

# Proxy User

Last Modified on 03/13/2024 8:08 am EDT

A proxy user is a work management system user, who is used by Exalate to make changes, such as:

- Creating issues
- Updating data

The proxy user impersonates external instances. All changes on local issues are performed on behalf of this user. You can use an existing user account or create a new one, specifically dedicated to Exalate.

**Note:** Changes made by the proxy user are not synchronized. If you set the administrator as a proxy user and create issues with the help of the '**create on behalf of**' Service Desk functionality, issues are not synchronized.

The proxy user configuration is different for each work management system.

## Azure DevOps

### Access to Azure DevOps Exalate Console (Log-in)

Exalate needs to authenticate to the Azure DevOps instance. You can provide such access using the Personal Access Token(PAT).

**Note:** Use the PAT to access the Exalate admin console. Please read [How to generate the PAT\(Personal Access Token\)](#) in your Azure DevOps instance.

### Proxy user in Azure DevOps

The proxy user is the Azure DevOps user account that fetches information from the Azure DevOps instance and updates Work items with incoming changes.

The configuration of the proxy user takes place during the first installation of Exalate. The user that is installing Exalate automatically becomes the proxy user.

You can change the proxy user when the app is installed and running. To do so navigate to **Exalate Menu Panel → General Settings → Configure** and enter the PAT of the new proxy user.

### Proxy User Permissions in Azure DevOps

The proxy user must be a Project Administrator and a member of the Project Collection Administrator group on the Organization level

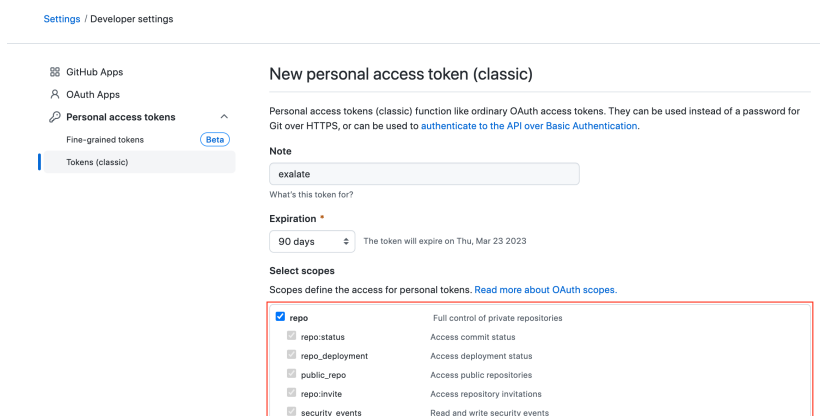
---

**Note:** You can add the Project Administrator to the Project Collection Administrator in your Azure DevOps under **Organization Settings → Permissions → Project Collection Administrators**.

## GitHub

### Access to GitHub Exalate Console (Log-in)

Use a personal access token to log in to Exalate for GitHub. The token needs to have access to private repositories with the **repo** scope.



Settings / Developer settings

GitHub Apps  
OAuth Apps  
Personal access tokens  
Fine-grained tokens (Beta)  
Tokens (classic)

### New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

exalate

What's this token for?

**Expiration**

90 days The token will expire on Thu, Mar 23 2023

**Select scopes**

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

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo.status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo_invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events

### Proxy User in GitHub

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

The proxy user configuration takes place during the first installation of Exalate. The user that installs the app automatically becomes a proxy user.

The proxy user must have a **Personal Access Token** with authorization to private repositories with the **repo** scope.

**Note:** For more information on how to generate a personal access token please see [GitHub: How to generate an access token](#).

### Proxy User Permissions in GitHub

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

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

[Permission levels for a user account repository](#)

[Permission levels for an organization](#)

[Repository permission levels for an organization](#)

## HP ALM/QC

**Warning:** We are moving Exalate for HP QC/ALM into basic maintenance mode. This transition period will last until November 2024, after which support will be halted completely. For more information, please see <https://exalate.com/blog/end-of-support-hp/>.

### Access to HP ALM/QC Exalate Console (Log-in)

Log in to the Exalate app admin console with the credentials of the HP ALM/QC admin user. This can be a user you set up during the first configuration of the Exalate app for HP ALM/QC.

### Proxy user in HP ALM/QC

You can set up a proxy user while your first-time Exalate configuration. The proxy user can be changed later on in the General Settings.

To change the proxy user, navigate to **Exalate Menu Panel → General Settings → Configure**.

### Proxy User Permissions in HP ALM/QC

The user should have administrative permissions to be able to see all the HP ALM/QC projects and user field configurations.

**Note:** For more information about user permissions in HP ALM/QC, please read [HP ALM/QC documentation](#).

## Jira Cloud

### Proxy user in Jira Cloud

The proxy user in Jira Cloud is the app user that is being created automatically. This user cannot be modified. The username is **Exalate** and the email address is **com.exalate.jiranode@connect.atlassian.com**.

The proxy user in Jira Cloud is a member of the following user groups:

- atlassian-addons
- atlassian-addons-admin
- jira-core-users
- jira-servicedesk-users
- jira-software-users

**Important:** At the moment, there is a security vulnerability in Exalate that allows you to access private project data with the Connect operation. We recommend making sure that the

proxy user has access only to public projects. For more info, please see [Security Vulnerability – You can access restricted project data with the Connect operation.](#)

## Proxy User Permissions in Jira Cloud

Jira Cloud grants correct permissions to apps through the **atlassian-addons-project-access** role. It is done after installing or updating an app. Jira Cloud also checks the permissions of existing apps across all Jira and Jira Service Desk projects and grants them the correct permissions.

If you want to ensure that the app has no access to the project - remove the group from the corresponding permission in the permission scheme.

The screenshot shows the 'Default software scheme' settings in Jira Cloud. It includes a breadcrumb trail 'Projects / Bugs 'R US / Project settings', a 'Permission helper' button, and an 'Actions' menu. The main heading is 'Default software scheme'. Below this, there is explanatory text about project permissions and a note that the project is using the 'Default software scheme'. A section titled 'Project Permissions' contains a table with two columns: 'Permission' and 'Users / Groups / Project Roles'. The table lists several permissions, with the 'Project Role (atlassian-addons-project-access)' role circled in red under the 'Browse Projects' permission.

Permission	Users / Groups / Project Roles
<b>Administer Projects</b> Ability to administer a project in Jira.	Project Role (Administrators) Project Role (atlassian-addons-project-access)
<b>Browse Projects</b> Ability to browse projects and the issues within them.	Project Role (atlassian-addons-project-access) Application access (Any logged in user)
<b>Manage sprints</b> Ability to manage sprints.	Project Role (atlassian-addons-project-access) Application access (Any logged in user)
<b>Service Project Agent</b> Allows users to interact with customers and access Jira Service Management features of a project.	Project Role (atlassian-addons-project-access)
<b>View Development Tools</b> Allows users in a software project to view development-related information on the issue, such as commits, reviews and build	Project Role (atlassian-addons-project-access)

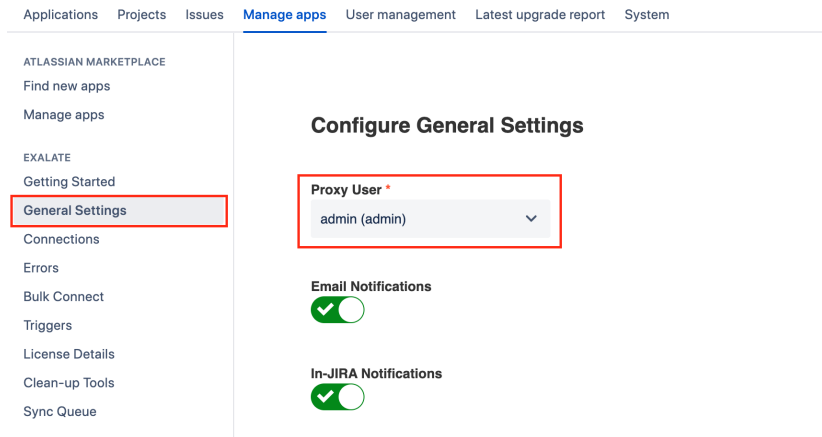
**Note:** For more information, please see the [Atlassian documentation](#)

## Jira on-premise

### Proxy user in Jira on-premise

By default, the proxy user is the user who installs Exalate. You can change the proxy user by navigating to **Exalate Menu Panel → General Settings → Configure.**

**Important:** At the moment, there is a security vulnerability in Exalate that allows you to access private project data with the Connect operation. We recommend making sure that the proxy user has access only to public projects. For more info, please see [Security Vulnerability – You can access restricted project data with the Connect operation.](#)



## Proxy User Permissions in Jira-on premise

In Jira on-premise, the proxy user needs to have the following permissions:

- Browse Project
- Create issue
- Edit issue
- Link issue
- Transition issue: change statuses (on issue transition)
- If comments are synchronized, the proxy user needs to add, edit, and delete a comment
- If attachments are synchronized, the proxy user needs to add, and delete attachments
- If work logs are synchronized, the proxy user needs to add, edit, and delete work logs
- If security levels are synchronized, the proxy user needs to access the security levels
- If you're using a trigger the proxy user must be able to search for issues

**Note:** In **Jira Service Management**, the proxy user needs to be a service desk agent.

## Salesforce

Proxy user in Salesforce

In Salesforce, the user that is installing Exalate automatically becomes a proxy user.

## ServiceNow

Access to ServiceNow Exalate Console (Log-in)

You can access the ServiceNow instance in one of these ways:

### Basic login:

In order to log in you use a Username and a Password. Exalate does not store the password in the database, but uses the rest connection to attempt to log in to the ServiceNow node.

## OAuth token:

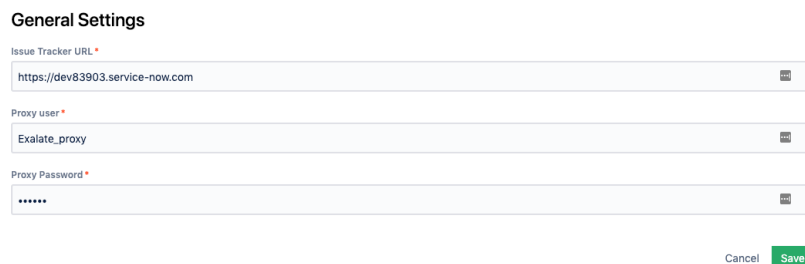
Authentication with a Username and an OAuth token. Exalate stores the token and uses it to access it. The token is refreshed every time the lifespan ends.

**Note:** OAuth token can be used as long as the refresh token is valid. For more information on how to set up the refresh token please see [Access the Exalate app in ServiceNow](#). You need to generate a new refresh token after the old one is expired. We suggest setting a longer lifespan for the refresh token.

## Proxy user in ServiceNow

To change the proxy user in Exalate for ServiceNow:

1. Log in to the Exalate admin console.
2. Navigate to **General Settings**.
3. Input details:
  - Servicenow instance URL.
  - Proxy user name.
  - Proxy user password.



The screenshot shows the 'General Settings' form in the Exalate admin console. It contains three input fields: 'Issue Tracker URL' with the value 'https://dev83903.service-now.com', 'Proxy user' with the value 'Exalate\_proxy', and 'Proxy Password' which is masked with dots. There are 'Cancel' and 'Save' buttons at the bottom right of the form.

## Proxy user permissions in ServiceNow

**Note:** Please read [what permissions the Proxy user must have in ServiceNow](#).

## Users and Permissions

**Note:** For security reasons, it is better to create a separate role with specific permissions for a proxy user instead of giving him an administrator role.

To integrate Exalate with ServiceNow you need 2 ServiceNow user accounts:

### Proxy User

The ServiceNow user account that fetches information from the ServiceNow instance and updates

the ServiceNow entities with incoming changes.

The proxy user can integrate various tables or attributes depending on the permissions defined by his user role in ServiceNow.

### Exalate Console users

ServiceNow user that is authorized to configure the Exalate app for ServiceNow. The Exalate console user must be an admin in your ServiceNow instance or the proxy user.

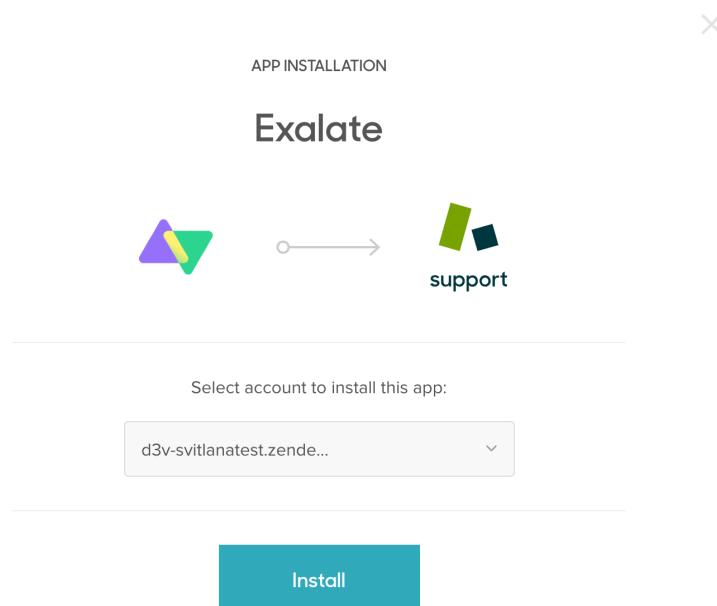
Exalate uses REST API to communicate with the ServiceNow issue tracker. By default, ServiceNow REST APIs use basic authentication or OAuth to authorize user access to REST APIs/endpoints. Therefore, the Exalate console users must have access to the ServiceNow instance admin configuration.

**Note: Role Management V2 REST API plugin must be installed** and activated on your ServiceNow instance. Starting with the New York version, this plugin is included by default. If you have recently updated your ServiceNow instance to the latest version, you need to **activate Role Management V2 REST API plugin manually**. For more information please see [ServiceNow contextual security](#).

## Zendesk

Proxy user in Zendesk

By default, Exalate sets Zendesk instance admin as the proxy user during installation.



**Note:** Exalate requires a dedicated Zendesk admin as a proxy user.

## Proxy user permissions in Zendesk

The proxy user can restrict the roles or groups that can access Exalate. It is possible to perform during the installation of the app or in the Apps and Integrations settings.



### App details

Version: 2.2  
Framework Version: 2.0  
Installed: March 16, 2021  
Email: [support@exalate.com](mailto:support@exalate.com)  
Location: Main Navigation, Ticket

### INSTALLATION

Title:

**Enable role restrictions?**

Select the roles that should have access to this app:

**Enable group restrictions?**

Select which groups should have access to this app:

By installing this app you hereby agree to the [Zendesk Marketplace Terms of Use](#).

### ON THIS PAGE

[Azure DevOps](#)

[GitHub](#)

[HP ALM/QC](#)

[Jira Cloud](#)

[Product](#)

[Jira on-premise](#)

[About Us](#)

[Salesforce](#)

[Release History](#)

[ServiceNow](#)

[API Reference](#)

[Zendesk Security](#)

[Pricing and Licensing](#)

#### Resources

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

#### Still need help?

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)