
matyan Documentation

Release 0.1

Artur Barseghyan <artur.barseghyan@gmail.com>

Nov 17, 2019

Contents

1 Prerequisites	3
2 Use cases and basic concepts	5
2.1 How would you know it could be useful?	5
2.2 Sample use-case	5
2.2.1 The use-case	5
2.2.2 Sample commits	6
2.2.3 Sample releases	6
2.3 Sample changelog output	6
3 Configuration	9
4 Tips and tricks	11
4.1 Write to file	11
5 Documentation	13
6 Installation	15
7 Usage examples	17
8 Testing	19
9 Writing documentation	21
10 License	23
11 Support	25
12 Author	27
13 Docs	29
13.1 Release history and notes	29
13.1.1 0.1	29
14 Indices and tables	31

Generate changelog from Git commits.

CHAPTER 1

Prerequisites

- Python 3.6, 3.7, 3.8 and PyPy

Use cases and basic concepts

Matyan does a good job generating changelog from Git commits. Think of DTAP development, GitFlow-like concepts, JIRA (or similar tool for handling tickets and reviewing pull requests).

2.1 How would you know it could be useful?

If the following applies to you, `matyan` could help:

- Project releases (tags) are numbered according to the [semantic versioning](#) or [sequence based identifiers](#).
- Project follows the DTAP.
- Testing, acceptance and production branches (hereafter referred as TAP branches) are protected.
- Direct commits to TAP branches are forbidden.
- All commits to TAP branches are made by pull requests.
- Pull requests are merged using GitHub or BitBucket web interface.

2.2 Sample use-case

2.2.1 The use-case

- JIRA is used for issues.
- All commits are prefixed with ID of the JIRA issue: for example, *MSFT-1234* or *NVDA-1234* (where first four letters identify the client commit was done for, it's pattern).
- There are 3 main (protected) branches: *dev*, *staging*, *master*. Direct commits to any of the 3 are forbidden. Any feature/bugfix comes via merge request.
- All branches do have meaningful prefixes. Example, *feature/MSFT-1234-Title-of-the-issue* or *bugfix/MSFT-1236-prevent-duplicate-postal-codes*.

- Release flow is *dev* -> *staging* -> *master*.

2.2.2 Sample commits

Consider the following commits into the dev branch:

branch: feature/MSFT-1238-Token-authentication

- MSFT-1238 Implement token authentication.
- MSFT-1238 Update authentication docs.

branch: feature/MSFT-1237-Improve-document-sharing

- MSFT-1237 Improve document sharing. Add option to share via GDrive.

branch: bugfix/MSFT-1236-prevent-duplicate-postal-codes

- MSFT-1236 Normalise postal codes for German addresses.
- MSFT-1236 Normalise postal codes for US addresses.
- MSFT-1236 Make postal code field unique for the country.

branch: deprecation/MSFT-1235-deprecate-old-api

- MSFT-1235 Deprecate API v 2.0.
- MSFT-1235 Update docs.

branch: feature/MSFT-1234-car-type-suggester

- MSFT-1234 Initial car type suggester implementation.
- MSFT-1234 Add insurance amount indication based on car weight.

2.2.3 Sample releases

All commits have been finally merged into master.

Releases have been made in the following way:

0.1

- Merged issues MSFT-1234, MSFT-1235 and MSFT-1236

0.2

- Merged issues MSFT-1237 and MSFT-1238

2.3 Sample changelog output

The generated change log would look as follows:

```
### 0.2

**Features**

*MSFT-1238 Token-authentication*
```

(continues on next page)

(continued from previous page)

```
- Implement token authentication.
- Update authentication docs.

*MSFT-1237 Improve document sharing*

- Improve document sharing. Add option to share via GDrive.

### 0.1

**Bugfixes**

*MSFT-1236 Prevent duplicate postal codes*

- Normalise postal codes for German addresses.
- Normalise postal codes for US addresses.
- Make postal code field unique for the country.

**Deprecations**

*MSFT-1235 Deprecate old api*

- Deprecate API v 2.0.
- Update docs.

**Features**

*MSFT-1234 Car type suggester*

- Initial car type suggester implementation.
- Add insurance amount indication based on car weight.
```


CHAPTER 3

Configuration

In order to customize names and texts, add a `.matyan.ini` in your project directory, from which you will be running the `generate-changelog` command.

Sample configuration:

```
[BranchTypes]
feature: Feature
bugfix: Bugfix
hotfix: Hotfix
deprecation: Deprecation

[OtherBranchType]
other: Other

[Unreleased]
unreleased: Unreleased

[IgnoreCommits]
exact: more
      clean up
      code comments
      more on docs
      repo
      working
      more on
      wip
      commit
```


CHAPTER 4

Tips and tricks

4.1 Write to file

```
generate-changelog --show-releases 2>&1 | tee changelog.md
```


CHAPTER 5

Documentation

Documentation is available on [Read the Docs](#).

CHAPTER 6

Installation

Latest stable version on PyPI:

```
pip install matyan
```

Or latest stable version from GitHub:

```
pip install https://github.com/barseghyanartur/matyan/archive/stable.tar.gz
```

Or latest stable version from BitBucket:

```
pip install https://bitbucket.org/barseghyanartur/matyan/get/stable.tar.gz
```

Usage examples

Generate changelog:

```
generate-changelog
```

Generate changelog skipping orphaned commits:

In some cases you only want to show what has been done with tickets and skip all non-ticket related commits.

```
generate-changelog --no-other
```

Generate changelog between two releases:

In some cases you only want to show what has been done since last release. The following example would generate changelog since version 0.0.1 to version 0.0.3.

```
generate-changelog 0.0.1..0.0.3
```

Generate changelog with releases info shown

```
generate-changelog --show-releases
```

Generate changelog between releases with releases info shown

```
generate-changelog 0.0.1..0.0.3 --show-releases
```


CHAPTER 8

Testing

Simply type:

```
./runtests.py
```

Or use tox:

```
tox
```

Or use tox to check specific env:

```
tox -e py36
```

Writing documentation

Keep the following hierarchy.

```
=====  
title  
=====  
  
header  
=====  
  
sub-header  
-----  
  
sub-sub-header  
~~~~~  
  
sub-sub-sub-header  
^^^^^^  
  
sub-sub-sub-sub-header  
++++++  
  
sub-sub-sub-sub-sub-header  
*****
```


CHAPTER 10

License

GPL-2.0-only OR LGPL-2.1-or-later

CHAPTER 11

Support

For any issues contact me at the e-mail given in the *Author* section.

CHAPTER 12

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

Contents:

13.1 Release history and notes

Sequence based identifiers are used for versioning (schema follows below):

```
major.minor[.revision]
```

- It's always safe to upgrade within the same minor version (for example, from 0.3 to 0.3.4).
- Minor version changes might be backwards incompatible. Read the release notes carefully before upgrading (for example, when upgrading from 0.3.4 to 0.4).
- All backwards incompatible changes are mentioned in this document.

13.1.1 0.1

2019-11-18

- Initial release.

CHAPTER 14

Indices and tables

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