How to Sync Ranks in Jira On-premise

Last Modified on 04/09/2024 9:25 am EDT

Introduction

You're setting up synchronization between 2 Agile boards and you're wondering how you can synchronize ranking. With ranking, you can reprioritize stories according to the need of the team. Stories higher in the list are considered to be of a higher priority than the ones lower in the list. One of the kicker features of Jira Software is the fact that you can rank stories by dragging and dropping. Jira implements this ranking using a global rank field of type LexoRank. For more information, pleae read this StackOverflow description.

Combining Ranking and Synching is a challenge.

Ranking and Syncing - a Master / Slave Relation

Whenever you consider how ranking should work taking into account a synchronized environment. Assume that Team A is prioritizing stories 1, 2, 3, 4 while Team B is prioritizing stories 4, 2, 3, 1 you get a conflict that needs to be resolved by talking to each other. It is not something that a tool can do. The solution proposed by Exalate is to have one side of a connection be the Ranking Master and the Ranking Slave. All prioritizing is done at the master, and the slave reflects the decisions made at the master. A typical situation might be that the product management team ranks the stories while the development team implements it in that way.

Setting up the Environment

Given 2 Jira instances

- **PM Jira** which is used by the product management team
- **Team Jira** which is used by the development team

On PM Jira a board is used to manage the sprints and epics, and the ranking of the stories. On Team Jira a board is used to keep track of the implementation of the stories. The PM Jira is considered as the ranking master, while the team Jira as the ranking slave. The ranking information flows from the master to the slave using the following settings.

Source side

Outgoing sync

```
/*
** Copy the ranking value from the issue to a custom key
** Rank is the standard ranking custom field.
*/

replica.customKeys."RankValue" = issue.customFields."Rank"?.value?.value
```
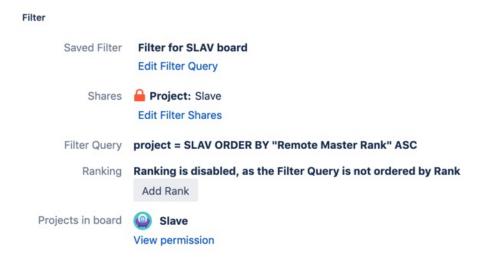
Destination side

- First set up a custom field of type 'text - single line' with a name like 'Remote Master Rank'
- Configure in the change/create processor following lines

Incoming sync

```
// Copy the customKeys field 'RankValue' to the text field 'Remote Master Rank'
issue.customFields."Remote Master Rank".value = replica.customKeys."RankValue"
```

- On the board which reflects the stories which are sent in, change the filter query to order by this 'Remote Master Rank' field

Impact of the Setup

- By using this approach, the slave board shows the stories in the priority as determined on the master
- Another board can still be used to prioritize the stories on the slave. The slave board can still be used to see how the priority is seen by the product management team
- Nothing stops to implement this bi-directionally, such that the team view on the prioritization can be compared to the prioritization of the product management team

The movie here shows the interaction between the master Jira (Blue), with the team Jira (Red). Changing the story MAS-13 is reflected on the red Jira.

See it in Action!

ON THIS PAGE