
jira-agile-toolbox

Jef Neefs

May 16, 2023

CONTENTS

1	API documentation	3
2	Indices and tables	7
	Index	9


```
pip install jira-agile-toolbox
```


API DOCUMENTATION

```
class jira_agile_toolbox.JiraAgileToolBox(jira_client)
```

a class which helps you do agile things with jira

Parameters

jira_client (*jira.JIRA*) – an instance of jira.JIRA

Example

```
>>> from jira import JIRA
>>> jira_client = JIRA("https://jira.atlassian.org")
>>> jat = JiraAgileToolBox(jira_client)
```

```
add_labels_to_all_sub_items_of_epic(epic, labels, keep_already_present=True, jql_query="")
```

adds labels to all ‘Issues in Epic’

Parameters

- **epic** (*str jira.Issue*) – and epic key as a string or the epic as a jira.Issue
- **labels** (*str list*) – the label to set as a string or the labels to set as a list
- **keep_already_present** (*bool*) – if this is set to False already present labels will be overwritten (defaults to True)
- **jql_query** (*str*) – a query of the form ‘project in (PROJ001,PROJ002)’ or ‘issuetype not in (‘Task’) AND status != Closed’ will be AND’ed after the autogenerated search

Example

```
>>> from jira_agile_toolbox import JiraAgileToolBox
>>> from jira import JIRA
>>> my_jira_client = JIRA("https://my-jira-server.com", basic_auth=(
↪ "MYUSERNAME", "MYPASSWORD")
>>> tb = JiraAgileToolBox(my_jira_client)
>>> tb.add_labels_to_all_sub_items_of_epic("PROJ001-001", ["label_to_set
↪ "])
```

this will append the “label_to_set” to all existing labels of all Issues in Epic

```
copy_fix_version_from_epic_to_all_items_in_epic(epic, keep_already_present=True,  
jql_query="")
```

copies fixVersions from the epic to all ‘Issues in Epic’ also applies to different projects as long as the version name is the same it works

Parameters

- **epic** (*str jira.Issue*) – and epic key as a string or the epic as a jira.Issue
- **keep_already_present** (*bool*) – if this is set to False already present fixVersions will be overwritten (defaults to True)
- **jql_query** (*str*) – a query of the form ‘project in (PROJ001,PROJ002)’ or ‘issuetype not in (‘Task’) AND status != Closed’ will be AND’ed after the autogenerated search

Example

```
>>> from jira_agile_toolbox import JiraAgileToolBox
>>> from jira import JIRA
>>> my_jira_client = JIRA("https://my-jira-server.com", basic_auth=(
    ↪ "MYUSERNAME", "MYPASSWORD")
>>> tb = JiraAgileToolBox(my_jira_client)
>>> epic = my_jira_client.issue("PROJ001-001")
>>> epic.fields.fixVersions
[<JIRA Version: name='0.0.10', id='31063'>]
>>> tb.copy_fix_version_from_epic_to_all_items_in_epic(epic)
>>> tb.get_all_issues_in_epic("JAT-001")[0].fields.fixVersions
[<JIRA Version: name='0.0.10', id='31063'>]
```

get_all_issues_in_epic(*epic, fields=None, jql_query=""*)

gets all ‘Issues in Epic’ as a list

Parameters

- **epic** (*str jira.Issue*) – and epic key as a string or the epic as a jira.Issue
- **fields** (*str list*) – a string or list of strings to limit the fields to get this helps to lower the amount of data to be sent around
- **jql_query** (*str*) – a query of the form ‘project in (PROJ001,PROJ002)’ or ‘issuetype not in (‘Task’) AND status != Closed’ will be AND’ed after the autogenerated search

Returns

a list of jira.Issues

Return type

list

Example

```
>>> from jira_agile_toolbox import JiraAgileToolBox
>>> from jira import JIRA
>>> my_jira_client = JIRA("https://my-jira-server.com", basic_auth=(
    ↪ "MYUSERNAME", "MYPASSWORD")
>>> tb = JiraAgileToolBox(my_jira_client)
>>> tb.get_all_issues_in_epic("JAT-001")
[<JIRA Issue: key='JAT-002', id='67'>, <JIRA Issue: key='JAT-003', id='68'
    ↪>, <JIRA Issue: key='JAT-004', id='69'>]
```

get_storypoints_from_epic(*epic, jql_query=""*)

searches for the epic and returns the number of storypoints as a dict

Parameters

- **epic** (*str jira.Issue*) – and epic key as a string or the epic as a jira.Issue
- **jql_query** (*str*) – a query of the form ‘project in (PROJ001,PROJ002)’ or ‘issuetype not in (‘Task’) AND status != Closed’ will be AND’ed after the autogenerated search

Returns

a dictionary containing total story points

Return type

dict

Example

```
>>> from jira_agile_toolbox import JiraAgileToolBox
>>> from jira import JIRA
>>> my_jira_client = JIRA("https://my-jira-server.com", basic_auth=(
    ↪ "MYUSERNAME", "MYPASSWORD")
>>> tb = JiraAgileToolBox(my_jira_client)
>>> tb.get_storypoints_from_epic("JAT-001")
{'total': 100, "Reported": 50, "Closed": 50}
```

rank_issues_at_top_of_project(*ranked_list*, *project*)

moves the provided *ranked_list* at the top of the backlog of the given project

Parameters

- **ranked_list** – a list of jira Issues
- **project** (*str*) – project key

Example

```
>>> from jira_agile_toolbox import JiraAgileToolBox
>>> from jira import JIRA
>>> my_jira_client = JIRA("https://my-jira-server.com", basic_auth=(
    ↪ "MYUSERNAME", "MYPASSWORD")
>>> tb = JiraAgileToolBox(my_jira_client)
>>> tb.rank_issues_by_list([my_jira_client.issue("JAT-001"), my_jira_
    ↪ client.issue("JAT-003")])
```

will produce following result

before	after
JAT-010	JAT-001
JAT-005	JAT-003
JAT-003	JAT-010
JAT-002	JAT-005
JAT-001	JAT-002

rank_issues_by_list(*ranked_list*, *on_top_of_issue*)

sorts the provided list by rank on top of the latter issue

Parameters

- **ranked_list** – list of issues to be sorted by rank index 0 has highest rank
- **on_top_of_issue** – issue on top of which these issues need to land

Example

```
>>> from jira_agile_toolbox import JiraAgileToolBox
>>> from jira import JIRA
>>> my_jira_client = JIRA("https://my-jira-server.com", basic_auth=(
```

(continues on next page)

(continued from previous page)

```
↪ "MYUSERNAME", "MYPASSWORD")
>>> tb = JiraAgileToolBox(my_jira_client)
>>> tb.rank_issues_by_list([my_jira_client.issue("JAT-001"), my_jira_
↪ client.issue("JAT-003")], my_jira_client.issue("JAT-005"))
```

will rank issues like:

before	after
JAT-010	JAT-010
JAT-005	JAT-001
JAT-003	JAT-003
JAT-002	JAT-005
JAT-001	JAT-002

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

INDEX

A

`add_labels_to_all_sub_items_of_epic()`
(*jira_agile_toolbox.JiraAgileToolBox* method),
3

C

`copy_fix_version_from_epic_to_all_items_in_epic()`
(*jira_agile_toolbox.JiraAgileToolBox* method),
3

G

`get_all_issues_in_epic()`
(*jira_agile_toolbox.JiraAgileToolBox* method),
4

`get_storypoints_from_epic()`
(*jira_agile_toolbox.JiraAgileToolBox* method),
4

J

`JiraAgileToolBox` (class in *jira_agile_toolbox*), 3

R

`rank_issues_at_top_of_project()`
(*jira_agile_toolbox.JiraAgileToolBox* method),
5

`rank_issues_by_list()`
(*jira_agile_toolbox.JiraAgileToolBox* method),
5