

# 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.

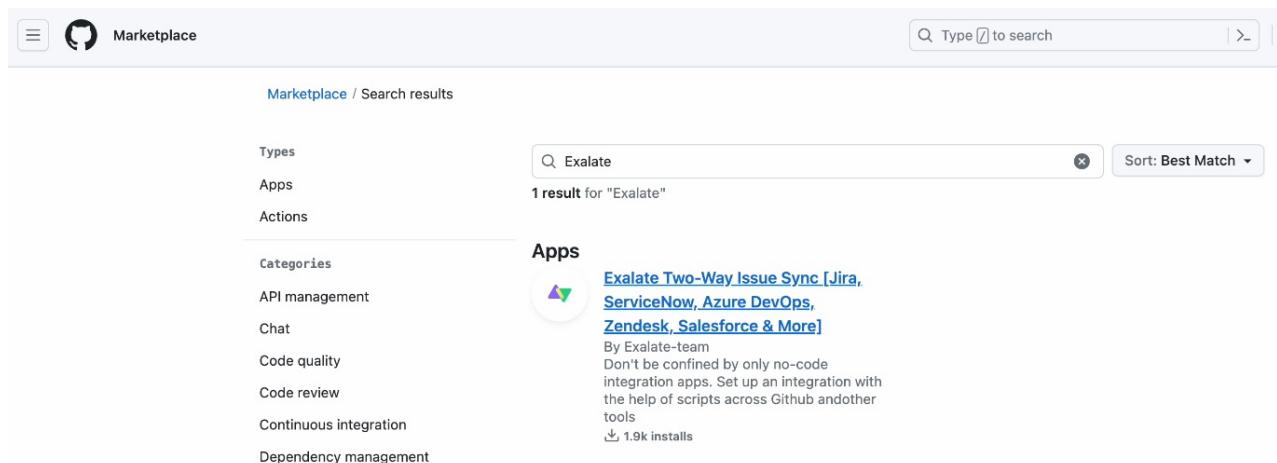
## Install Exalate from the Marketplace

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.

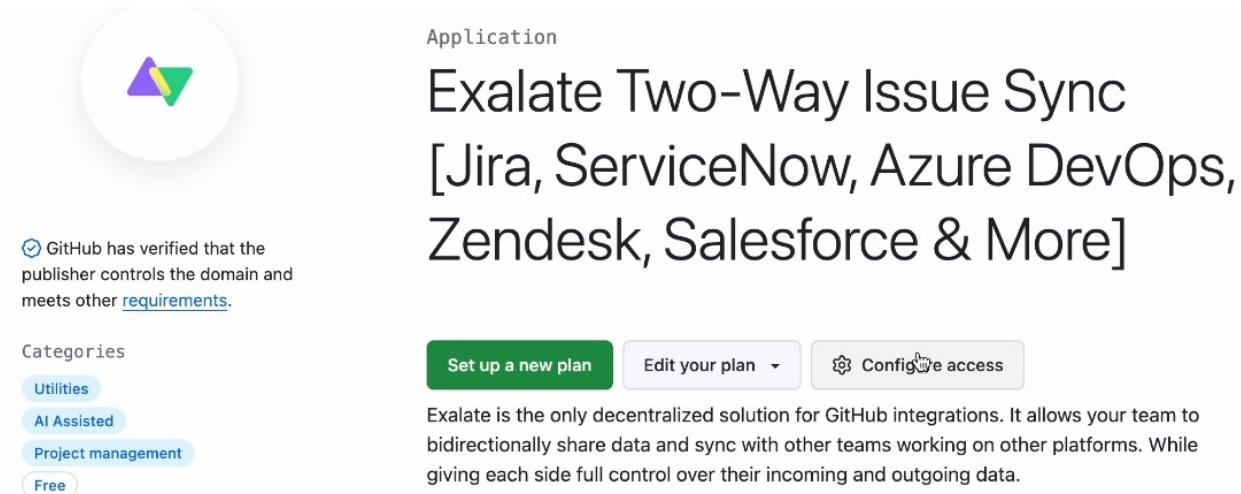
**Note:** To install Exalate, you need to be an organization owner or have admin permissions in a repository.

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



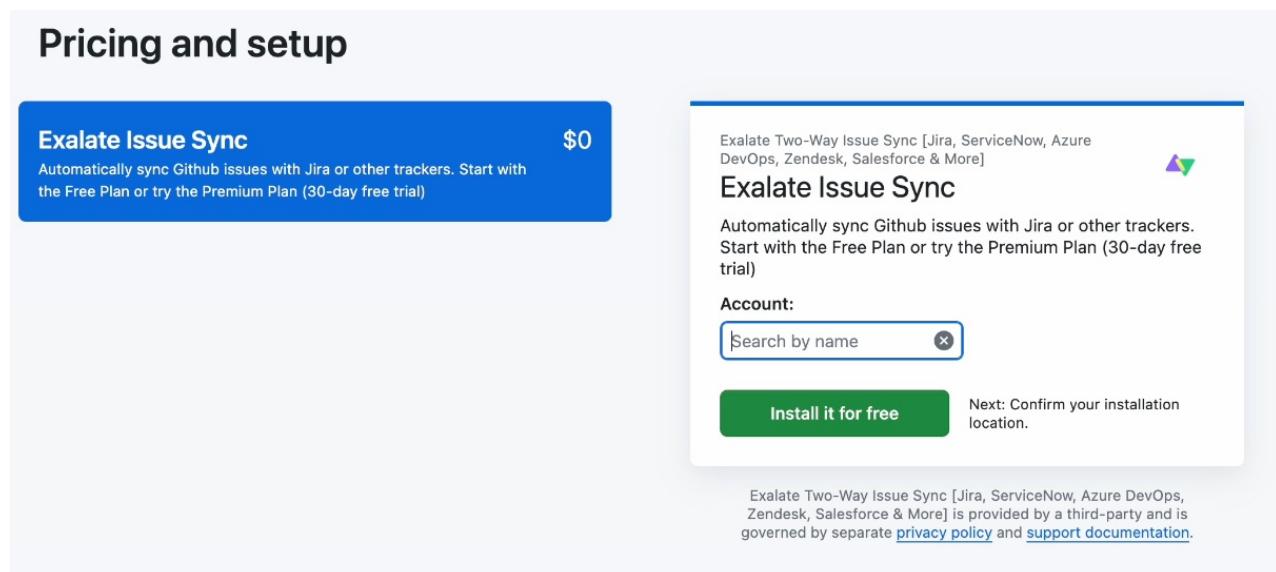
The screenshot shows the GitHub Marketplace search results for the term "Exalate". The search bar at the top contains "Exalate". Below the search bar, there are filters for "Types" (set to "Apps"), "Actions", and "Categories" (which includes "API management", "Chat", "Code quality", "Code review", "Continuous integration", and "Dependency management"). The main results section is titled "Apps" and shows a single result: "Exalate Two-Way Issue Sync [Jira, ServiceNow, Azure DevOps, Zendesk, Salesforce & More]". This result is described as being by "Exalate-team" and noting that "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 shows "1.9k installs".

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



The screenshot shows the GitHub Marketplace page for the "Exalate Two-Way Issue Sync" application. At the top, there's a circular icon with the Android logo. Below it, the text "Application" and the app's name are displayed. A green badge indicates that GitHub has verified the publisher. A note states that the publisher controls the domain and meets other requirements. The app is categorized under "Utilities", "AI Assisted", "Project management", and "Free". Three buttons are available: "Set up a new plan" (green), "Edit your plan" (grey), and "Configure access" (grey). A descriptive text block explains that Exalate is a decentralized solution for GitHub integrations, allowing bidirectional data sharing and sync between teams on various platforms. It also mentions giving each side full control over their incoming and outgoing data.

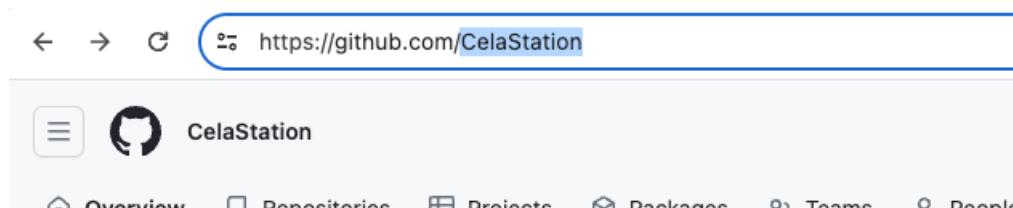
3. Scroll down and search for the account to associate with Exalate.



The screenshot shows the "Pricing and setup" section of the Exalate app page. It features a blue box for the "Exalate Issue Sync" plan, which is \$0 and includes a 30-day free trial. Below this, there's a search bar for "Account" with the placeholder "Search by name". A green button says "Install it for free". To the right, there's a note about the app being provided by a third-party and links to the "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:



The screenshot shows a web browser with the URL "https://github.com/CelaStation" in the address bar. The GitHub interface is visible, showing the "Overview" tab is selected. The organization name "CelaStation" is displayed, along with icons for "Repositories", "Projects", "Packages", "Teams", and "People".



**CelaStation**

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**

**ⓘ** 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

Cancel Save

6. Click **Save** when done.

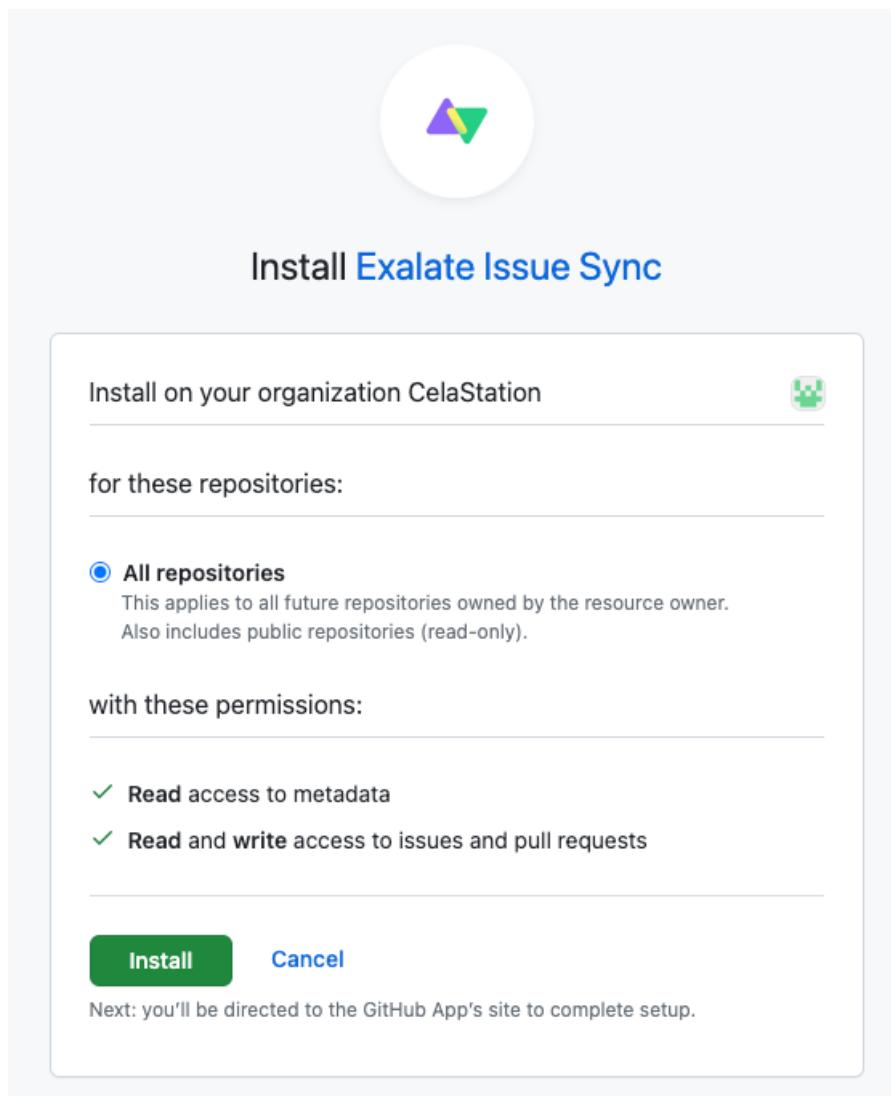
7. Click **Complete order and begin the installation**.

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**.



**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.



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 node unlock request for https://github.com

Inbox x



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.

The screenshot shows the Exalate admin console. On the left, there is a sidebar with the Exalate logo, a GitHub icon, and a 'Log in' button. The main content area has a purple header bar with the Exalate logo. Below the header, the text 'Email validation successful' is displayed in a purple box. To the right of this text is a large Exalate logo. At the bottom of the page, there is a green 'Continue' button. At the very bottom, there is a footer with links: 'More Integrations', 'Documentation', 'EULA', 'Support', and '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

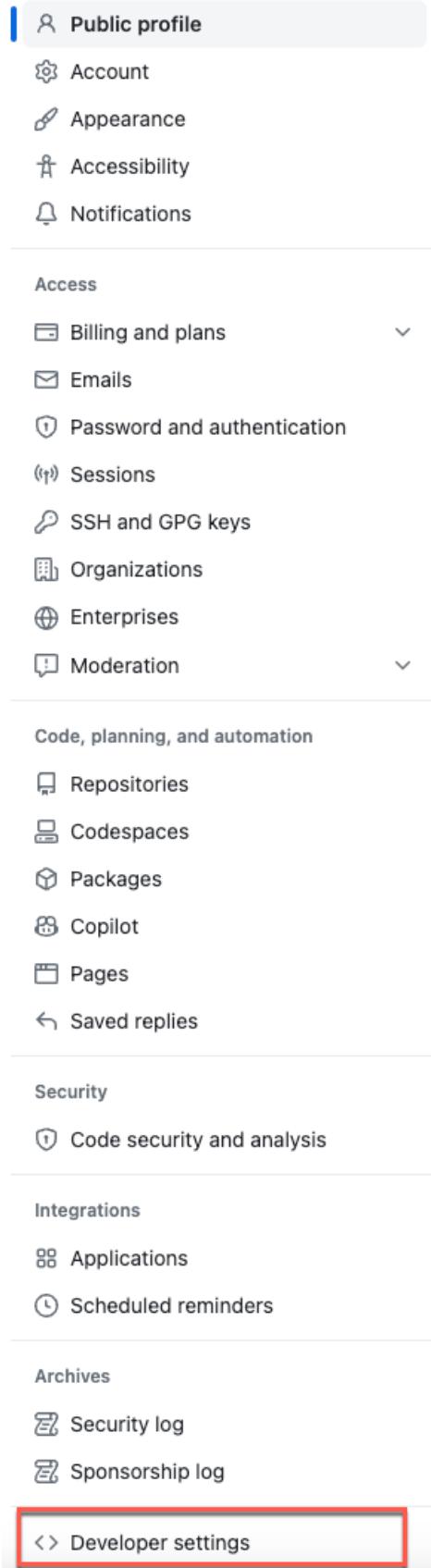
**Note:** You need a Personal Access Token to log in to Exalate for GitHub.

1. Log in to your GitHub account.

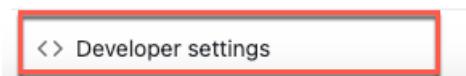
The GitHub user must be an organization member or have admin access to a repository where Exalate is installed.

**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**.



The image shows the GitHub Settings sidebar. The 'Public profile' section is selected. Other sections include 'Account', 'Appearance', 'Accessibility', and 'Notifications'. The 'Access' section contains 'Billing and plans', 'Emails', 'Password and authentication', 'Sessions', 'SSH and GPG keys', 'Organizations', 'Enterprises', and 'Moderation'. The 'Code, planning, and automation' section contains 'Repositories', 'Codespaces', 'Packages', 'Copilot', 'Pages', and 'Saved replies'. The 'Security' section contains 'Code security and analysis'. The 'Integrations' section contains 'Applications' and 'Scheduled reminders'. The 'Archives' section contains 'Security log' and 'Sponsorship log'. The 'Developer settings' option is highlighted with a red box.

-  **Public profile**
-  [Account](#)
-  [Appearance](#)
-  [Accessibility](#)
-  [Notifications](#)
- **Access**
  -  [Billing and plans](#) 
  -  [Emails](#)
  -  [Password and authentication](#)
  -  [Sessions](#)
  -  [SSH and GPG keys](#)
  -  [Organizations](#)
  -  [Enterprises](#)
  -  [Moderation](#) 
- **Code, planning, and automation**
  -  [Repositories](#)
  -  [Codespaces](#)
  -  [Packages](#)
  -  [Copilot](#)
  -  [Pages](#)
  -  [Saved replies](#)
- **Security**
  -  [Code security and analysis](#)
- **Integrations**
  -  [Applications](#)
  -  [Scheduled reminders](#)
- **Archives**
  -  [Security log](#)
  -  [Sponsorship log](#)
- [Developer settings](#)

3. Select **Personal access 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:gpg\_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

Expires on **Wed, Feb 21 2024**.

Delete

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**.

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](#).

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input type="checkbox"/> <b>repo:status</b>	Access commit status
<input checked="" type="checkbox"/> <b>repo_deployment</b>	Access deployment status
<input type="checkbox"/> <b>public_repo</b>	Access public repositories
<input type="checkbox"/> <b>repo:invite</b>	Access repository invitations
<input type="checkbox"/> <b>security_events</b>	Read and write security events
<input checked="" type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input checked="" type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input type="checkbox"/> <b>read:packages</b>	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams, read and write org projects
<input checked="" type="checkbox"/> <b>write:org</b>	Read and write org and team membership, read and write org projects
<input checked="" type="checkbox"/> <b>read:org</b>	Read org and team membership, read org projects
<input checked="" type="checkbox"/> <b>manage_runners:org</b>	Manage org runners and runner groups
<input checked="" type="checkbox"/> <b>admin:public_key</b>	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](#) 