

What is the Difference between Exalate for Jira Cloud & Jira On-premise?

Last Modified on 03/04/2024 10:56 am EST

Most of the functionality provided in Exalate for Jira on-premise is also available in Exalate for Jira Cloud.

This article describes the major differences between the Cloud and the on-premise app.

Licensing

Exalate App for Jira On-premise

Jira on-premise app licenses are perpetual and the purchase price includes 12 months of maintenance (support and version updates).

Exalate App for Jira Cloud

Jira Cloud is a monthly subscription-based license. Therefore, you are automatically billed for apps together with host products.

For more information check the [Exalate pricing and licensing details](#).

Hosting and Data Storage

Exalate App for Jira On-premise

You host the app on-premise.

Jira on-premise URL is also the Exalate URL after the app installation.

Exalate App for Jira Cloud

We are hosting the Exalate app for Jira Cloud on our own servers. Check [DATA SECURITY AND PRIVACY STATEMENT](#) for more details.

It means that you have 2 separate servers: one where you are running the Jira Cloud instance itself and another one with the Exalate app on it.

The Exalate app has its own Exalate URL, which is different from the Jira Cloud instance URL.

There is a possibility of connectivity delays between the Jira Cloud and the Exalate app. The physical location of the servers that host the app and the instance might affect the connectivity performance.

Backups and Updates

Exalate app for Jira Cloud

As far as the Atlassian Jira Cloud is a hosted service, the Exalate app for Jira Cloud is upgraded

automatically.

Atlassian automatically detects updates to Atlassian Connect apps with a polling service and updates your app. [More details about automatic updates.](#)

We perform a backup every hour on the Exalate servers to avoid data loss.

If a customer unsubscribes from our Cloud app we mark stored Customer Data, for deletion. The data is deleted after 180 days at the latest if the customer does not re-subscribe. However, the customer can [contact us](#) to ask for an earlier deletion.

Exalate app for Jira On-premise

Since the app is hosted on-premise, you decide how often to back up data.

Monitoring

Exalate app for Jira On-premise

As for the Exalate on-premise apps, monitoring could be done only by the Jira on-premise admin.

Exalate app for Jira Cloud

Since the Exalate cloud app is hosted on our own servers, we do system health monitoring for the app.

Connection Type

Exalate app for Jira Cloud

Jira Cloud Instance could not be set as a private instance.

Check [how the traffic is different between private and public Jira instances.](#)

Note: When you set up a Connection with a Jira on-premise instance that is behind the firewall, you should use the **Exalate URL** as far as it's different from the Atlassian Jira Cloud instance URL.

Scripting

Most of the functionality provided in the Exalate app for Jira on-premise is also available in the Exalate app for Jira Cloud, but there are limitations because not all APIs are available for Jira Cloud.

Below you can see a list of the most common differences in [Groovy scripting](#):

- APIs:
 - Cloud: [REST API](#)
 - On-premise:

- [REST API](#)
 - [JAVA API](#)
- Exalate API for Jira Cloud is different from the Exalate API for Jira on-premise
- Exalate runs on different Java versions:
 - Server - depends on your Jira version
 - Cloud - is determined by Exalate, since we host cloud apps on our own servers
- The app architecture is different for Cloud and on-premise apps: this affects advanced scripting, which would be different depending on the platform
- Some of the [script helpers](#) are platform specific:

Jira On-premise	Jira Cloud
<ul style="list-style-type: none"> ◦ attachmentHelper -- The attachmentHelper provides functions to manipulate attachments. ◦ commentHelper -- The commentHelper provides functions to manipulate comments. ◦ debugHelper -- This method helps to raise an IssueTrackerException error with the value of the field which is not syncing. ◦ nodeHelper -- The nodeHelper is an object which can be used in the Sync Rules scripts to get specific information from the underlying tracker. ◦ syncHelper -- The syncHelper is an object which can be used in the Sync Rules scripts to schedule sync on your instance. ◦ userHelper -- The userHelper provides functions to manipulate users. ◦ workflowHelper -- The workflowHelper provides functions to transition issues. If the transition does not have any conditions and validators, its name can be used to perform the transition. 	<ul style="list-style-type: none"> ◦ attachmentHelper -- The attachmentHelper provides functions to manipulate attachments. ◦ commentHelper -- The commentHelper provides functions to manipulate comments. ◦ debugHelper -- This method helps to raise an IssueTrackerException error with the value of the field which is not syncing. ◦ nodeHelper -- The nodeHelper is an object which can be used in the Sync Rules scripts to get specific information from the underlying tracker. ◦ syncHelper -- The syncHelper is an object which can be used in the Sync Rules scripts to schedule sync on your instance. ◦ workflowHelper -- The workflowHelper provides functions to transition issues. If the transition does not have any conditions and validators, its name can be used to perform the transition. ◦ workLogHelper -- The workLogHelper provides functions to manipulate worklogs.

Follow the guide to [set up your first synchronization](#).

ON THIS PAGE

[Licensing](#)

[Hosting and Data Storage](#)

[Backups and Updates](#)

Product

[Monitoring](#)

[Release History](#)

[Connection Type](#)

[Glossary](#)

[API Reference](#)

[Security](#)

[Pricing and Licensing](#)

Resources

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

Still need help?

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)