```

Testing

For running the test suite, use the following command:

```
# the tests will run in watch mode by default
npm test
```

Since the tests run in watch mode by default, some users may encounter errors about too many files being open. In this case, it may be beneficial to install Watchman.

Check out this course, Testing JavaScript, for more info on testing React and JavaScript.

Applying Lint Styleguide

To check the code for formatting and linting errors, run the following command:

```
npm run lint
```

These errors will also be displayed during development but won't prevent the code from compiling.

To fix the formatting and linting errors, run the following command instead:

```
npm run format
```

These commands use ESLint to check and fix the code.

If you forget to run this command, automated PR checks will also run these commands, but the PR will be blocked if there are any errors.

Coding Tips

- Ask questions if you are stuck.
- Use CSS variables.
- Always use rel="noreferrer" on all target="_blank" links.

Community

Do you have questions? Join the conversation in our Discord.

Triage Team

The Triage team is inspired by expressjs/express. This team exists to create a path for making contributions to this project and open source. All Triage Team members are expected to follow the Triage Guide.

> **There are no minimum requirements to become a member of the Triage Team.**

For those interested in getting involved in the project or just open source in general and wanting to join the Triage Team, please leave a comment in the `🍕opensauced-contributors-chat` channel on Discord.

Funding

OpenSauced is a part of GitHub Sponsors. If you would like to contribute, please note the sponsor page for details on how funds are distributed. If you have made any contributions to the project directly or indirectly, please consider adding your profile to the `FUNDING.yml` file.

License

By contributing to the OpenSauced project, you agree that your contributions will be licensed by a specific License. You can find this information in the `LICENSE` file of the repo you are contributing to.

Code of Conduct

Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to make participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language.
- Being respectful of differing viewpoints and experiences.
- Gracefully accepting constructive criticism.
- Focusing on what is best for the community.
- Showing empathy towards other community members.

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances.
- Trolling, insulting/derogatory comments, and personal or political attacks.

- Public or private harassment.
- Publishing others' confidential information, such as a physical or electronic address, without explicit permission.
- Other conduct that could reasonably be considered inappropriate in a professional setting.

Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned with this Code of Conduct or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include:

- Using an official project e-mail address.
- Posting via an official social media account.
- Acting as an appointed representative at an online or offline event.

Representation of a project may be further defined and clarified by project maintainers.

Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at hello@briandouglas.me. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

Attribution

This Code of Conduct is adapted from the Contributor Covenant version 1.4, available at https://contributor-covenant.org/version/1/4.

Triage Guide

How to Join the Triage Team

1. Sign up for opensauced.pizza.
2. Leave a comment in the `🍕opensauced-contributors-chat` channel on Discord.

Triage Process

When a new issue or pull request (PR) is opened, the issue will be labeled with `needs triage`. If a triage team member is available, they can help ensure all the required information is provided.

Adding Issue and PR Labels

Depending on the issue or PR, there are several labels the triage team can add for further classification, as follows:

- `needs triage`: This can be kept if the triager is unsure which next steps to take.
- `awaiting more info`: If more information has been requested from the author, apply this label.
- `question`: User questions that do not appear to be bugs or enhancements.
- `discuss`: Topics for discussion. It might end in an `enhancement` or `question` label.
- `bug`: Issues that present reasonable conviction there is a reproducible bug.

- `enhancement`: Issues that are found to be a reasonable candidate feature additions.
- `style`: Minor CSS or visual changes.

In all cases, maintainers may close issues if they don't receive a timely response when further information is sought or when additional questions are asked.

Labeling `good first issue`

Issues labeled as `good first issue` represent a curated list of easy contributions for new contributors. These issues are meant to help folks make their first contribution to open source and should not require excessive research or triaging on the contributor's part.

All `good first issue` should include one or more of the following:

- A solution.
- A suggestion for a solution.
- Links to components or in which issue occurs.

Please be noted:

- Issues that `needs triage` cannot be labeled as `good first issues`.
- It is better to have no `good first issue` labeled issues than to have a `good first issue` confusing enough to deter a contributor from contributing.

Approaches and Best Practices for Getting Into Triage Contributions

- Review the project's Contribution Guidelines if they are present. In a nutshell, commit to the community's standards and values.

- Review the documentation; for most projects, it is just the `README.md`. Ensure you understand the key APIs, semantics, configurations, and use cases.

- Writing your own test apps to reaffirm your understanding of the key functions might be helpful. This may identify some gaps in documentation; record those as they might be good PRs to open.

- Skim through the issues backlog; identify low-hanging and mostly new issues. From those, attempt to recreate issues based on the OP (Original Poster) description and ask questions if required. No question is a bad question!

Removal of Triage Role

There are a few cases where members can be removed from the triage role:

- Breaking the Code of Conduct or project contributor guidelines.
- Abuse or misuse of the role as deemed by the Triage Team.
- Lack of participation for more than six months.

If any of these happen, we will discuss them as a part of the triage portion of the regular Triage Team meetings. If you have questions, feel free to contact any Triage Team members.

Helpful Hints

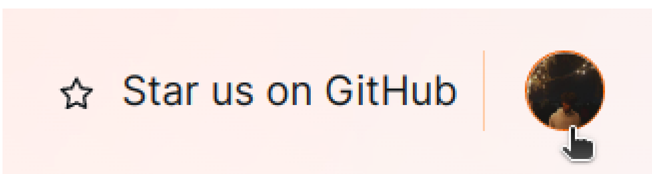
- When reviewing the list of open issues, there are some common types and suggested actions:
    - **New/unattended issues or simple questions**: A good place to start.
    - **Hard bugs & ongoing discussions**: Feel free to chime in and help.
    - **Issues that imply gaps in the documentation**: Open PRs with changes or help the user to do so.
- For recurring issues, it is helpful to create functional examples to demonstrate (publish as gists or a repo).
- Review and identify the maintainers. If necessary, mention one or more of them with `@` if you are unsure what to do.
- Ensure all your interactions are professional, welcoming, and respectful to the parties involved.

Set Up Authentication

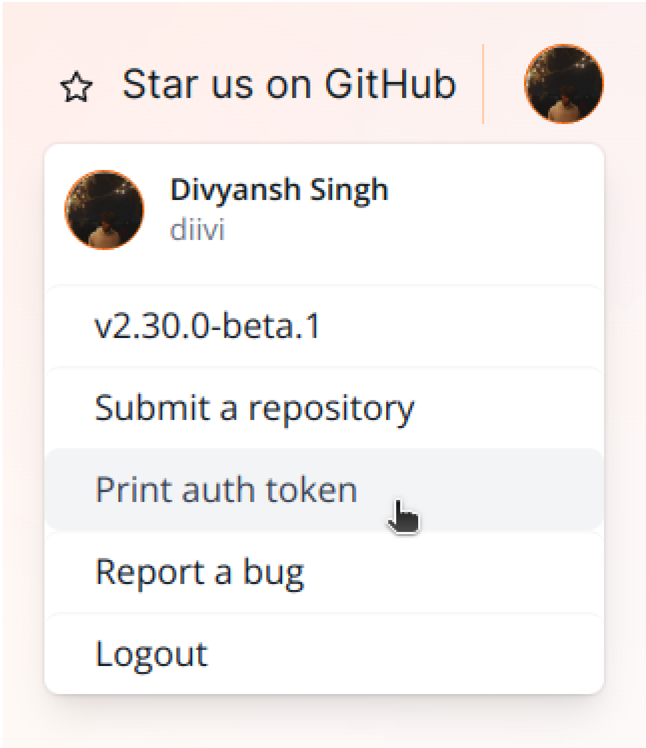
You need to obtain an authentication token to interact with the OpenSauced public API as an authenticated user.

The following steps outline how to obtain an authentication token from the hot.opensauced.pizza website:

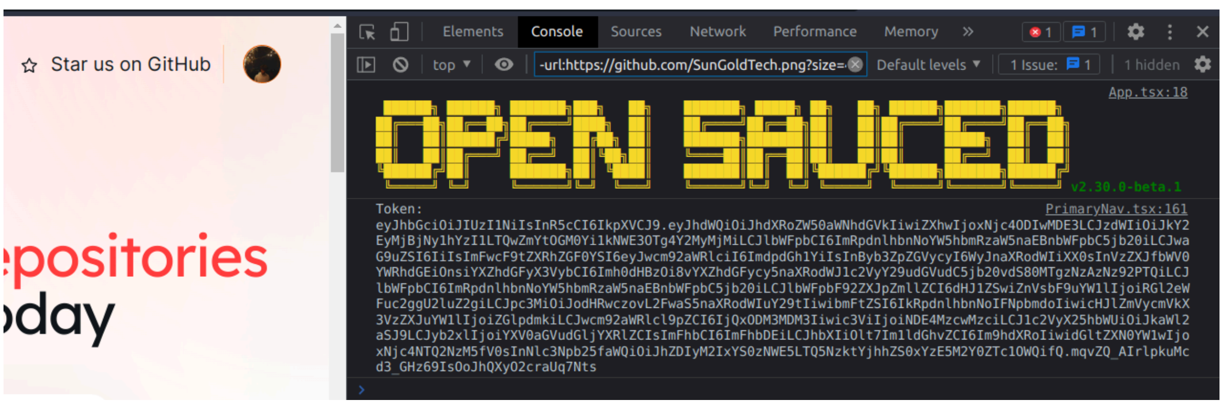
1. Click on your avatar in the top right corner of the page. This will open a dropdown menu.



2. Click on the **Print auth token** option. Don't worry, this is a safe operation. The token is only printed to the console.



3. Copy the token that is printed to the console.



4. You can now use this token to make authenticated requests to the OpenSauced public API by including it in the Authorization header of your requests. For example:

```
const response = await fetch(
```

OpenSauced Maintainers Guide

Welcome to the OpenSauced Maintainers Guide for contributors who are interested in being part of the maintainer team for OpenSauced repositories.

At OpenSauced, we empower contributors to work in the open and support maintainers to make data-driven decisions for their projects.

It's important to us that we maintain a healthy environment for contributors and maintainers of our projects.

How to Join the Maintainers Team

- Sign up for opensauced.pizza.
- Join our Discord community.
- Prove your pizza's worth!

What does the Maintainers Team do?

Well, like all things in tech, it depends. We have different maintainer teams for each of the repositories. So, it's important that you communicate with the admin maintainer if you have any questions.

Core Responsibilities

Your responsibilities may include:

- Reviewing and merging pull requests (PRs).
- Providing comments and responses on PRs, issues, discussions, and Discord messages.
- Triage issues and bug reports.
- Maintaining project documentation.
- Collaborating with the community.

Creating a positive space for contributors at all stages is one of your most important responsibilities.

As a maintainer, you must:

- adhere to our Code of Conduct and be an example for treating contributors with respect,
- communicate with the other maintainers in a timely and reasonable manner,
- understand and adhere to project standards.

Committing and Merging Changes

1. For commit and pull request standards, please refer to the introduction to contributing.
2. Before merging in changes, always run the project locally if UI changes were made.
3. **Squash and merge commits** when you merge in a PR.

Thank you for being so interested in becoming a maintainer! Please reach out in our team discussions if you need help, guidance, or clarification at any time.

Setting Up a Repository with Git and GitHub

Using the GitHub CLI

How to Install the GitHub CLI

The GitHub CLI allows you to fork repositories, create issues, pull requests, and more from the command line.

Follow these instructions to install GitHub CLI on Mac, Windows, or Linux.

How to Authenticate with the GitHub CLI

From the terminal, you will need to authenticate with the GitHub CLI. You can do this by running the following command:

```
gh auth login
```

Then, follow the prompts to authenticate with the GitHub CLI.

How to Fork and Clone a Repository with the GitHub CLI

A fork is a remote copy of a repository, allowing you to experiment freely with changes without affecting the original project.

A clone is a local copy of a repository that includes all the files, branches and commits.

To fork and clone a repository with the GitHub CLI, run the following command:

```
gh repo fork open-sauced/<REPO_NAME>
```

The GitHub CLI will fork the project in your GitHub account and will ask you if you want to clone the repository on your local machine.

How to Add a Remote Repository

Adding a remote repository allows you to pull in changes from the original repository and keep your forked copy of the repository up to date.

To add a remote repository, run the following command:

```
git remote add upstream https://github.com/open-sauced/app.git
```

How to View the Remote Repositories Locally

To view the remote repositories that your local repository is connected to, run the following command:

```bash
git remote -v
```

You should see the following output:

```
origin git@github.com:YOUR_GITHUB_USERNAME/app.git (fetch)
origin git@github.com:YOUR_GITHUB_USERNAME/app.git (push)
upstream git@github.com:open-sauced/app.git (fetch)
upstream git@github.com:open-sauced/app.git (push)
```

Using the GitHub Website and the Command Line

If you prefer to set up your repository using the GitHub website and the command line, follow this detailed guide from the official GitHub documentation.

Resolve Merge Conflicts

When you are working on any of the OpenSauced repositories, you might run into a merge conflict. A merge conflict occurs when multiple conflicting changes are made to the same line in a file. Merge conflicts happen the most when you open a pull request, as the release process generally updates `npm-shrinkwrap.json`.

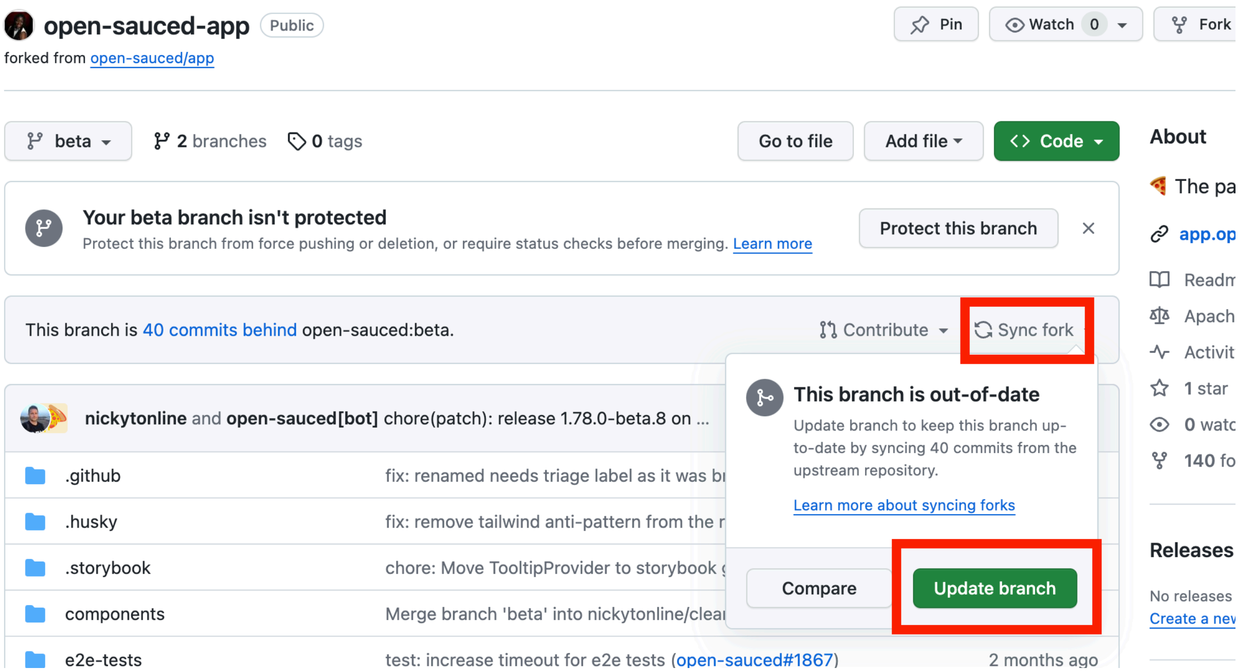
In this guide, we will talk about how to resolve merge conflicts and how to keep your branch up to date.

How to Keep Your Branch Updated

It is common for your branch to fall behind the main repository's branch. So it is important to keep it up to date as you are contributing.

Using GitHub

To update your branch on GitHub, you can go to your forked copy of the project and click on `Sync fork` and then the `Update branch` button.



Using Git

To update your branch using Git and the terminal, you can use the following commands:

1. Change directories to the correct project.

```
cd project-name
```

2. Add the `upstream` remote.

```
git remote add upstream https://github.com/upstream-username/upstream-repository.git
```

3. Pull the latest changes from the `upstream`.

```
git pull upstream main-branch-name
```

Note: Some of the OpenSauced repositories will use `main` for the main branch while others like the app repository, will use `beta` for the main branch name.

How to Check for Merge Conflicts Locally

If you are actively working on a change to an OpenSauced repository, you can check for potential merge conflicts by running a few commands in the terminal.

1. Make sure you are on the correct branch where the changes are being made.

```
cd app
git checkout <your-branch>
```

2. Fetch the latest changes from the OpenSauced main repository.

```
git fetch upstream
```

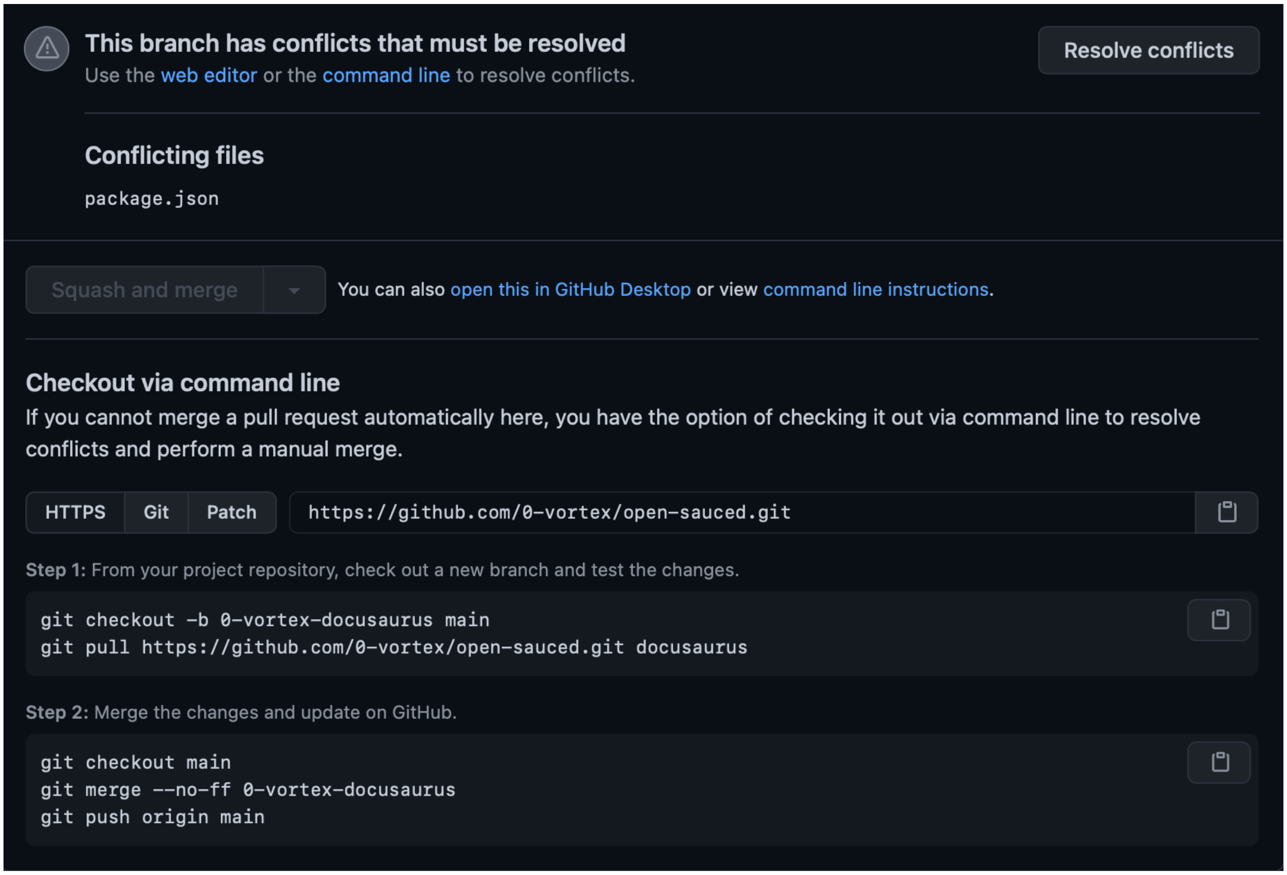
3. See the differences between your branch and the OpenSauced main branch.

```
git diff <your-branch> upstream/main
```

Note: Some of the OpenSauced repositories will use `main` for the main branch while others, like the app repository, will use `beta` for the main branch name.

```
01:28:55 in open-sauced on ᚠ (git)-[docusaurus|merge]- [=$+?] via ⬢ v16.3.0
➜ git diff package.json
diff --cc package.json
index 3337138,574378c..0000000
--- a/package.json
+++ b/package.json
@@@ -85,9 -85,9 +85,15 @@@
      "build": "node scripts/build.js",
      "clean": "rimraf src/tests/__snapshots__/",
      "test": "npm run clean && node scripts/test.js --env=jsdom --updateSnapshot",
++<<<<<<< HEAD
 +    "storybook": "start-storybook -p 6006 --no-dll",
 +    "build-storybook": "build-storybook --no-dll",
 +    "lint": "eslint --ext .js,.jsx .",
++=======
+     "storybook": "start-storybook -p 6006 -s ./.storybook",
+     "build-storybook": "build-storybook -s ./.storybook",
+     "lint": "eslint .",
++>>>>>>> origin/main
      "lint:fix": "eslint --fix --ext .js,.jsx .",
      "release": "standard-version --git-tag-fallback"
    },
```

NOTE: If you have already opened up a pull request, then you can see if there is a conflict at the bottom of the PR. Even though it is possible to resolve conflicts through GitHub, it is best practice to resolve them locally.



How to Resolve Merge Conflicts

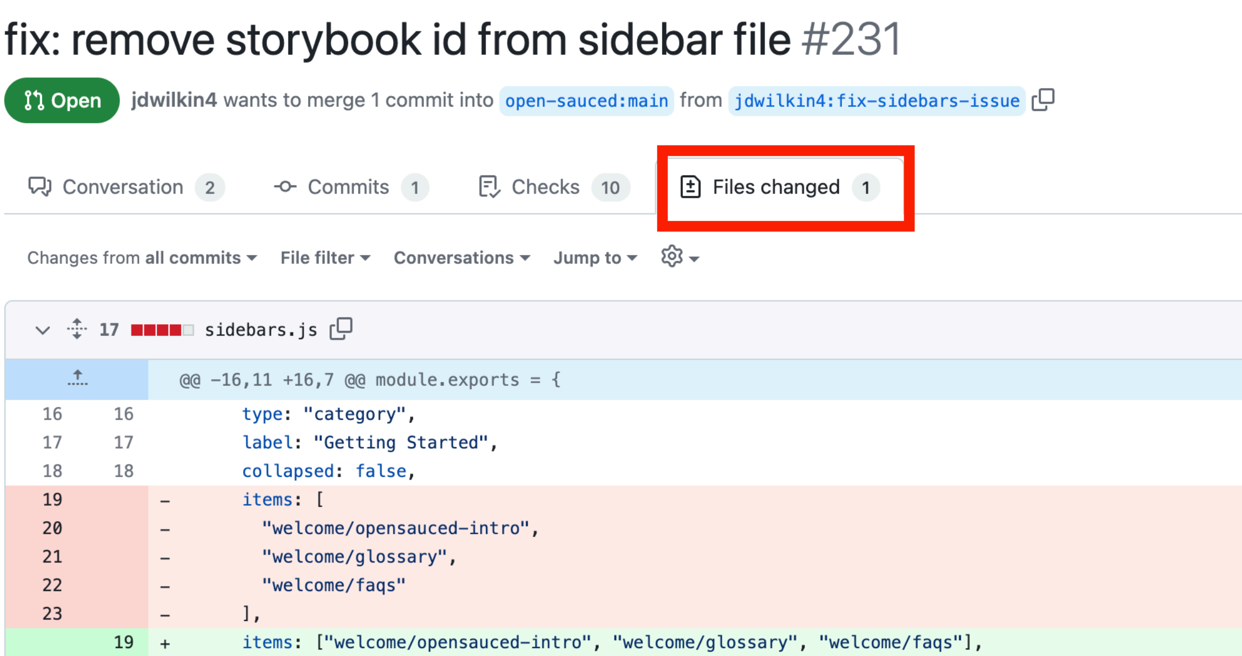
If you have merge conflicts, it is best to use a text editor to resolve them. Identify the conflicting files and open them up in your editor of choice. For the conflicting sections, you have an option to either keep the incoming changes, keep your changes or keep both sets of changes. The conflicting sections will be marked with `<<<<<<<`, `=======`, and `>>>>>>>` symbols.

Here is a guide for an in depth walkthrough of the process. Once you have resolved all of the conflicts, then you can stage, commit and push your changes to your branch.

```
git status
git add .
git commit -m "fix: resolving merge conflicts"
git push
```

Reviewing Your Pull Request

Once you have pushed up your changes, you can review them in your pull request. Under the files changed tab, you will see all of the recent changes including all conflicts resolved.



Dependency Updates

When dealing with dependency and lock file updates, there are multiple use cases to consider; however, as a baseline, the OpenSauced triage team will not prioritize parallel main features as seen in the roadmap.

However when that happens, it is advised to:

- fast-forward `npm-shrinkwrap.json`
- fast-forward deleted and modified `upstream/beta` changes to `package.json`
- fast-forward your added lines to `package.json`
- run `npm ci` to delete local modules and create dependency resolution from `upstream/beta`

Visual diffing is advised; however, not following the git commit history procedure will result in a rogue pull request that creeps into dependency updates.

Generally speaking, just adding things to a lock file will not be troublesome, and since this is a licensed project, we should be careful when adding dependencies.

@open-sauced/check-engines

Description

The `npm` package `@open-sauced/check-engines` is designed to help contributors install dependencies conforming to the `engines` property in `package.json`.

Dependencies

This package uses the following modules:

- npm-install-checks

Installation

```
npm install --save-dev @open-sauced/check-engines
```

Add the verification scripts to your `scripts` section in the `package.json` file:

```
{
  "scripts": {
    "preinstall": "npx @open-sauced/check-engines"
```

The reason why we provide `npx` in the `scripts` section is for the people using this as a development enhancement, interactive configurations or trimmed dependency trees, where using `npx` is preferred over installing all the dependencies at once.

Usage

Use your favorite package manager to install dependencies in your project or, if you set it as a global verification system:

```
{
  "scripts": {
    "check-engines": "npx @open-sauced/check-engines"
    "preinstall": "npm run check-engines",
    "prestart": "npm run check-engines"
  }
}
```

Advanced Usage

If you have an API or any other non-library type of application, you can decouple this package from any install scripts and just use it as a verification:

```
{
  "scripts": {
    "check-engines": "npx @open-sauced/check-engines"
    "prestart": "npm run check-engines"
  }
}
```

A more traditional approach not using `pre` or `post` scripts, this example enables the check only for local machine development:

```
{
  "scripts": {
    "check-engines": "npx @open-sauced/check-engines"
    "start": "...",
    "dev": "npm run check-engines && npm start -- --watch"
  }
}
```

FAQs

Usage on an Older `node` and `npm` Versions

Older `node` and `npm` versions won't be able to run this package; depending on the versions, the scripts section could be ignored completely.

If you have that use case, this package is only worth enabling for progressive contributors frequently missing the legacy support of the respective module and forcefully upgrading dependencies - them running newer versions will force the error message and explicitly disable.

Why Not Use `check-engines` or `engine-strict`?

As described in the `npm@6` engine-strict docs:

> Prior to npm 3.0.0, this feature was used to treat this package as if the user had set `engine-strict`. It is no longer used.

In `npm@6` and later, this was re-introduced as a config flag.

This package is designed with multiple legacy use cases in mind.

Library Usage

If you use this module in a library package, be advised that any `*install` script will run in the parent module when installed.

For example, given a module `@demo-org/demo-package` with a `preinstall: "npx @open-sauced/check-engines"` script, running `npm install @demo-org/demo-package` will require your locally installed `node` and `npm` versions to match the `engines` section of your `package.json`. If that is not set, nothing should happen, and this package is a stray dependency in either `@demo-org/demo-package` or the module you are running this command in.

Contributing

We're always happy to onboard people into open source!

Check out the repository at @open-sauced/check-engines. ❤

@open-sauced/conventional-commit

Description

The `npm` package `@open-sauced/conventional-commit` is designed to help users `git commit` using commitizen and conventional commits.

Dependencies

This package uses the following modules:

- cz-cli
- cz-conventional-changelog

Installation

```
npm install --save-dev @open-sauced/conventional-commit
```

Add the verification scripts to your `scripts` section in the `package.json` file:

```
{
  "scripts": {
    "push": "npx @open-sauced/conventional-commit"
```

The reason why we provide `npx` in the `scripts` section is for the people using this as a development enhancement, interactive configurations, or trimmed dependency trees, where using `npx` is preferred over installing all the dependencies at once.

Usage

All you have to do is run the script next to your `package.json`:

```
npx @open-sauced/conventional-commit
# or
npx conventional-commit
```

Advanced Usage

The most common use case for this package is to run it instead of the `git commit` command inside your `npm` scripts:

```
{
  "scripts": {
    "push": "npx @open-sauced/conventional-commit"
  }
}
```

or

```
{
  "scripts": {
    "push": "npx conventional-commit"
```

If you want to ensure local-only usage:

```json
{
  "scripts": {
    "push": "conventional-commit"
  }
}
```

FAQs

Contributing

We're always happy to onboard people into open source!

Check out the repository at @open-sauced/conventional-commit. ❤

@open-sauced/semantic-release-conventional-config

Description

The `npm` package `@open-sauced/semantic-release-conventional-config` is designed to help `npm` packages auto-release to `npm` or `ghcr` registries while generating GitHub releases and changelog using conventional commit convention.

Version 2 supports alpha and beta pre-releases using corresponding branches.

Dependencies

This package uses the following modules:

- `@semantic-release/commit-analyzer`
- `@semantic-release/release-notes-generator`
- `@conventional-changelog/conventional-changelog`
- `@semantic-release/changelog`
- `@semantic-release/npm`
- `@google/semantic-release-replace-plugin`
- `@cbhq/semantic-release-license`

- `@semantic-release/git`
- `@semantic-release/github`
- `@eclass/semantic-release-docker`
- `@semantic-release/exec`
- `execa`

🖥️ Requirements

Most important limitations are:

- `GITHUB_TOKEN` for everything
- `NPM_TOKEN` for public `npm` library
- `docker` containers need to be built beforehand

You can skip here if you are using an elevated Private Access Token. However, we don't recommend going down that path.

No force push or admin cherries branch protections for the following branches:

- `main` - required
- `alpha` - optional, pre-release branch
- `beta` - optional, pre-release branch
- `next` - optional, next channel
- `next-major` - optional, next major
- `vX[.X.X]` - maintenance releases

If you use more than the `main` branch, optionally create an environment that limits where pushes can come from and enable the merge strategy.

We use `production` in our examples; if you copy-paste them, you will find this new environment generated in your settings! 🍕

🧪 GitHub Actions Usage

Since version 3, it has been possible to use `semantic-release` without any trace of it or the OpenSauced configuration anywhere in the dependency tree.

Docker containers are pushed as part of the release, so they mirror the availability of `npm` packages.

The simplest use case for a typical `npm` package is almost zero install downtime from `ghcr` and no more local tooling:

```yaml
name: "Release container"

on:
  push:
    branches:
      - main

jobs:
  release:
    environment:
      name: production
      url: https://github.com/${{ github.repository }}/releases/tag/${{ env.RELEASE_TAG }}
    runs-on: ubuntu-latest
    steps:
      - name: "☁️ checkout repository"
        uses: actions/checkout@v2
        with:
          fetch-depth: 0
```

Marketplace actions should default to the major tag and are essentially more stable as we have to curate every release.

A more traditional approach, the only thing really different here is a minor pull overhead and using set outputs instead of environment variables:

```yaml
name: "Release"

on:
  push:
    branches:
      - main

jobs:
  release:
    environment:
      name: production
      url: https://github.com/${{ github.repository }}/releases/tag/${{ steps.semantic-release.outputs.release-tag }}
    name: Semantic release
    runs-on: ubuntu-latest
    steps:
      - name: "☁️ checkout repository"
        uses: actions/checkout@v2
        with:
          fetch-depth: 0

      - name: "🚀 release"
        id: semantic-release
        uses: open-sauced/semantic-release-conventional-config@v3
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
          NPM_TOKEN: ${{ secrets.NPM_TOKEN }}

      - name: "♻️ cleanup"
        run: |
```

📦 `npm` Usage

You can opt to use this package in your local tooling. Proceed as you would normally would, replacing `npm` with your package manager of choice and install the package:

```
npm install --save-dev @open-sauced/semantic-release-conventional-config
```

The shareable config can then be configured in the `semantic-release` configuration file:

```
{
  "extends": "@open-sauced/semantic-release-conventional-config"
}
```

Now, all you need to do is create a release:

```
npx semantic-release
```

🔧 Configuration

See each plugin official documentation for the required installation and configuration steps.

`npm`

Set `private` to true in `package.json` if you want to disable `npm` or change the scope of package using `publishConfig`.

Keep one of the `files` or `main` keys in your `package.json` accurate, depending on whether you are building a library or an application.

If you publish, make sure to also provide a valid `NPM_TOKEN`, as `.npmrc` authentication is ignored in our config!

GitHub Actions

Unless you have an `action.yml` present in your root folder, this module is not added to the release config.

If you have an `action.yml` present, our config will attempt to adjust the container version to the newly pushed `npm` and `docker` tags.

Docker

Unless you have a `Dockerfile` present in your root folder, this module is not added to the release config.

If you have a `Dockerfile` present, our config will attempt to push to `ghcr.io`.

Environment Variables

Using our configuration comes with some sensible defaults:

- `DOCKER_USERNAME=$GITHUB_REPOSITORY_OWNER`

- `DOCKER_PASSWORD=$GITHUB_TOKEN`
- `GIT_COMMITTER_NAME="open-sauced[bot]"`
- `GIT_COMMITTER_EMAIL="63161813+open-sauced[bot]@users.noreply.github.com"`
- `GIT_AUTHOR_NAME` - parsed from commit `$GITHUB_SHA`
- `GIT_AUTHOR_EMAIL` - parsed from commit `$GITHUB_SHA`

Feel free to change any of the above to whatever suits your purpose. Our motivation is to keep `GITHUB_TOKEN` and/or `NPM_TOKEN` the only necessary requirements.

We are actively investigating ways to drop the two remaining variables as well!

Workflow Examples

Node Application

This example requires `"private": true,` in your `package.json` and simplifies the workflow to lightning-fast deployment:

```
release:
  environment:
    name: production
    url: https://github.com/${{ github.repository }}/releases/
tag/${{ env.RELEASE_TAG }}
  name: Semantic release
  runs-on: ubuntu-latest
  steps:
    - name: "☁ checkout repository"
      uses: actions/checkout@v2
      with:
```

`npm` Library

For `npm` libraries, we need to set the environment URL manually and set a `NPM_TOKEN` environment variable. This also disables `docker builds`:

```yaml
name: "Release"

on:
  push:
    branches:
      - main

jobs:
  release:
    environment:
      name: npm
      url: https://www.npmjs.com/package/@open-sauced/semantic-release-conventional-config/v/${{ env.RELEASE_VERSION }}
    name: Semantic release
    runs-on: ubuntu-latest
    steps:
      - name: "☁️ checkout repository"
        uses: actions/checkout@v2
        with:
          fetch-depth: 0

      - name: "🔧 setup node"
        uses: actions/setup-node@v2.1.5
        with:
          node-version: 16

      - name: "🔧 install npm@latest"
        run: npm i -g npm@latest
```

An up-to-date version of the example above is available at @open-sauced/semantic-release-conventional-config.

Docker Image

For `docker builds`, it's best to build your node application in parallel with the container and re-use the artifact at a later stage:

```yaml
name: "Release"

on:
  push:
    branches:
      - main

jobs:
  docker:
    name: Build container
    runs-on: ubuntu-latest
    steps:
      - name: "☁ checkout repository"
        uses: actions/checkout@v2

      - name: "🔧 setup buildx"
        uses: docker/setup-buildx-action@v1

      - name: "🔧 cache docker layers"
        uses: actions/cache@v2
        with:
          path: /tmp/.buildx-cache
          key: ${{ runner.os }}-buildx-${{ github.sha }}
          restore-keys: |
            ${{ runner.os }}-buildx-

      - name: "🔧 docker meta"
```

An up-to-date version of the example above is available at open-sauced/open-sauced.

Pre-Releases

This workflow requires the creation of `alpha` and `beta` protected branches while templating every commit to be conventional. It does not support squashing without creating extremely complex conflict resolution:

```
name: "Release"

on:
  push:
    branches:
      - main
      - beta
      - alpha

jobs:
  release:
    environment:
      name: npm
      url: https://www.npmjs.com/package/
open-sauced-semantic-config-test/v/${{
steps.release.outputs.version }}
    name: Semantic release
    runs-on: ubuntu-latest
    steps:
      - name: "☁️ checkout repository"
        uses: actions/checkout@v2
        with:
          fetch-depth: 0

      - name: "🚀 release"
        id: semantic-release
```

FAQs

Which Assets are Pushed to Git?

The following assets are added to git using `@semantic-release/git`:

```
{
  "assets": [
    "LICENSE",
    "LICENSE.md",
    "COPYING",
    "COPYING.md",
    "CHANGELOG.md",
    "package.json",
    "package-lock.json",
    "npm-shrinkwrap.json",
    "public/diagram.svg",
    "action.yml"
  ]
}
```

What is the Commit Convention?

The following commit rules are enforced by `@semantic-release/commit-analyzer`:

```
{
  "preset": "conventionalcommits",
  "releaseRules": [
    { "type": "build", "release": "minor" },
```

How to Enrich the Static Distribution?

The following assets are packed into the github release download using `@semantic-release/github`:

```json
{
  "assets": [
    {
      "path": "pack/*.tgz",
      "label": "Static distribution"
    }
  ]
}
```

How to Start Using Pre-Releases?

Create the `alpha` and/or `beta` branches and protect them from being deleted or pushed directly by non-administrators.

Switch your branching strategy to `merge` and enable conventional commits checking.

As described in the semantic-releases recipes, you will have to resolve merge conflicts between `alpha`, `beta`, and `main` branches.

Contributing

We're always happy to onboard people into open source!

Check out the repository at @open-sauced/semantic-release-conventional-config. ❤

Learn How to Land a Tech Job Using OpenSauced

When you are looking for a tech job, it is important to stand out amongst the sea of candidates. One great way to stand out and showcase your skills is through Open Source contributions.

Open Source provides you the opportunity to showcase your code, documentation, communication and collaboration skills to potential employers. You also have the ability to grow your professional network and build real connections within the tech industry through open source.

OpenSauced can help you craft an open source profile that is attractive to hiring managers and recruiters and help you land that next role.

In this guide, you will learn how to use OpenSauced features like your personal Dev Card, Highlights, Lists and Insights to showcase your open source work and build connections to help you land a job.

How to Translate Open Source Contributions into Career Opportunities

It is important to keep track of your open source contributions and show the impact you have left on the projects and organizations. By organizing your work, you will be able to talk about these meaningful contributions on your resume and in job interviews.

In this guide, we have created a table that you can use to categorize and keep detailed notes for each contribution you have made. We will break down each of these table components and talk about how to use this in your resume and job profile, and you can see a full example at the end of this guide.

Personal Information

In this section of the table, you can include information like your name, email, GitHub profile, LinkedIn profile, OpenSauced profile, blog, etc. When it comes to the professional profiles, think about which profiles you want to show off to employers that highlights your skills and experience relevant to the types of jobs you are interested in.

Section	Details
Personal Information	
Name	
Contact Information	
LinkedIn	
GitHub profile	
OpenSauced profile	

Career Goals

If you are new to the industry, you will want to list out which area of tech you are interested in pursuing. If you are looking for your next tech role, then you can list your current role or the role you wish to switch to.

If you are looking to level up in your role, you should list it in this section and list out the key skills required for that next level. You should also link to a few job posts to get an idea of key skills are required for that next level you are working towards.

Section	Details
Career Goals	

Section	Details
Desired Job Role	[Your Desired Role]
Desired Job level	[Junior, Mid, Senior, etc]
Key Skills to Develop	[Skills to Focus On]
Industry/Area of Interest	[Frontend, Backend, Fullstack, ML, AI, etc.]
Example Job Links	[Link to desired jobs]

How to Match Job Requirements to Your Current Skills

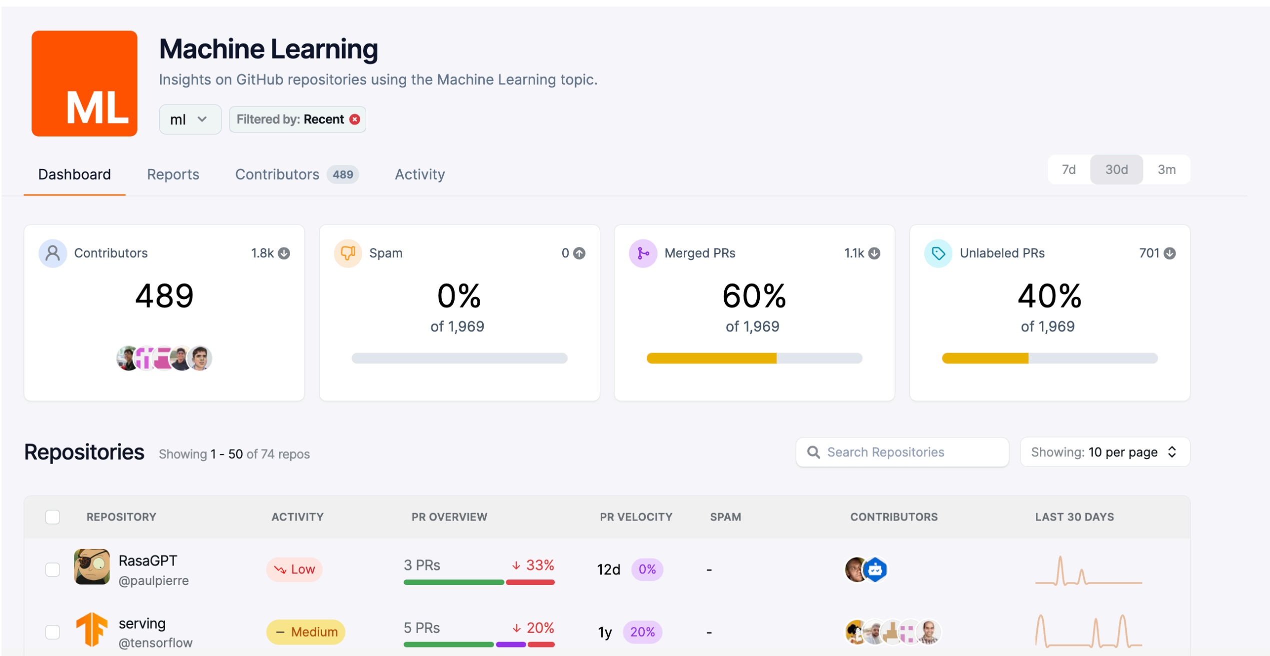
A common mistake for many job seekers is to send out the same resume template to all jobs. But in today's tech market it is important to stand out and tailor your resume to match your skills with the requirements listed.

In the previous section, you identified desired roles and key skills needed for those jobs. When you are reading through job descriptions, you will want to identify key skills that are repeatedly mentioned and add that to your career goals section.

As you fill out the remaining sections for project contributions, check to see if the contributions match the common skills requested in those jobs. If not, then you have just identified a skill gap and will need to find projects and issues that will allow you to strengthen those skills.

Finding Open Source Projects to Contribute to

After you have filled out the career section of the table, you will want to find open source projects that align with your goals. You can use the explore feature on OpenSauced to explore new repositories in a variety of topics like JavaScript, Python, and Machine learning.



Finding a good project to contribute to takes time. So make sure to look at the frequency of contributions to the project and the types of issues and features the project needs help with. If it aligns with your career goals, then start off with smaller issues and then move to larger features over time. You will also want to join the project's community chat room or Discord channel to connect with other contributors and core team members of the project.

Open Source Project Contribution

In this next section, you will want to list out the meaningful contributions you have made to different open source projects. This can include documentation, code features, bug fixes, etc. Remember that, you will want to list out meaningful contributions that match the goals you laid out earlier.

Basic Project Information

For each project contribution, you will want to list out the project name along with the PR and issue links. Then you will want to describe in a few short sentences why you chose this project and how it will help you with your career goals.

Section	Details
Project Name	[Project Name]
Project URL	[Project URL]
Why This Project?	[Reasons for Choosing This Project]
Type of Contribution	[Code, Documentation, Community Support, etc.]

Learn and Development

Once your contribution has been approved and merged into the project, you will want to reflect on what you have learned during this process. What types of new skills or technologies did you learn while working on this issue? What challenges did you face? How did you interact with the community when you needed clarification on an issue or needed help with the pull request?

Keeping track of this information is helpful because you can reference these experiences later when crafting your resume. Potential employers will want to see how you work with others, how you problem solve and how you deliver solutions in a project.

Feel free to link to the pull request and issue discussions as references on how you worked with contributors in this project.

Section	Details
Learning and Development	
New Skills Learned	[List of New Skills]
Challenges Overcome	[Challenges Faced and Resolved]
Community Interaction	[Meetups, Discussions, etc.]

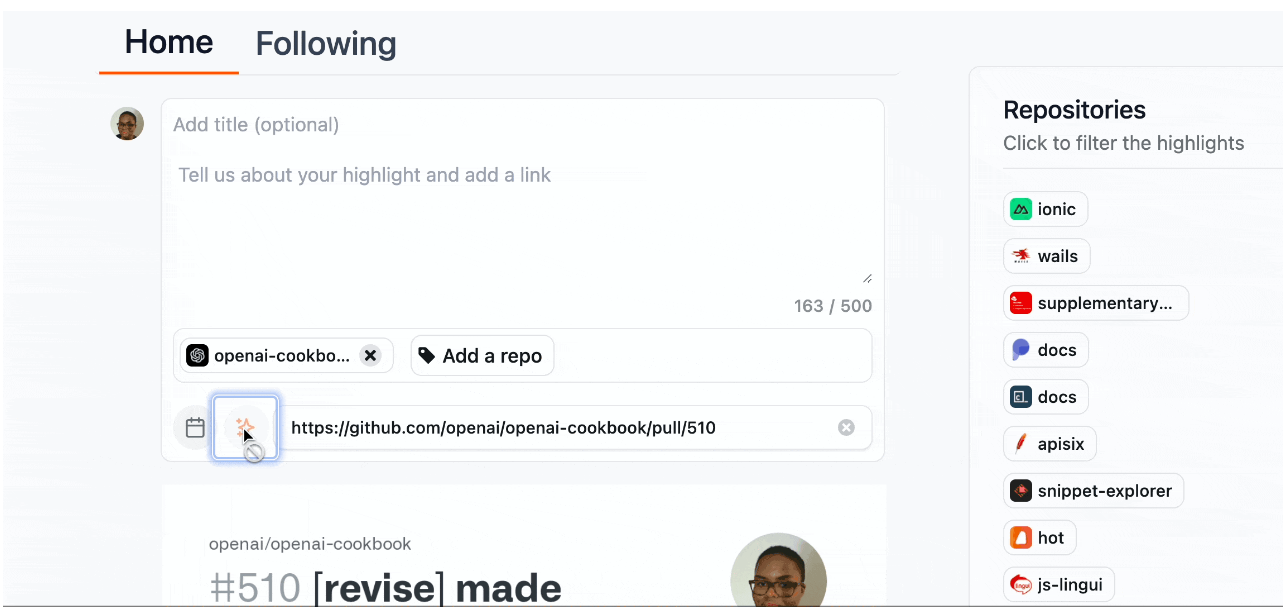
Contribution Tracking

For each project contribution, you will want to list the date and brief summary on what the issue was and how you solved it. It would also be good to list out any acknowledgements you received from other contributors or core team members on the project because it highlights the impact you left.

Section	Details
Contribution Tracking	
Date	[Date of Contribution]
Contribution Summary	[Summary of Contribution with Links]
Feedback Received/Milestones Achieved	[Feedback and Achievements]

Using the Highlights Feature to Track Your Contributions

The Highlights feature is the place you can display your favorite open source contributions, and share your open source story.



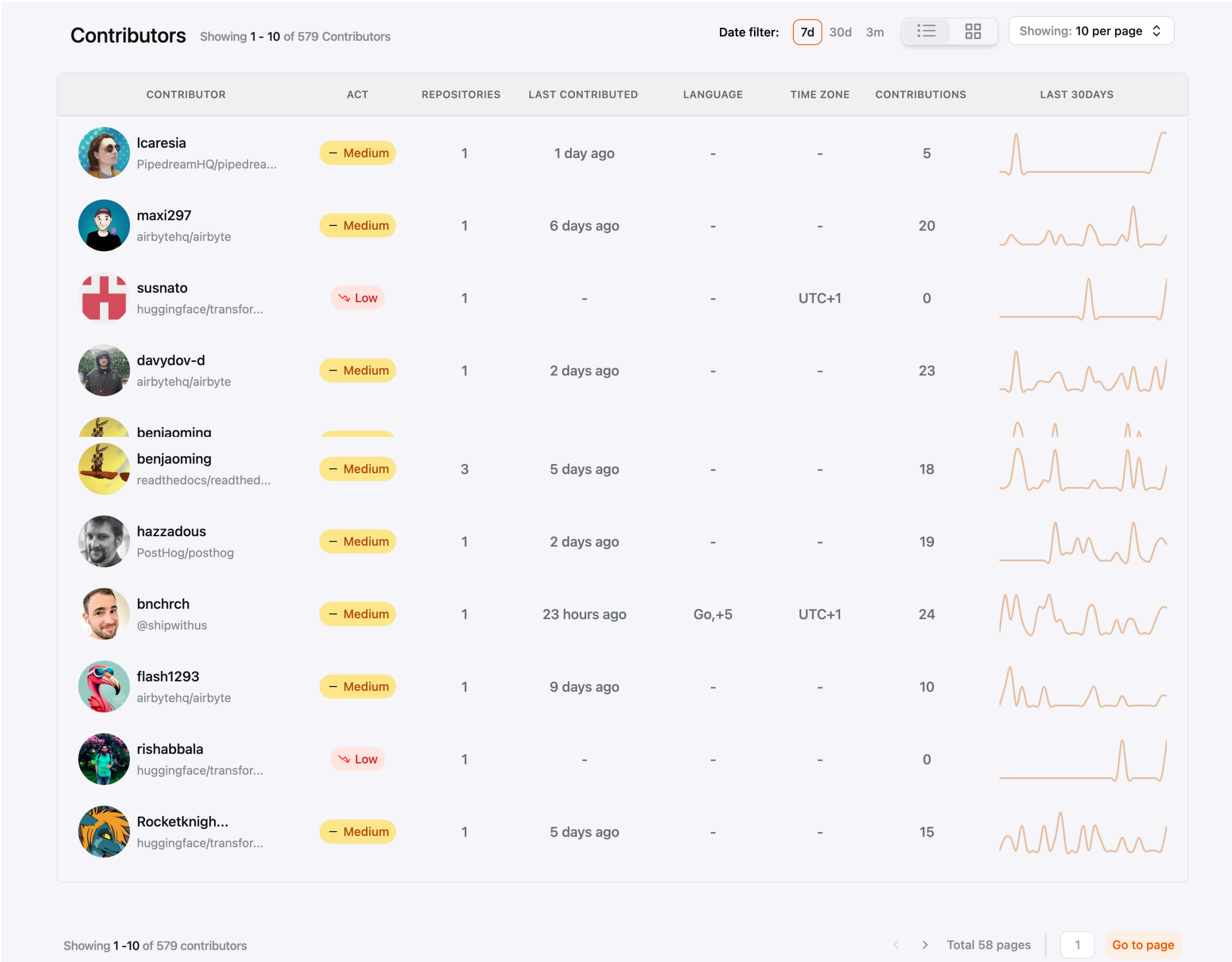
You can use this feature to help you track the open source contributions that you are most proud of and share your accomplishments with your network. All of your highlights will appear on your OpenSauced profile, so sharing your profile is another way to let potential employers see the types of contributions you've made and the highlight you've identified as the most important.

Using the Lists Feature to Add People You Interacted with During Your Contributions

The Lists feature enables you to categorize, monitor, and analyze different groups of contributors within open source projects.

You can use this feature to create a list of people you interacted with during your open source work. If you continually work with the same individuals, then they can act as referrals and references for jobs.

Lists can also be used to track other people in your network to reach out to about their contributions.



Contributors Showing 1 - 10 of 579 Contributors

Date filter: 7d 30d 3m

Showing: 10 per page

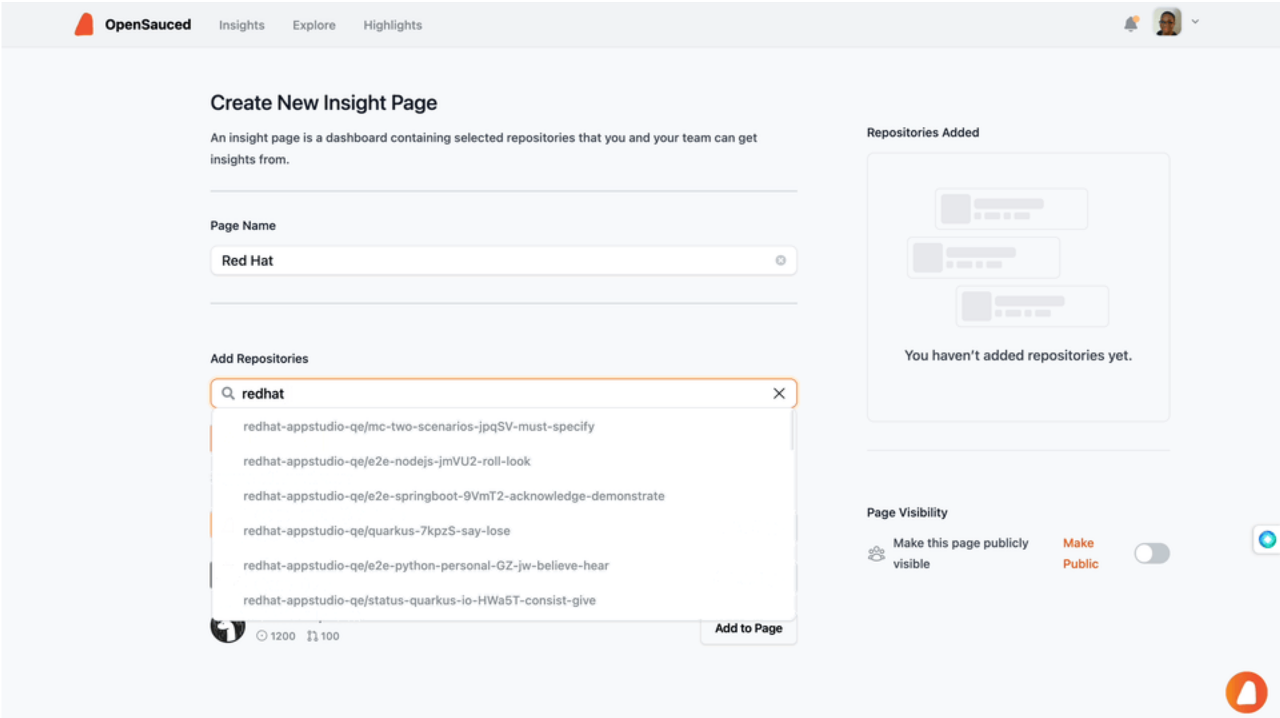
CONTRIBUTOR	ACT	REPOSITORIES	LAST CONTRIBUTED	LANGUAGE	TIME ZONE	CONTRIBUTIONS	LAST 30DAYS
lcaresia PipedreamHQ/pipedrea...	Medium	1	1 day ago	-	-	5	
maxi297 airbytehq/airbyte	Medium	1	6 days ago	-	-	20	
susnato huggingface/transfor...	Low	1	-	-	UTC+1	0	
davydov-d airbytehq/airbyte	Medium	1	2 days ago	-	-	23	
benjaoming readthedocs/readthed...	Medium	3	5 days ago	-	-	18	
hazzadous PostHog/posthog	Medium	1	2 days ago	-	-	19	
bnchrch @shipwithus	Medium	1	23 hours ago	Go,+5	UTC+1	24	
flash1293 airbytehq/airbyte	Medium	1	9 days ago	-	-	10	
rishabbala huggingface/transfor...	Low	1	-	-	-	0	
Rocketknigh... huggingface/transfor...	Medium	1	5 days ago	-	-	15	

Showing 1 -10 of 579 contributors

Total 58 pages 1 Go to page

Using the Insights Feature to Track the Repositories You Contributed to

The Insights feature is a comprehensive view of a project's health and individual contributions. You can use it to track the repositories you contributed to or to track repositories of open source companies that are hiring.



Networking and Professional Development

Some larger features can lead to networking and other professional development opportunities, like speaking at meetups or writing for tech publications. It is important that you list out who you mainly corresponded with for each contribution and any opportunities that came from this contribution.

If you have corresponded with the same core contributors and maintainers over the past few contributions, try reaching out about collaborating on other projects together. You never know where an opportunity will lead.

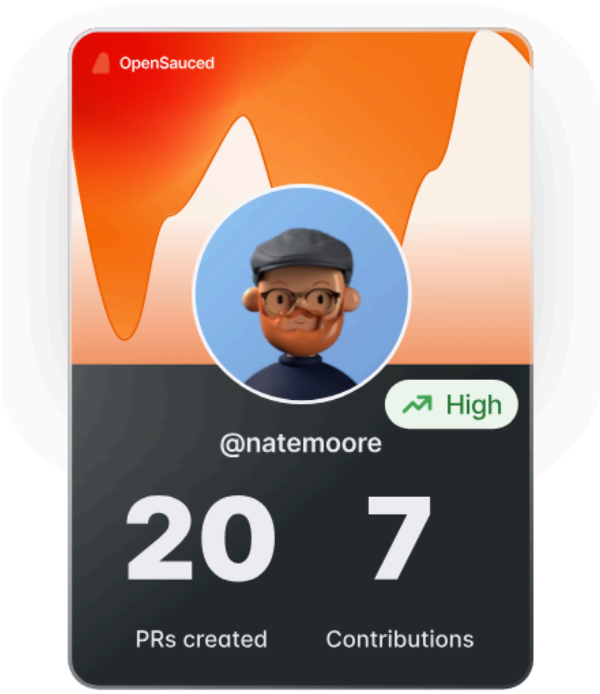
Section	Details
Networking and Professional Development	

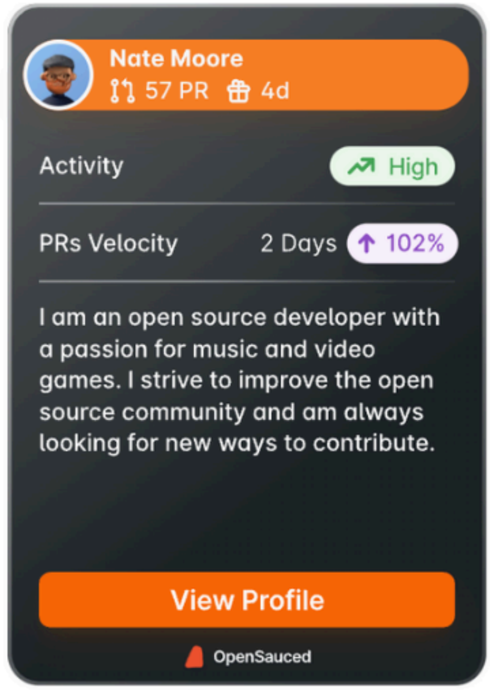
Section	Details
Contacts Made	[Contacts Made and Their Roles]
Insights Gained from Networking	[What You Learned from Networking]
Future Networking Opportunities	[Planned Networking Activities]

Using Your OpenSauced Dev Card for Networking

Your OpenSauced Dev card serves as your "open source business card" to showcase all of your recent open source activities. You will find recent activity concerning the projects you contribute to, frequency of pull requests and projects you are interested in.

You can easily share your Dev card on social media platforms like X/Twitter and LinkedIn to gain attention from potential recruiters and hiring managers.





Reflection and Next Steps

In this section, you will want to reflect on what went well with this contribution and areas in which you struggled with.

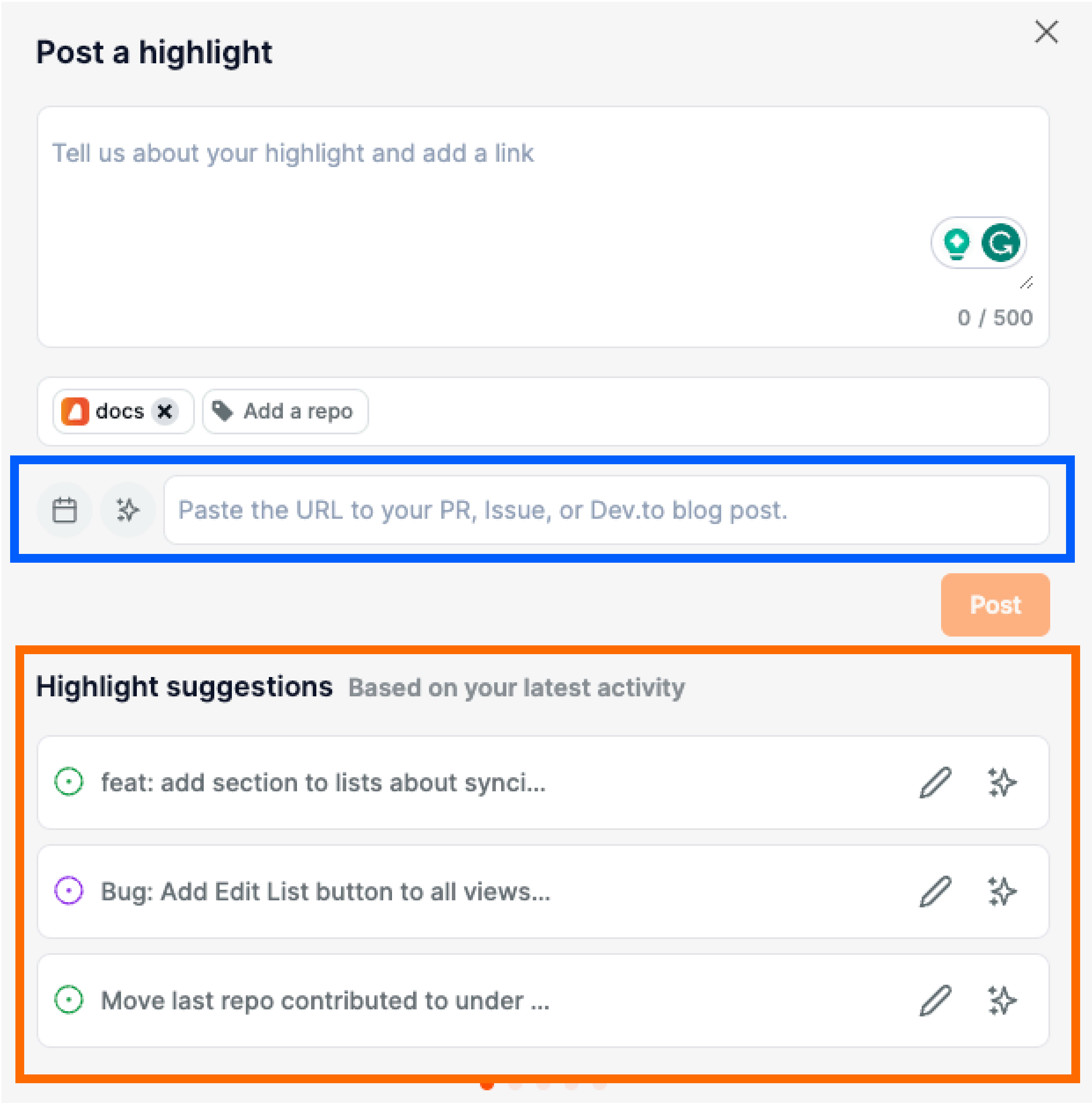
It is important to tie these reflections back to your career goals and desired job roles you defined earlier. If you are weaker in skills that are required for the jobs are seeking, use future open source contributions as a way to strengthen these skills.

Section	Details
Reflection and Next Steps	
Key Takeaways from This Project	[What You Learned]
Areas for Improvement	[Areas to Work On for Future Projects]
Next Project/Contribution Goal	[Your Next Goal in Open Source Contribution]

Using the Highlights Feature to Update Your Professional Profile

After each contribution, you will want to update your LinkedIn profile and resume to highlight what you have done recently. A lot of recruiters and hiring managers are on the look out for talented individuals on LinkedIn, so it is important to keep your profile up to date of your recent accomplishments.

The Highlights feature allows you to post about your recent open source contributions on OpenSauced and share it with other social media outlets like Twitter and LinkedIn.



Section	Details
Resume and Professional Profile Update	

Section	Details
Updates Made to Resume/LinkedIn	[Details of Updates]
Key Accomplishments Highlighted	[Key Points Highlighted]
Future Plans for Profile Enhancement	[Plans for Further Enhancing Your Profile]

Example Chart for Frontend Developer

Section	Details
Personal Information	
Name	Jane Smith
Contact Information	jane.smith@email.com
Professional Profile (LinkedIn/GitHub/OpenSauced profile)	LinkedIn, GitHub, OpenSauced

Section	Details
Career Goals	
Desired Job Role	Frontend Developer
Key Skills to Develop	React.js, CSS-in-JS, Web Accessibility
Industry/Area of Interest	Web Development, UI/UX Design
Open Source Project Contribution No.1	
Project Name	Awesome React Components
Project URL	GitHub Repository, OpenSauced Insight Page
Why This Project?	Interested in learning about creating reusable UI components.
Type of Contribution	Implemented a new responsive carousel component.
Learning and Development	
New Skills	Styled Components, Carousel Design Patterns

Section	Details
Learned	
Challenges Overcome	Overcame performance issues with large datasets.
Community Interaction	Participated in design discussions on the project's Discord channel.
Contribution Tracking	
Date	2023-03-10
Contribution Summary	Pull Request #456
Feedback Received/ Milestones Achieved	Positive feedback on the elegant design; milestone: merged into main branch.
Networking and Professional Development	
Contacts Made	Connected with project maintainers and frontend enthusiasts.
Insights Gained	Learned about upcoming trends in frontend

Section	Details
from Networking	development.
Future Networking Opportunities	Planning to attend the project's monthly virtual meetup.
Reflection and Next Steps	
Key Takeaways from This Project	Improved skills in component design and collaboration.
Areas for Improvement	Intend to deepen knowledge in state management with Redux.
Next Project/ Contribution Goal	Contribute to a project focused on enhancing web accessibility.
Resume and Professional Profile Update	
Updates Made to Resume/LinkedIn	Successfully implemented a responsive carousel component, enhancing user experience and receiving positive feedback. Demonstrated expertise in React.js and CSS-in-JS.
Key Accomplishments Highlighted	Highlighted successful implementation of the responsive carousel.

Section	Details
Future Plans for Profile Enhancement	Attend frontend development conferences and workshops.

Additional Resources

If you're looking for more resources about the job hunt, check out these OpenSauced posts:

- Building Your DevRel Resume with Open Source
- How to Talk About Your Open Source Experience in a Tech Interview