

Review of JIRA Integration Micro Service

Usha SR

Student, Department of ECERV College of
Engineering Bangalore-59, India

Dr. Chinamy R

Asst. Professor, Department of ECERV College
of Engineering Bangalore-59, India

Abstract:- In many organizations, their applications put forth action items for users, and the efforts put into it will not be tracked. Certainly, a team needs all action items to be put in a single place for proper planning. And not always the dialogue will be open between the action item creator and the person who work upon it. Nowadays, more than 65,000 companies around the world rely on Jira Software to manage their projects. Jira is known for its flexibility to support any type of project and extensibility to work with thousands of apps and integrations. Jira integration is a microservice that can be used by the existing applications as per their requirements. This microservice injects most of the benefits of Jira Software into applications.

Keywords:- *Microservice, action-item.*

I. INTRODUCTION

Jira Software helps software teams plan, track, and release great software. It's a hosted service from Atlassian for agile software development, with features sprint planning, scrum and kanban boards, agile reports, dashboards, and more. There are over a thousand Jira-related plugins in the Atlassian marketplace. The opportunities for extending Jira Software are endless!

Not all applications are perfect, hence collaboration of software is essential for a better user experience. For example: Adding new features (like notification service) in applications will be time-consuming and sometimes unnecessary, hence integration with Jira software could be used as a temporary or easy solution, as it sends out an email to the assignee of the action item. Also opens a new communication channel between the actor and watcher of action item.

In this study, Jira Software was analyzed to resolve draw-backs of existing applications used by colleges, by integrating applications with Jira software via microservice created.

Potential areas where Jira integration is needed are:

- When the application doesn't manage the status and the efforts put on action items produced.
- When applications need the statistics based on the status of the action item.
- When applications don't provide the notification for the person/team who has to work on the action item.
- When the application lacks to segregate the action items based on the teams in the organization.
- When an application doesn't manage the linkage between action items.

II. JIRA SOFTWARE FEATURES

Jira Software provides many features, a few of them are listed below which could be the base for microservice creation:

- Logging of all the efforts, changes, and transitions made on each Jira ticket.
- Option to create a new Workflow with different statuses as per the need.
- Ease in configuring the filters as per requirement. E.g. Filter to get all the Jira tickets of a particular team's project.
- Ability to link Jira tickets and represent in dashboard.
 - Creation of Jira Ticket
 - Modification of Jira ticket
 - Transitioning Jira ticket
 - Update the action item in the application

III. FEATURES OF MICROSERVICE

Understanding the key features of Jira software and potential areas where integration is needed, Jira Integration microservice can be created with basic features listed below:

A. Creation of Jira Ticket

For every action item, there can be a corresponding Jira ticket to track the status of work. Any creation of Jira ticket needs basic field values, they are project Key, issue type, summary, etc. Based on the issue type chosen the list of required fields and workflow will be different. Fields that may be utilized: "projectKey", "issuetype", "assignee", "components", "priority", "reporter", "watchers", "epic link" etc.

Every team in the organization will have a different Jira project and hence project key will be unique. So when teams need all the action items to be put in a single place then the project key will be significant. And the team lead can assign the Jira ticket to any team member and a suitable notification will be sent by Jira software.

B. Modification of Jira ticket

To have accountability between the action item and the Jira ticket, changes made on the action item should be reflected on the Jira ticket. Fields in the Jira ticket will be modified when the corresponding fields in the action item are changed. Project key or issue type need not have to be updated, but other Jira fields can be updated.

C. Transitioning Jira ticket

Each Jira ticket will have a workflow based on the issue type. Workflow can be configured based on the issue type as per the user's needs. Based on the progress on the action item suitably the Jira status can be changed via REST API. For example, a Jira ticket can be transitioned from "In progress" to "Done" when an action item is completed.

D. Update action item in application

To have sync between the action item and the Jira ticket, changes made on the Jira should be reflected on the action item. For this webhooks are provided by Jira for backward communication

IV. RELATED WORK

In this section will mention how the Jira software is being integration to support other applications, and it also being used as source of truth in few case studies.

Ortu et al. [1] analysed the developer networks of 7 open source projects hosted in JIRA. They found open source is not an child of lesser God. They found nearly 60 percent of comments was from other team members. They analysed the average community issue fixing time and found it varies across the communities. And showed the independence of the average issue resolution time from the other factor considered, such as the community size and the kind of issues maintenance and priority.

Kaur, A et al. [2] developed a bug reporting collection system which can be possibly employed for the collection of bugs from the open source Jira bug tracking system (for Apache projects) and the reports related to all the attributes of bugs are generated. The gathered information was used for various purposes which includes applications like: classification of bugs based on one-line and long description and accordingly bugs can be classified as concurrency, security or semantic bugs; prediction of severity of bugs using machine learning algorithms. The performance was evaluated using various evaluation metrics such as precision, recall and accuracy.

Luders. C et al. [4] in this paper they had developed visualization tool which is integrated with JIRA to get the details of issues. Tool gives the graph as output indicating the issues linkage. With the Issue Link Map visualization users were able to see all linked issues at a glance.

Pooja Mital et al. [6] they discussed about plugins of Jira Automation Tool to enhance the basic functionality. They proved that by opting the best of these plugins would not only bridge the gap between development and operations teams but also escalate and optimize the process of software development. An impression of using other plugins at different phases of software development may also be carried out and analyzed as a part of subsequent research. Their research or analysis was useful for the researchers to interpret the praxis of DevOps accompanying its existing plugins compatibility.

Cabera Diego et al. [8] this paper discusses about Emotion classification which is the task of determining emotions that are expressed in non-neutral text. The task is complex because text can express multiple emotions. The state-of-heart lacks freely available annotated data and external resources and only offers few classification tools, especially in the domain of Open Source Software (OSS). They explored two multi-label classifiers, i.e. HOMER and RAKEL, and lexica, i.e. SenticNet 5, the NRC Word-Emotion Association Lexicon and the NRC Affect Intensity Lexicon,

to develop an emotion classifier for text related to OSS. The classifiers were evaluated on collections of JIRA Issue Tracker comments and Stack Overflow posts. They evaluated against EmoTxt, a state-of-the-art emotion classifier for OSS related text, a random baseline and the most frequent class one. They used multi-label and single-label metrics, as well as statistical significance testing.

M. I. Lunesu et al. [11] In this paper they presented a risk assessment procedure based on process modeling and simulation, and tested it on three open source JIRA project developed using an agile approach, with requirements collected as user stories and managed as atomic unit of work, using JIRA popular issue management tool. The process was organized into a sequence of basic activities, and can thus be modeled using an event-based simulation. The developers are in turn modeled as agents, who decide the units of work they develop and complete. To be able to work with real data, they linked the simulator with JIRA to collect the needed information (used process, team composition, project size, number of issues, estimated effort) and show the reliability of the tool.

V. IMPLEMENTATION OF JIRA INTEGRATION

Top level system designed for integrating the microservice is shown in Fig.1. Few endpoints related to features of microservice will be exposed and database will be set up to persist the mapping between the action item and Jira ticket created.

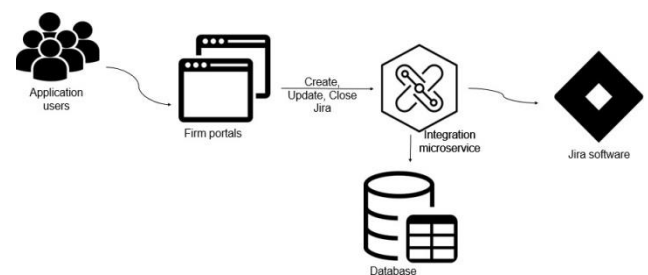


Fig. 1: Jira Integration Microservice

Endpoints exposed are:

- To get list of projects across all jira instances Description: This to provide option for the user to select the project of any Jira instance which microservice supports.
- Request url: /jira/project Request method: GET Request response:
 - Status 200 returned with response : [{ "key": "SBP", "name": "Spring boot project"},...]
 - Status 500 returned if any exception occurs.
- To get the list of issue type in jira project. Description: This endpoint facilitates user to select the issue type suitable for action item. Request url: /jira/project/ projectKey /issuetype Request method: GET Request response:
 - Status 200 returned with response : ["id": "10000", "name": "Story", },...]
 - Status 500 returned if any exception occurs.

1) To create jira ticket

Description: This endpoint can be used for automatic or manual creation of Jira ticket from an application. Request url: /jira/issue

Request method: POST

```
Request body: { "entityId":
"1123", "entityType": "RISK",
"projectKey": "SBP",
"issueTypeId": "10000",
"assignee": "usha",
"summary": "Title of Risk",
"description": "Risk description",
"duedate": "2022-11-29"
}
```

Request response:

- Status 200 returned with response :

```
{
  "id": "1432",
  "key": "SBP-11",
  "self": "http://host:port/browse/SBP-11"
}
```
- Status 500 returned if any exception occurs.

The request body received will be validated before creating the payload suitable for Jira software.

- To update fields of Jira ticket
- To transition Jira ticket to desired status Request url: /jira/issue/status

Request method: POST

Request Body: "entityId": "1123", "entityType": "RISK", "toStatus": "Done"

```
}
```

Request response:

 - Status 200 returned with response: "SUCCESS"
 - Status 500 returned if any exception occurs.

The request body received will be modified before creating the payload suitable for Jira software.

VI. CONCLUSION

In this paper, we intended to provide the information that will be useful while creating the Jira Integration microservice. Referring to this paper, the readers will get to know the actions that need to be taken and complete work accordingly. [9] this documentation will be useful to get the Jira software endpoints and suitably new features can be added to microservice as per the organization's need.

REFERENCES

- [1.] Ortu, Marco, et al. "Measuring and understanding the effectiveness of jira developers communities" IEEE/ACM 6th International Workshop on Emerging Trends in Software Metrics
- [2.] Kaur, Arvinder, and Shubhra Goyal Jindal. "Bug report collection system (BRCS)." 2017 7th International Conference on Cloud Computing, Data Science Engineering-Confluence. IEEE, 2017.
- [3.] Choetkiertikul, Morakot. "Developing analytics models for software project management" International Conference on Engineering Management of

Communication and Technology

- [4.] Lu'ders, C. M., Raatikainen, M., Motger, J., Maalej, W "Openreq issue link map: A tool to visualize issue links in jira" IEEE 27th International Requirements Engineering Conference (RE) (pp. 492-493).
- [5.] <https://docs.gitlab.com/ee/integration/jira/>
- [6.] Mittal, Pooja, and Poonam Mehta "Optimization of Software Development Process by Plugin Integration with Jira—A Project Management Tool in Devops", Proceedings of the International Conference on Innovative Computing Communications (ICICC)
- [7.] ALDAN, C, ., Demir, E. (2020, October) "Detection of Duplicate Bug Reports in Jira and Bugzilla Tools" In 2020 Turkish National Software Engineering Symposium (UYMS) (pp. 1-4). IEEE. ALDAN, C, ., Demir, Description: This endpoint can be used for automatic or manual modification of Jira ticket from an application. Request url: /jira/issue
- [8.] Request method: PUT Request body: "entityId": "1123", "entityType": "RISK", "issueTypeId": "10000", "assignee": "usha", "summary": "Edited: Title of Risk", "description": "Edited: Risk description", "duedate": "2022-11-30"
- [9.] Request response: Status 200 returned with response: "SUCCESS"
- [10.] Status 500 returned if any exception occurs. The request body received will be validated before creating the payload suitable for Jira software.
- [11.] Cabrera-Diego, Luis Adrian, Nik Bessis, and Ioannis Korkontzelos "Classifying emotions in Stack Overflow and JIRA using a multi-label approach" Knowledge-Based Systems 195.
- [12.] <https://developer.atlassian.com/server/jira/platform/rest-apis/>
- [13.] Gorbunova, A. A., A. S. Trunov, and V. I. Voronov. "Intelligent analysis of technical support requests in service desk ticketing systems" International Conference on Engineering Management of Communication and Technology (EMCTECH). IEEE
- [14.] M. I. Lunesu, R. Tonelli, L. Marchesi and M. Marchesi, "Assessing the Risk of Software Development in Agile Methodologies Using Simulation," in IEEE Access, vol. 9, pp. 134240-134258.