How to Sync Tempo Worklogs using External Scripts in Jira On-premise

This article shows how to synchronize Tempo worklogs with external scripts.

> **Disclaimer**
>
> - There is no guarantee on its stability, as it didn't run through a complete validation cycle.
> - The feature can evolve and even be retracted in case it doesn't meet our quality criteria.
> - Please send feedback on our experimental features to support@exalate.com.

Introduction

With Exalate you can sync Tempo worklogs.

Synchronization of Tempo worklogs requires an exchange of the following data:

- worklog author
- issue
- description
- date: created at, update at, start date, start time,
- worked (time spent)
- remaining estimate
- tempo worklog id (only for Jira Cloud)
- Jira worklog id (if the remote side does not have tempo app, tempo worklogs would be synced into a Jira worklog fields)

> Exalate supports only create worklog events synchronization. If the worklog has been updated or deleted these changes would not be synchronized.

Tempo stores all worklogs in the default Jira worklog fields. Therefore you don't need to use any external script to synchronize tempo worklogs on Jira Server.

Jira worklogs synchronization requires worklog author lookup. By default, Exalate tries to find the worklog author on the remote side and create a worklog on behalf of that author.

The author lookup is performed by email, but you can also configure other ways to find corresponding users on the local system (by full name, username or any other arbitrary logic that you might have).

Below you can find the example of the simple worklog synchronization on Jira Server.

Source side (Jira Server)

To send the data from tempo add the code below to the **Outgoing sync(data filter).**

```
replica.workLogs = issue.workLogs
replica.remainingEstimate = issue.remainingEstimate
```

Destination side (Jira Server)

To apply received tempo worklogs locally add the code below to the **Incoming sync for new issues(create/change processor).**

```
issue.workLogs = workLogHelper.mergeWorkLogs(issue, replica)
```

Additionally, you can configure user mapping in the *Incoming sync* processors or set a default worklog author in case the original author is not found.

For more details, please read workLogs synchronization on Jira On-premise.