

Install Exalate for GitHub

Last Modified on 01/15/2026 8:02 pm EST

Introduction

You have two options for installation:

- install Exalate for GitHub from the marketplace.
- install Exalate for GitHub on docker.

Once the installation itself is done, the next two steps are:

- generate a personal access token to log into the Exalate admin console.
- configure proxy user permissions.

Note: Exalate can be installed on Github Cloud and GitHub Enterprise Cloud.

Install Exalate from the Marketplace

Exalate for GitHub is available on the [marketplace](#).


You can install Exalate for organizations or user accounts and grant access to repositories. The app is hosted on our server.

Note: To install Exalate, you need to be an organization owner or have admin permissions in a repository.

1. Go to the Marketplace and search for "exalate".

The screenshot shows the GitHub Marketplace interface. At the top, there's a header with the GitHub logo and 'Marketplace' text. A search bar on the right contains the text 'Type / to search'. Below the header, the page is titled 'Marketplace / Search results'. On the left, there's a sidebar with 'Types' (Apps, Actions) and 'Categories' (API management, Chat, Code quality, Code review, Continuous integration, Dependency management). The main content area shows a search for 'Exalate' with '1 result for "Exalate"'. Under the 'Apps' section, the first result is 'Exalate Two-Way Issue Sync [Jira, ServiceNow, Azure DevOps, Zendesk, Salesforce & More]' by the 'Exalate-team'. The description states: 'Don't be confined by only no-code integration apps. Set up an integration with the help of scripts across Github and other tools'. It also shows '1.9k installs'.

2. Click **Set up a new plan**



Application

Exalate Two-Way Issue Sync

[Jira, ServiceNow, Azure DevOps, Zendesk, Salesforce & More]

GitHub has verified that the publisher controls the domain and meets other [requirements](#).

Categories

- Utilities
- AI Assisted
- Project management
- Free

Set up a new plan Edit your plan Configure access

Exalate is the only decentralized solution for GitHub integrations. It allows your team to bidirectionally share data and sync with other teams working on other platforms. While giving each side full control over their incoming and outgoing data.

3. Scroll down and search for the account to associate with Exalate.

Pricing and setup

Exalate Issue Sync

Automatically sync Github issues with Jira or other trackers. Start with the Free Plan or try the Premium Plan (30-day free trial)

\$0

Exalate Two-Way Issue Sync [Jira, ServiceNow, Azure DevOps, Zendesk, Salesforce & More]

Exalate Issue Sync

Automatically sync Github issues with Jira or other trackers. Start with the Free Plan or try the Premium Plan (30-day free trial)

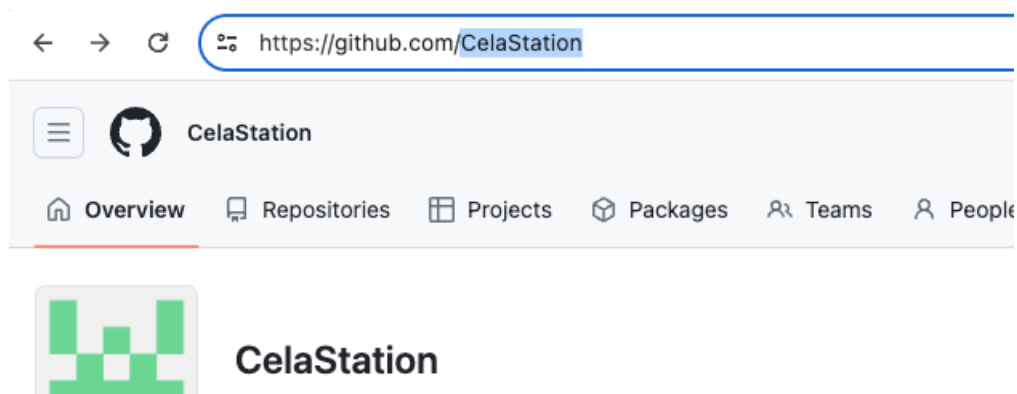
Account:

Install it for free Next: Confirm your installation location.

Exalate Two-Way Issue Sync [Jira, ServiceNow, Azure DevOps, Zendesk, Salesforce & More] is provided by a third-party and is governed by separate [privacy policy](#) and [support documentation](#).

In **Account**, you can either install Exalate for your entire account, or enter the name of a specific organization within your GitHub account. For our example, we enter the name of an organization we created.

You can search for an organization by name, or go to your GitHub instance and copy it:



4. When done click **Install it for free**.

5. Now review your order and associate your billing information with your organization.

Due today
Prorated for Nov 22nd-Dec 21st


\$0.00

i An organization owner or billing manager must link their personal billing information with this organization account. You can switch to a business account to use your business' billing information by [signing](#) the [GitHub Customer Agreement](#).

☒

Allow my billing information to be linked with this organization

Confirm your details shown below. To make changes, [edit your billing information](#)



CelaStation

City
State
Zip
Country

Cancel

Save

6. Click **Save** when done.

7. Click **Complete order and begin the installation**.

By clicking "Complete order and begin installation", you agree to Exalate Two-Way Issue Sync [Jira, ServiceNow, Azure DevOps, Zendesk, Salesforce & More]'s [Privacy Policy](#). You previously agreed to the [Marketplace Terms of Service](#).

Complete order and begin installation

Next: Authorize Exalate Two-Way Issue Sync [Jira, ServiceNow, Azure DevOps, Zendesk, Salesforce & More] to access your account.

8. On the next screen click **Install**.



Install Exalate Issue Sync

Install on your organization CelaStation 

for these repositories:

☒ **All repositories**

This applies to all future repositories owned by the resource owner.
Also includes public repositories (read-only).

with these permissions:

- ✓ **Read** access to metadata
- ✓ **Read and write** access to issues and pull requests

Install




Cancel

Next: you'll be directed to the GitHub App's site to complete setup.


Note: By default, Exalate is installed on the account that is currently logged in. If you want to change to a different one, click **Cancel** and follow the prompts to select a different one.

Note: After clicking **Install**, you are redirected to the Exalate admin console. We recommend saving the link to easily access Exalate in the future. You can always recover your Exalate app link here: <https://connect.exalate.net/github/find>

9. On the **Registration** page fill out the required fields.


 [github.com](#)
 [Registration](#)

Registration

Email  *

Contact name*

Organization*

Phone*

By clicking Agree and submit below, you agree to our [End User License Agreement](#)

[Agree and submit](#)

[More Integrations](#) [Documentation](#) [EULA](#) [Support](#) [Report a bug](#)

10. Click **Agree and submit**.

11. An email confirmation is sent to the email you specified for your account. With this info Exalate:

- Verifies your instance.
- Activates an evaluation license.
- Makes an admin account to receive error notifications.

12. Open your email and click **Verify Exalate instance**.



Exalate License Manager <sales@exalate.com>

to me ▾

1:20 PM (0 minutes ago)



Hi,

Thank you for trying out Exalate

Click here to verify your Exalate node:

[Verify Exalate instance](#)

After verifying your instance, you will get an admin account with a 30-day evaluation license.

Use this link to access the Exalate admin console: <https://githubnode-...>

Check out this short guide how to verify your Exalate instance: [How to verify your Exalate instance](#)

If you have any questions - please [raise a ticket on our service portal](#).

Enjoy,

The Exalate team

13. After the validation is successful you are redirected back to the Exalate console. Click **Continue** to go on to the **Registration** page.



github.com



Log in

Email validation successful



Your email was successfully verified

[Continue](#)

[More Integrations](#) [Documentation](#) [EULA](#) [Support](#) [Report a bug](#)

14. To login to the Exalate console, enter your [personal access token](#) in the **Token** field.

Log in

Use your GitHub personal access token.

For more details, check [how to generate a personal access token](#).

Token*

Exalate is not storing any credential information.

[Read more about security](#).

Log in

[More Integrations](#) [Documentation](#) [EULA](#) [Support](#) [Report a bug](#)

How to Generate a Personal Access Token

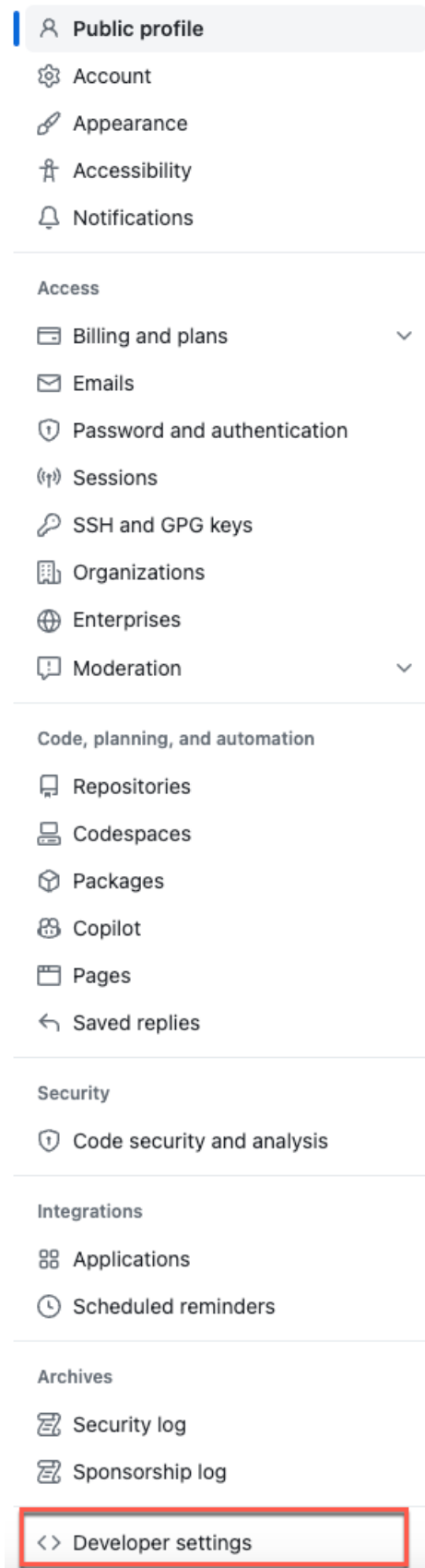
Note: You need a Personal Access Token to log in to Exalate for GitHub.

1. Log in to your GitHub account.

The GitHub user must be an organization member or have admin access to a repository where Exalate is installed.

Note: The user account that wishes to access the Exalate dashboard needs to have **organization visibility** set to **Public**.

2. In GitHub, go to **Settings > Developer settings**.



3. Select **Personal access tokens > Tokens (classic)**

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

Personal access tokens (classic)

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Token1 — admin:enterprise, admin:pgp_key, admin:org, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete:packages, delete_repo, project, repo, workflow, write:discussion, write:packages

Last used within the last week

Delete

Expires on Wed, Feb 21 2024.

4. Click **Generate new token** to generate a new access token. Select the scopes or permissions you'd like to grant this token. To use your token to access repositories from the command line, select **repo**.

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

Edit personal access token (classic)

If you've lost or forgotten this token, you can regenerate it, but be aware that any scripts or applications using this token will need to be updated.

Regenerate token

Note

Token1

What's this token for?

Expiration

This token expires on Wed, Feb 21 2024. To set a new expiration date, you must [regenerate the token](#).

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

☒ repo

Full control of private repositories

☒ repo:status

Access commit status

☒ repo_deployment

Access deployment status

☒ public_repo

Access public repositories

☒ repo:invite

Access repository invitations

☒ security_events

Read and write security events

☒ workflow

Update GitHub Action workflows

☒ write:packages

Upload packages to GitHub Package Registry

☐ read:packages

Download packages from GitHub Package Registry

☒ delete:packages

Delete packages from GitHub Package Registry

☒ admin:org

Full control of orgs and teams, read and write org projects

☒ write:org

Read and write org and team membership, read and write org projects

☒ read:org

Read org and team membership, read org projects

☒ manage_runners:org

Manage org runners and runner groups

☒ admin:public_key

Full control of user public keys

Once the token has been generated, you can now use it to log in to your Exalate admin console.

Important: Make sure to save the token, because you can access it only once.

Note: Starting from version 5.1.4 we apply security improvements involving migration to OAuth for the Exalate application for Github.

Configuring Proxy User Permissions

By default, Exalate for GitHub proxy user is the repository admin or the organization owner, who is installing Exalate.

Proxy user permissions in GitHub

The proxy user has the same permissions as an admin or an organization owner in GitHub.

Note: For more information about user permissions, please read these GitHub Docs articles:

- [Permission levels for a user account repository](#)
- [Permission levels for an organization](#)
- [Repository permission levels for an organization](#)

Install Exalate for GitHub on Docker

Learn more about [How to install Exalate for GitHub on docker](#)

Product

[About Us](#) ?

[Release History](#) ?

[Glossary](#) ?

[API Reference](#) ?

[Security](#) ?

[Pricing and Licensing](#) ?

Resources

[Subscribe for a weekly Exalate hack](#) ?

[Academy](#) ?

[Blog](#) ?

[YouTube Channel](#) ?

[Ebooks](#) ?

Still need help?

[Join our Community](#) ?

[Visit our Service Desk](#) ?

[Find a Partner](#) ?

Video Tutorial