

IssueSyncTool

v. 0.0.1

Tran Duy Ngoan

01.11.2024

Contents

1	Introduction	1
2	Description	2
3	rtc_client.py	3
3.1	Function: get_xml_tree	3
3.1.1	Method: login	3
3.1.2	Method: update_workitem	3
3.1.3	Method: create_workitem	3
4	sync_issue.py	4
4.1	Function: process_cli_argument	4
4.2	Function: process_configuration	4
4.3	Function: sync_data	4
4.4	Function: SyncIssue	4
4.5	Class: Logger	4
4.5.1	Method: config	4
4.5.2	Method: log	5
4.5.3	Method: log_warning	5
4.5.4	Method: log_error	5
5	tracker.py	7
5.1	Class: Ticket	7
5.1.1	Method: is_synced_issue	7
5.1.2	Method: connect	8
5.1.3	Method: get_ticket	8
5.1.4	Method: get_tickets	8
5.1.5	Method: create_ticket	8
5.1.6	Method: update_ticket	8
5.2	Class: GithubTracker	8
5.2.1	Method: connect	8
5.2.2	Method: get_tickets	8
5.2.3	Method: get_ticket	8
5.2.4	Method: create_ticket	8
5.2.5	Method: update_ticket	8
5.3	Class: GitlabTracker	8
5.3.1	Method: get_ticket	8
5.3.2	Method: get_tickets	8

5.3.3	Method: create_ticket	8
5.3.4	Method: update_ticket	8
5.4	Class: RTCTracker	8
5.4.1	Method: connect	9
5.4.2	Method: get_ticket	9
5.4.3	Method: get_tickets	9
5.4.4	Method: create_ticket	9
5.4.5	Method: update_ticket	9
5.5	Class: Tracker	9
5.5.1	Method: create	9
5.5.2	Method: get_support_trackers	9
6	user.py	10
6.1	Class: User	10
6.2	Class: UserManagement	10
6.2.1	Method: get_user	10
7	Appendix	11
8	History	12

Chapter 1

Introduction

(under construction)

Chapter 2

Description

(under construction)

Chapter 3

rtc_client.py

3.1 Function: get_xml_tree

Parse xml object from file. issuesynctool-rtc-client-rtcclient =====

Imported by:

```
from IssueSyncTool.rtc_client import RTCClient
```

3.1.1 Method: login

Authenticate and establish a session with RTC. issuesynctool-rtc-client-rtcclient-get-workitem -----

3.1.2 Method: update_workitem

3.1.3 Method: create_workitem

Chapter 4

sync_issue.py

4.1 Function: process_cli_argument

4.2 Function: process_configuration

4.3 Function: sync_data

Update source (original) issue due to information from appropriate destination one.

Defined sync attributes:

- ‘Title’: add issue ID as prefix e.g ‘[123] Ticket title’when creating on destination tracker
- ‘Story point’: when planning existing issue on destination tracker
- ‘Version’: when planning existing issue on destination tracker

Update destination issue due to information from source.

Defined sync attributes:

- ‘Status’: status is synced from original ticket, not allow to update directly on destination tracker

4.4 Function: SyncIssue

4.5 Class: Logger

Imported by:

```
from IssueSyncTool.sync_issue import Logger
```

Logger class for logging message.

4.5.1 Method: config

Configure Logger class.

Arguments:

- output_console
/ *Condition*: optional / *Type*: bool / *Default*: True /
Write message to console output.
- output_logfile
/ *Condition*: optional / *Type*: str / *Default*: None /
Path to log file output.

- `dryrun`
/ *Condition*: optional / *Type*: bool / *Default*: True /
If set, a prefix as 'dryrun' is added for all messages.

Returns:

(no returns)

4.5.2 Method: log

Write log message to console/file output.

Arguments:

- `msg`
/ *Condition*: optional / *Type*: str / *Default*: " /
Message which is written to output.
- `color`
/ *Condition*: optional / *Type*: str / *Default*: None /
Color style for the message.
- `indent`
/ *Condition*: optional / *Type*: int / *Default*: 0 /
Offset indent.

Returns:

(no returns)

4.5.3 Method: log_warning

Write warning message to console/file output.

Arguments:

- `msg`
/ *Condition*: required / *Type*: str /
Warning message which is written to output.
- `indent`
/ *Condition*: optional / *Type*: int / *Default*: 0 /
Offset indent.

Returns:

(no returns)

4.5.4 Method: log_error

Write error message to console/file output.

- `msg`
/ *Condition*: required / *Type*: str /
Error message which is written to output.
- `fatal_error`
/ *Condition*: optional / *Type*: bool / *Default*: False /
If set, tool will terminate after logging error message.

- indent
/ *Condition*: optional / *Type*: int / *Default*: 0 /
Offset indent.

Returns:

(no returns)

Chapter 5

tracker.py

5.1 Class: Ticket

Imported by:

```
from IssueSyncTool.tracker import Ticket
```

Normalized Ticket with required information for syncing between trackers issuesynctool-tracker-ticket-update -----

Update issue on tracker with following supported attributes:

- title
- assignee
- labels

5.1.1 Method: is_synced_issue

Verify whether the ticket is already synced or not.

It bases on the title of issue, it should contain destination ID information. E.g [1234] Title of already synced ticket
issuesynctool-tracker-trackerservice =====

Imported by:

```
from IssueSyncTool.tracker import TrackerService
```

Abstraction class of Tracker Service issuesynctool-tracker-trackerservice-connect -----

Method to connect to tracker issuesynctool-tracker-trackerservice-get-ticket -----

Method to get single ticket by id from tracker issuesynctool-tracker-trackerservice-get-tickets -----

Method to get all tickets which satisfied given condition|query issuesynctool-tracker-trackerservice-create-ticket -----

Method to create a new ticket on tracker system issuesynctool-tracker-trackerservice-exclude-issue-by-condition -----

Process to verify whether the given issue is satisfied the exclude conditions issuesynctool-tracker-jiratracker
=====

Imported by:

```
from IssueSyncTool.tracker import JiraTracker
```

5.1.2 Method: connect

5.1.3 Method: get_ticket

5.1.4 Method: get_tickets

5.1.5 Method: create_ticket

5.1.6 Method: update_ticket

5.2 Class: GithubTracker

Imported by:

```
from IssueSyncTool.tracker import GithubTracker
```

5.2.1 Method: connect

5.2.2 Method: get_tickets

5.2.3 Method: get_ticket

5.2.4 Method: create_ticket

5.2.5 Method: update_ticket

5.3 Class: GitlabTracker

Imported by:

```
from IssueSyncTool.tracker import GitlabTracker
```

Tracker client to integrate with issues on Gitlab.

Except, get_tickets which allow to get issues from gitlab, group and project level, other method requires project information to interact properly with inside issues. issuesynctool-tracker-gitlabtracker-connect -----

5.3.1 Method: get_ticket

5.3.2 Method: get_tickets

5.3.3 Method: create_ticket

5.3.4 Method: update_ticket

5.4 Class: RTCTracker

Imported by:

```
from IssueSyncTool.tracker import RTCTracker
```

5.4.1 Method: connect

5.4.2 Method: get_ticket

5.4.3 Method: get_tickets

5.4.4 Method: create_ticket

5.4.5 Method: update_ticket

5.5 Class: Tracker

Imported by:

```
from IssueSyncTool.tracker import Tracker
```

5.5.1 Method: create

5.5.2 Method: get_support_trackers

Chapter 6

user.py

6.1 Class: User

Imported by:

```
from IssueSyncTool.user import User
```

6.2 Class: UserManagement

Imported by:

```
from IssueSyncTool.user import UserManagement
```

6.2.1 Method: get_user

Chapter 7

Appendix

About this package:

Table 7.1: Package setup

Setup parameter	Value
Name	IssueSyncTool
Version	0.0.1
Date	01.11.2024
Description	Tool to synchronize issues between trackers (Github Issue, Gitlab Issue, RTC, JIRA)
Package URL	python-issue-sync-tool
Author	Tran Duy Ngoan
Email	Ngoan.TranDuy@vn.bosch.com
Language	Programming Language :: Python :: 3
License	License :: OSI Approved :: Apache Software License
OS	Operating System :: OS Independent
Python required	>=3.0
Development status	Development Status :: 4 - Beta
Intended audience	Intended Audience :: Developers
Topic	Topic :: Software Development

Chapter 8

History

0.0.1	11/2024
<i>Initial version</i>	