Contact

Daniele Fiorot

danyfiorot@gmail.com

https://github.com/danyfiorot

https://www.linkedin.com/in/danielefiorot/

Jira Mandatory Settings

1. Sprint Name MUST contain the Team Name
2. All Team Sprints MUST be created on the same Jira Scrum Board

How to Update the Reports (https://danyfiorot.github.io/pyjirametrics/index.html)

1. Please do all steps decribed on session **Before Start**

New Reports Version

1. Check if all needed Projects are correctly set in the **secrets/settings.json** file, property "**projects**".
2. Running the Metrics Update

On Windows OS, double click on **generate_metrics.bat** file. On Linux OS, run the following command on **jira_analysis** folder:

```
danielef@WNB024405BHZ MINGW64 /d/repos/jira_analysis (main)
$ sh scripts/shellscripts/generate_metrics.sh
```

3. In the end of the process you should see the message "*All set! \o/*", it means the reports updated successfully.

Old Reports Version (it will be deprecated soon)

1. Check if all needed Sprints are correctly set in the **secrets/settings.json** file, property "**all_cycles**".
2. Running the Metrics Update

On Windows OS, double click on **generate_metrics_old.bat** file. On Linux OS, run the following command on **jira_analysis** folder:

```
danielef@WNB024405BHZ MINGW64 /d/repos/jira_analysis (main)
$ sh scripts/shellscripts/generate_metrics_old.sh
```

3. In the end of the process you should see the message *"All set! \o/"*, it means the reports updated successfully.

Before Start

On Windows OS:

1. Install and Use the Linux Bash Shell on Windows: https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/
2. Install Python (https://www.python.org/downloads/) Check if Python is correctly installed:

```
python3 -V
```

3. Clone Repositories Both repositories MUST BE on the same parent folder. Clone or Download the repository https://github.com/danyfiorot/jira_analysis

```
git clone git@github.com:danyfiorot/jira_analysis.git
```

Clone or Download the repository https://github.com/danyfiorot/pyjirametrics

```
git clone git@github.com:danyfiorot/pyjirametrics.git
```

4. Install Libs On scripts folder, run the following command, libs will be installed:

```
danielef@WNB024405BHZ MINGW64 /d/repos/jira_analysis/scripts
sh shellscripts/first_setup.sh
```

5. Create json credentials file as the sample below (scripts/secrets/credentials.json)

```
{
 "host": "https://jira.com",
 "user": "MyUserID",
 "password": "mypassword",
 "email": "MyUserID@mymail.com"
}
```

Best Practices Links

- https://peps.python.org/pep-0008/
- https://www.w3schools.com/python/
- https://docs.python.org/3/reference/import.html#package-relative-imports

Jira REST API

- https://developer.atlassian.com/cloud/jira/platform/rest/v2

Test Coverage (To Run the Scripts you must run the tests)

PyTest: On main folder, run the following command, tests will run and reports will be generated:

```
danielef@WNB024405BHZ MINGW64 /d/repos/jira_analysis/scripts/tests (main)
python3 -m pytest -s
```

On main folder, run the following command, reports will be generated:

```
danielef@WNB024405BHZ MINGW64 /d/repos/jira_analysis (main)
$ python3 -m pytest -s scripts/tests/test_zzz_reports.py
```

Coverage: On main folder, run the following command, tests will run, coverage will be checked and reports will be generated:

```
danielef@WNB024405BHZ MINGW64 /d/repos/jira_analysis (main)
$ sh scripts/shellscripts/run-coverage.sh
```

Future Improvements

1. Test lib JIRA
2. Test lib logging
3. Map jira inconsistencies
4. Creat API to run from Web

Jira Access

Python Lib: https://github.com/pycontribs/jira

Steps to Create an API Token from your Atlassian Account:

1. Log in to https://id.atlassian.com/manage/api-tokens
2. Click Create API token.
3. From the dialog that appears, enter a memorable and concise Label for your token and click Create.
4. Click Copy to clipboard, then paste the token to your script. Reference: API tokens

GIT

Clone

```
$ git clone git@bitbucket.org:danielef_cit/abi-us-helper.git
```

Common Commands to Push the updates to Master

```
git status
git add .
git commit -m "First Commit \o/"
git push –all
```

Status

```
# Message when files have not been staged (git add)
$ git status
```

Add

```
# To add all files not staged:
$ git add .

# To stage a specific file:
$ git add index.html

# To stage an entire directory:
$ git add css
```

Commit

```
# Adding a commit with message
$ git commit -m "Commit message in quotes"
```

Branch

```
# Create a new branch
$ git branch <branch_name>

# List all remote or local branches
$ git branch -a

# Delete a branch
$ git branch -d <branch_name>
```

Checkout - Get on specific Branch

```
# Checkout an existing branch
$ git checkout <branch_name>

# Checkout and create a new branch with that name
$ git checkout -b <new_branch>
```

Merge

```
# Merge changes into current branch
$ git merge <branch_name>
```

Pull - Get a version

```
git pull <branch_name> <remote_URL/remote_name>
```

Push - Update the Branch

```
$ git push <remote_URL/remote_name> <branch>

# Push all local branches to remote repository
$ git push —all
```