

---

# jira-agile-toolbox

*Release unknown*

**Jef Neefs**

**Jun 05, 2021**



# CONTENTS

<b>1</b>	<b>API documentation</b>	<b>3</b>
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Index</b>	<b>9</b>



```
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’

**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=(
    ↪ "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