JIRA PLUGIN FOR THE ALICE INSTANCE

A. Kondratyev[a], A. Bondyakov

Meshcheryakov Laboratory of Information Technologies, Joint Institute for Nuclear Research

Russia, 141980, Dubna, Moscow region, Joliot-Curie, 6

E-mail: [a] kondratyev@jinr.ru

As a bug tracking and project management system, the ALICE experiment uses Jira Software, which provides a wide range of configuration options. Jira also allows one to significantly expand the basic functionality with custom plugins. In this work, the LinkedIssuesHasStatus plugin for the JIRA service of the ALICE experiment is developed and implemented. The plugin returns tickets that link issues in the specified status and that are linked with the specified linking type.

Keywords: ALICE, Jira, Java

Andrey Kondratyev, Aleksey Bondyakov

Copyright © 2021 for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-3041/185-189-paper-34.pdf

1. Introduction

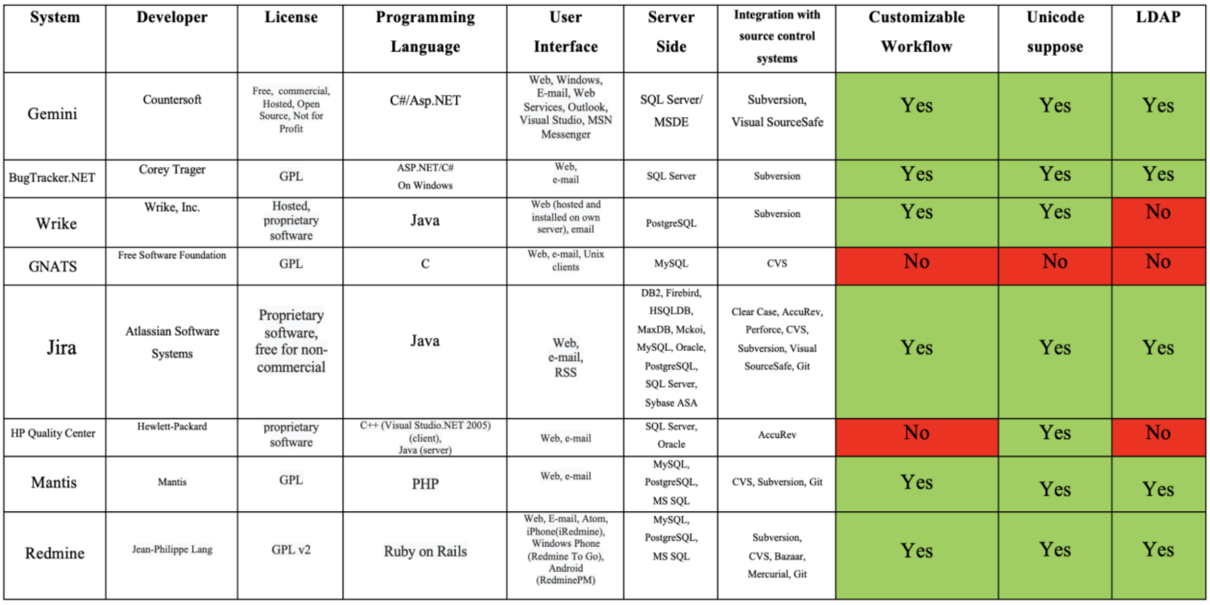
Jira [1] is based on Java [2] and runs on several popular database management systems and operating systems.

The main accounting element in the system is an issue. It contains the project name, theme, type, priority, components and content. The issue can be extended with additional fields, applications or comments. It can be edited, or one can simply change its status, for example, from "open" to "closed". Possible state transitions are determined through a custom workflow. Any changes to the issue are logged. Jira has a large number of configuration options: for each application, a separate task type can be defined with its own workflow, a set of statuses, one or several views (screens). In addition, with the help of so-called "schemes", for each individual Jira project, one can define its own access rights, behavior and visibility of fields, etc.

2. Jira features

Table 1 shows popular bug tracking and project management systems, which received updates no earlier than 2018, and their main features. All information was obtained from open sources and the Internet. As shown, the Jira server side works with all popular databases and has integration with the majority of source control systems. Jira enables workflow customization and has different user interfaces such as applications for Android [3] and iOS [4].

Table 1. Popular bug tracking and project management systems



System	Developer	License	Programming Language	User Interface	Server Side	Integration with source control systems	Customizable Workflow	Unicode suppose	LDAP
Gemini	Countersoft	Free, commercial, Hosted, Open Source, Not for Profit	C#/Asp.NET	Web, Windows, E-mail, Web Services, Outlook, Visual Studio, MSN Messenger	SQL Server/ MSDE	Subversion, Visual SourceSafe	Yes	Yes	Yes
BugTracker.NET	Corey Trager	GPL	ASP.NET/C# On Windows	Web, e-mail	SQL Server	Subversion	Yes	Yes	Yes
Wrike	Wrike, Inc.	Hosted, proprietary software	Java	Web (hosted and installed on own server), email	PostgreSQL	Subversion	Yes	Yes	No
GNATS	Free Software Foundation	GPL	C	Web, e-mail, Unix clients	MySQL	CVS	No	No	No
Jira	Atlassian Software Systems	Proprietary software, free for non-commercial	Java	Web, e-mail, RSS	DB2, Firebird, HSQLDB, MaxDB, Mckoi, MySQL, Oracle, PostgreSQL, SQL Server, Sybase ASA	Clear Case, AccuRev, Perforce, CVS, Subversion, Visual SourceSafe, Git	Yes	Yes	Yes
HP Quality Center	Hewlett-Packard	proprietary software	C++ (Visual Studio.NET 2005) (client), Java (server)	Web, e-mail	SQL Server, Oracle	AccuRev	No	Yes	No
Mantis	Mantis	GPL	PHP	Web, e-mail	MySQL, PostgreSQL, MS SQL	CVS, Subversion, Git	Yes	Yes	Yes
Redmine	Jean-Philippe Lang	GPL v2	Ruby on Rails	Web, E-mail, Atom, iPhone(iRedmine), Windows Phone (Redmine To Go), Android (RedminePM)	MySQL, PostgreSQL, MS SQL	Subversion, CVS, Bazaar, Mercurial, Git	Yes	Yes	Yes

3. ALICE Jira instances

The ALICE experiment [5, 6, 7] has two Jira instances, i.e. *develop* [8] and *production* [9]. The *develop* instance is used to test new features, versions and plugins. The *production* instance is for general use. Both instances have the following system specifications:

- Jira v.8.17;
- CentOS 7;
- PostgreSQL.

According to statistics obtained in July 2021, the ALICE Jira *production* instance has:

- 4,350 registered users;
- 106 projects in 15 categories;
- 32,354 issues;
- 68 statuses and 51 issue types;

- more than 20 custom plugins installed;
- API for user and e-group management.

Due to the large number of users, projects and links between issues, the LinkedIssuesHasStatus plugin was developed and implemented to make working with issues more convenient, faster and more efficient. The plugin returns tickets that link issues in the specified status and that are linked with the specified linking type.

4. Jira plugins and JQL functions

Jira plugins are also known as Jira apps or Jira add-ons. These are pieces of software that plug into the Jira task management platform and provide new and enhanced functionality.

There are two ways to install plugins into the Jira instance:

- Install the plugin using the Jira marketplace;
- Upload the custom plugin yourself.

JQL functions are among the extension points that JQL (Jira Quick Language) provides to developers. Functions allow calculating values at runtime within a JQL query. They are easy to write and can be surprisingly powerful.

5. Plugin scheme

Figure 1 illustrates the LinkedIssuesHasStatus plugin scheme implemented on the ALICE Jira *develop* instance.

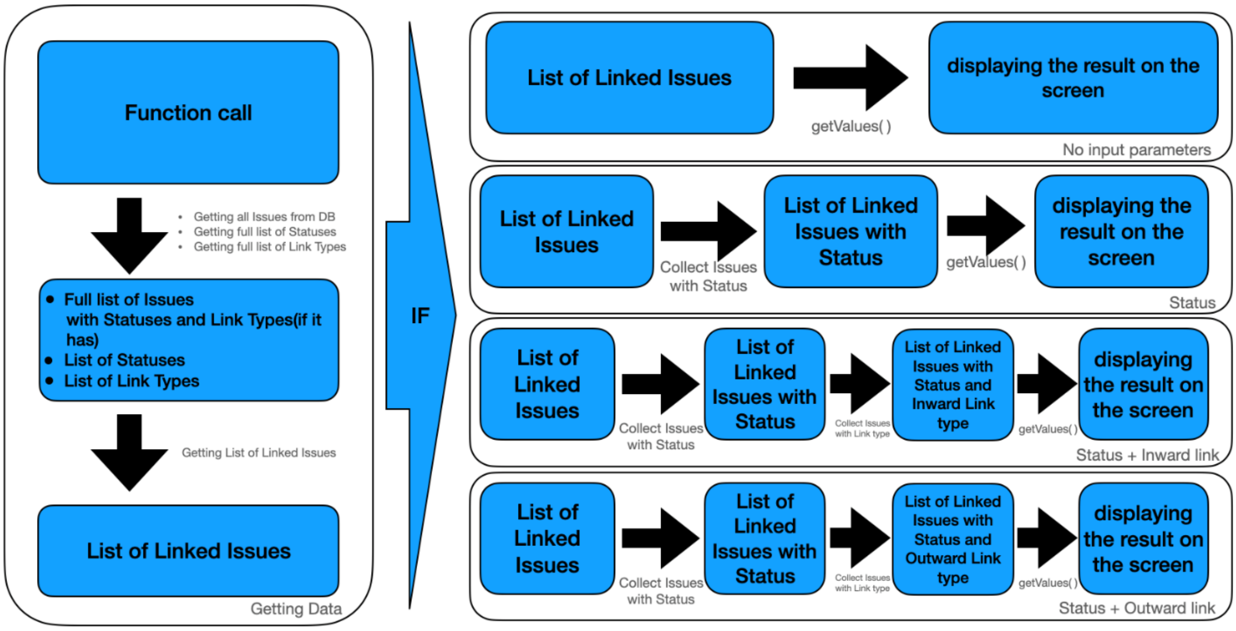

Function call
Getting all Issues from DB
Getting full list of Statuses
Getting full list of Link Types
Full list of Issues with Statuses and Link Types(if it has)
List of Statuses
List of Link Types
Getting List of Linked Issues
List of Linked Issues
Getting Data
IF
List of Linked Issues
getValues( )
displaying the result on the screen
No input parameters
List of Linked Issues
Collect Issues with Status
List of Linked Issues with Status
getValues( )
displaying the result on the screen
Status
List of Linked Issues
Collect Issues with Status
List of Linked Issues with Status
Collect Issues with Link type
List of Linked Issues with Status and Inward Link type
getValues( )
displaying the result on the screen
Status + Inward link
List of Linked Issues
Collect Issues with Status
List of Linked Issues with Status
Collect Issues with Link type
List of Linked Issues with Status and Outward Link type
getValues( )
displaying the result on the screen
Status + Outward link

Figure 1. LinkedIssuesHasStatus plugin scheme

After calling a JQL function, a full list of issues, a list of statuses and a list of issue link types are generated. Lists of statuses and link types are created since each Jira instance can have its own custom statuses and link types. It is possible to come up with the status and link type as one wants.

A list of linked issues is then generated.

After that, the plugin can work in four ways:

- The function is called without input parameters.

Using a special Java method getValues(), a list of linked issues obtained earlier is displayed on the screen.

- The function is called with a status as an input parameter.

Issues with a status called in the JQL function as an input parameter are collected from the list of linked issues obtained earlier. Then, using the getValues() method described above, a list of linked issues that have the requested status is displayed on the screen.

- It is possible to call a JQL function with more than one status as input parameters. The output will show a list of linked issues that have a status declared as the first parameter and a link to the issues with a status declared as the second parameter.

For example, three statuses "Waiting", "Closed" and "Open" are declared as the first, second and third input parameters. As an output, issues with the "Waiting" status that have a link to the issues with the "Closed" status and issues with the "Waiting" status that have a link to the issues with the "Open" status are shown.

There are no restrictions on the number of statuses as input parameters.

- The function is called with a status and an inward link type as input parameters.

After all statuses as input parameters it is possible to put an issue link type.

In Jira, links to issues have 6 default types. Custom types can also be created. These types have inward or outward descriptions.

The inward description denotes a description of how the issue is affected by other issues.

In this case, issues with a status called in the JQL function as an input parameter are collected from the list of linked issues obtained earlier. After that, the link type from the input is compared with the list of link types. Issues with an inward link type are collected from the list of issues with the requested status. Then, using the getValues() method described above, a list of linked issues that have the requested status and requested inward link type is displayed on the screen.

- It is possible to call a JQL function with more than one status as input parameters. The link type should be the last parameter. The output will show a list of linked issues that have a status declared as the first parameter, a link to the issues with a status declared as the second parameter and an issue link type requested as the last parameter.

- The function is called with a status and an outward link type as input parameters.

Another description for issue link types is the outward description, i.e. a description of how the issue affects other issues.

In this case, issues with a status called in the JQL function as an input parameter are collected from the list of linked issues obtained earlier. After that, the link type from the input is compared with the list of link types. Issues with an outward link type are collected from the list of issues with the requested status. Then, using the getValues() method described above, a list of linked issues that have the requested status and requested outward link type is displayed on the screen.

- It is possible to call a JQL function with more than one status as input parameters. The link type should be the last parameter. The output will show a list of linked issues that have a status declared as the first parameter, a link to the issues with a status declared as the second parameter and an issue link type requested as the last parameter.

6. Results and future plans

As a result, the LinkedIssuesHasStatus plugin was developed, implemented and tested on the ALICE Jira *develop* instance.

Based on the logs, the following results were obtained:

- The developed plugin is workable;
- No errors were found while the plugin was running.

Future plans:

- Integrate the plugin into the *production* instance;
- Update the plugin regularly;
- Identify and correct errors.

7. References

[1] Atlassian Jira main page // URL: https://www.atlassian.com/ru/software/jira (accessed 01.07.2021)

[2] Java main page // URL: https://www.java.com/ (accessed 01.07.2021)

[3] Android main page // URL: https://www.android.com/intl/ru_ru/ (accessed 01.07.2021)

[4] iOS main page // URL: https://www.apple.com/ru/ios/ (accessed 01.07.2021)

[5] ALICE Collaboration, F. Carminati, P. Foka, P. Giubellino, A. Morsch, G. Paic, J-P Revol et al., *ALICE: Physics Performance Report – Vol. 1* // J. Phys. G: Nucl. Part. Phys. - №30 – 2004 - PII: S0954-3899(04)83684-3

[6] ALICE Technical Design Report: Upgrade of the Online – Offline computing system ,J. Adam et al. // Tech. rep. CERN-LHCC-2015-006/ALICE-TDR-019 - 2015

[7] Interaction technology of the jAliEn client and ALICE central services, Korenkov V.V., Kondratyev A.O., Bondyakov A.S.// *Modern Information Technologies and IT Education,* Vol. 3, ISSN:2411-1473 - 2019, in Russian

[8] ALICE Jira *develop* instance // URL: https://dev2.its.cern.ch/ (accessed 01.07.2021)

[9] ALICE Jira *production* instance // URL: https://alice.its.cern.ch/ (accessed 01.07.2021)