# developer.skatelescope.org Documentation

***Release 0.1.0-beta***

**Marco Bartolini**

**Jun 23, 2021**

# HOME


**1 How to use** **1**

**2 GitLab API** **3**
2.1 How to add a new API endpoint . . . . . 4

**3 Jira API** **5**

**4 ReadtheDocs API** **7**

**5 Nexus API** **9**

**6 MongoDB API** **11**

**7 Local Module Deployment and Testing** **13**

**8 SKA DevSecOps Services API** **15**

CHAPTER

ONE

# HOW TO USE

Please install all the requirements with your preferred python virtual environment with the following bash command:

```
virtualenv venv
source venv/bin/activate
python3 -m pip install -r requirements.txt
python3 -m pip install -r requirements-dev.txt
```

Note: Depending on your environment you may need to update your 'setuptools' to install dependencies: 'python3 -m pip install -U setuptools'.

Run lint with the following command:

```
make lint
```

This project uses tox environments. Run tests with the following command:

```
make test
```

or test only a single feature (set the FEATURE variable):

```
make test_feature FEATURE=jira_api
```

Note: Tests assume you have Python 3.8 available.

Clean with the following command:

```
make clean
```

CHAPTER

TWO

# GITLAB API

The module ska.gitlab_api provides a class named GitLabApi which enables following:

- Get git diffs of a commit in a project
- Add a comment to a merge request (MR)
- Remove a comment from a merge request (MR)
- Get MR information
- Get MR Approvals (Approver related settings)
- Get project level merge request approvals settings (Maintainer related settings)
- Get all commits in an MR
- Get Project settings information
- Get Project members
- Get list of merge request comments
- Get a single comment in an MR
- Modify a single comment in an MR
- Search a string in project
- Get file information in a branch
- Create merge requests
- Create branches
- Delete branches

It is also needed to define the OS variables in a `PrivateRules.mak` file:

```
PRIVATE_TOKEN = yourauthenticationtoken
GITLAB_API_REQUESTER = yourrequesteruser
```

## 2.1 How to add a new API endpoint

1. Define the API method with a clear explanation in docstring following the rules:
   - Please include all the mandatory fields in the actual API documentation (*GitLab's API documentation*) as mandatory arguments (*no value assignment*) and include all the optional arguments with defaul values.
   - Please use plurals in the method name if you are returning a list of results instead of a single item.
   - Please use appropriate verbs in method names: `get`, `create`, `add`, `modify`, `remove`. Please refrain with starting any other words.
   - If the method is returning a single item add `_info` postfix or plural form as explained above.
   - **Important:** if the API endpoint is returning a single item, you should use `await self.gl.getitem(url)` method.
   - **Important:** if the API endpoint is returning a list, you should use `self.gl.getiter(url)` method in order to ensure traversal of all results. Otherwise, it will only return first 20(default defined by GitLab API) results.
2. Write BDD tests for the API capability (relevant scenario and test fixtures)
3. Update the documentation in this README file

CHAPTER

THREE

# JIRA API

The module ska.jira_api provides a class named SKAJira which enables following:

- Get Project lists
- Get a single issue
- Create single/multiple issue(s)
- Add a comment to a given issue
- Delete a comment

Note that, the authenticated user must have necessary permissions for the above functionality to work correctly.

It is also needed to define the OS variables in a `PrivateRules.mak` file:

```
JIRA_USERNAME = jirauser
JIRA_PASSWORD = strongpassword
JIRA_URL = https://jira.skatelescope.org
IGNORE_PROJECTS = IGP # list of project keys ignored
```

CHAPTER

FOUR

# READTHEDOCS API

The module ska.readthedocs_api provides a class named ReadTheDocsApi which enables following:

- Get subprojects of a project
- Get details of a single subproject with given parent project name

Note that, the authenticated user must have necessary permissions for the above functionality to work correctly.

It is also needed to define the OS variables in a `PrivateRules.mak` file:

```
RTD_TOKEN = authtokenforreadthedocs
```

CHAPTER

FIVE

# NEXUS API

The module ska.nexus_api provides a class named NexusApi which enables following:

- Get components lists given a nexus repository
- Get single component information given id
- Remove a component from its associated repository given id
- Upload a component to a nexus repository

Note that, the authenticated user must have necessary permissions for the above functionality to work correctly.

It is also needed to define the OS variables in a `PrivateRules.mak` file:

```
NEXUS_API_USERNAME = jirauser
NEXUS_API_PASSWORD = strongpassword
NEXUS_URL = https://jira.skatelescope.org
```

CHAPTER

SIX

# MONGODB API

The module ska.mongodb_api provides a class named MongoDBApi which enables following:

- Select the Database and the Collection
- Get a list of all collections and documents on a collection
- Remove a document by passing its field and name
- List outdated documents using the age
- Insert one or more documents
- Get the collection size

Note that, the authenticated user must have necessary permissions for the above functionality to work correctly.

The initial sequence to work with the MongoDB API should be :

- 1º Initialize it with a valid client
- 2º Select the Database
- 3º Select the Collection
- 4º Do the desired changes to that collection

CHAPTER

SEVEN

# LOCAL MODULE DEPLOYMENT AND TESTING

For local testing it might useful to install the modules with editable dependencies at the root (where `setup.py` is located):

```
pip install -e .
```

CHAPTER

EIGHT

# SKA DEVSECOPS SERVICES API

These are all the packages, functions and scripts that form part of the project.

- *SKA DevSecOps Services*