kgish / **assembla-to-jira-migration-public**

Branch: **master** ▾ **assembla-to-jira-migration-public** / **README.md** **Find file** **Copy path**

kgish Fixed support links. e08c082 27 minutes ago

1 contributor

167 lines (110 sloc) 7.68 KB

Assembla to Jira Migration

A collection of advanced tooling which provides seamless data migration from Assembla to Jira.

This is by far the best Assembla to Jira migration toolset around. Here are some of the reasons why this toolset beats out all others hands down:

- Fully automated requiring minimal manual actions.
- Configuration file with various options.
- Import users (names, emails, roles and permissions).
- Import tickets, comments, attachments and relationships.
- Retain ticket status, labels, ranking and custom fields.
- Link back to original Assembla tickets for reference.
- Save relevant Assembla context in user-defined fields.
- Embed image thumbnails in descriptions and comments.
- Convert markdown, users, urls, attachments and other links.
- Timetracking: estimated and remaining.
- Retain watchers of tickets for notifications.
- Create scrum or kanban board with workflow.
- Map the Assembla milestones to Jira sprints.
- Populate the backlog, future and current sprints.
- Assign stories to epics.
- Resolve cross linking between external projects.
- Account for the API differences between hosted and cloud.
- Tons of great documentation and trouble-shooting guide.

If you need extra help just look here.

Introduction

Have you ever wanted to use Jira instead of Assembla, but were afraid that the switch to Jira was too risky?

Are you worried that your business-critical data in Assembla will get corrupted or even lost during the conversion?

Jira already offers a number of standard add-ons to make certain migrations easier. Unfortunately, it does not yet offer a tool for migrating from Assembla (bummer).

However, **you are now in luck!** By using this Assembla-to-Jira migration toolset, it is very easy to export all of the relevant Assembla data and import most (if not all) of it into a Jira project without loss or corruption of data.

By using the Assembla API and the Jira API together, both environments are hooked up in order to make all necessary data transformations run smoothly and automatically.

Most of the actions are done automatically via a pipeline of scripts. Just define the required parameters in the `.env` configuration file, and you are ready to go.

Some manual actions are required since the Jira API does not support all of the required data transformations, however these actions are few and clearly documented below. It is very important NOT to skip these manual actions because a successful migration depends on them being done properly at the right time.

While not required, it is still best to start with a fresh installation, e.g. one in which the desired project has not yet been created and none of the project users are present yet. Otherwise, unexpected problems might occur.

At the beginning of the migration, a quick scan is made to detect if a given project already exists. If not then the new project is created for you, as well as all of the Assembla users who have not yet been created.

Although the official Atlassian documentation states that the Jira API for hosted server is nearly identical to the cloud server, there are some subtle, tricky differences that can bite you when you least expect.

Don't worry, if you read all of the instructions below, follow them carefully without skipping anything, you can avoid these annoying bumps in the road on your way to a successful migration.

Need help? Look here.

Installation

The toolset has been written with the Ruby programming language. In order to be able to use it, you will have to have downloaded and installed the following items on your computer:

- Ruby
- Bundler
- Git

Ensure that you have the correct version of ruby installed and set to using it for scripts.

```
$ rvm install `cat .ruby-version`
$ rvm use `cat .ruby-version`
```

Once this has been done, you can checkout and install the toolset from the github repository.

```
$ git clone https://github.com/kgish/assembla-to-jira-migration.git assembla-to-jira
$ cd assembla-to-jira
$ gem install bundler
$ bundle install
```

At this point everything should be ready to use as explained in the following sections.

Pipeline Steps

The complete migration from start to finish consists of a series of scripts which are to be executed in order.

Like a pipeline, each script processes data and generates a dump file to store the intermediate results. This output is used in turn as input for the following script.

The reason for doing this that if something goes wrong you do not lose everything and can restart from the previous step.

Each step will generate a log of the results in the form of a csv file for reference purposes, e.g. detecting which requests failed and why. For example, importing tickets will create the `data/jira/:space/jira-tickets.csv` file where `:space` is the Assembla space name `ASSEMBLA_SPACE` in the `.env` file hyphenated with the `hostname-port`. For example: `space-name-jira-example-org-8080`.

While the script is being executed, information will be logged to the console. Be sure to inspect the information, as certain instruction might be given that you must follow before continuing to the next step.

Assembla export

First we export all the data from Assembla.

1. Space (spaces, space_tools, users, user roles, tags, milestones, ticket statuses, ticket custom fields, documents, wiki pages and tickets)
2. Tickets (comments, attachments, tags, associations)
3. Report users
4. Report tickets

Jira import

Now that all of the Assembla data is available, we can now take this and import it into Jira.

5. Create project (and board)
6. Create issue link types
7. Get settings (issue types, priorities, resolutions, roles, statuses and projects)
8. Import users
9. Download ticket attachments
10. Create custom fields
11. Import tickets
12. Resolve/update ticket links
13. Import ticket comments
14. Import ticket attachments & update links
15. Update ticket status (resolutions)
16. Update ticket associations
17. Update ticket watchers
18. Resolve/update ticket and comment external links
19. Move stories to epics
20. Rank tickets (cloud only)

Scrum/Kanban board

Using the Agile extension, create the sprints and populate the scrum/kanban board.

21. Create sprints
22. Update board

Cleanup

23. Manual cleanup actions

Congratulations, you did it!

References

- Assembla

    - Homepage
    - API Reference
    - Markdown

- Jira

    - Homepage
    - JIRA Server platform REST API reference
    - JIRA Cloud REST API Reference
    - Markdown
    - Upgrade to Atlassian Account

Support

Do you require assistance with the migration or need some new functionality that is not yet part of this package?

No worries, I can certainly help you out.

Feel free to contact me!

Author

Kiffin Gish < kiffin.gish@planet.nl >

Gishtech => Advanced Software Development for the Web

"You're never too old to learn new stuff..."