Zalando Turnstile Documentation

Release 2.1

Zalando SE

March 04, 2016

Contents

Turnstile is a configurable and extensible local git hook. You can find it on Github and on Pypi.

Contents:

CHAPTER 1

Developers's Guide

Useful documentation on how to enhance and extend Turnstile

1.1 Commit Specification Scheme

1.1.1 Overview

Git commits provided for SCMs such as github.com, github enterprise or stash has to follow a certain template in order to monitor commits in an automated way.

This document defines such a template.

A commit as defined in this document contains of a specification part and an actual message part.

The specification part can be a uniform resource identifier (URI) according to RFC 3686, a github reference or a JIRA ticket key.

The specification must be the first "word" of the commit.

CHAPTER 2

User's Guide

How to use the turnstile

2.1 Checks

2.1.1 Branch Pattern Check

Checks if the branch names matches any regex pattern on list defined in the repository options. Master branch is always allowed.

```
branch-pattern:
    allowed:
        - "^release/R"
        - "^feature/"
```

2.1.2 Branch Release Check

Check if the release of a release branch (`release\*`) matches a pattern. By default this pattern is `^R(?:\d|\_|\.)+$` but it's configurable:

```
branch-release:
    pattern: '*.'
```

2.1.3 Branch Type Check

Checks if the branch type is the allowed types list in the repository options. The branch type is the prefix of the branch name, for example `feature/CD-100` is a feature branch.

Master branch is always allowed.

```
branch-type:
    allowed:
        - release
        - feature
```

2.1.4 Protect Master Check

If this check is enabled turnstile will prevent commits to master.

2.1.5 Specification Check

This check verifies if the commit message starts with a valid reference to a specification. Turnstile supports several formats to link to the specification. By default it allows only *URI* specification but you can change the allowed formats:

```
specification:
    allowed_format: ['uri', 'github', 'jira']
```

Github

Checks if the specification is a valid github reference.

Jira

Checks if the specification is a valid Jira ticket key.

URI

Checks if a specification URI is a valid and absolute. This check is ignored for merge commits.

By default only HTTPS and offline URIs are accepted but you can change the allowed schemes:

```
specification:
    allowed_format: ['uri']
    allowed_schemes: ['https', 'ftp']
```

2.2 Repository Configuration

2.2.1 Overview

Turnstile read the configuration from a `.turnstile.yml` file on the root of the repository.

2.2.2 Common Parameters

parameter	Description
checks	List of checks you want to run

2.2.3 Check Parameters

Some checks have check specific configuration that can be specified in a parameter with the check name. Please look in the check documentation for more information.

2.2.4 List of checks

- *Branch Pattern Check* - Check if branch name matches one of the allowed patterns
- *Branch Release Check* - Check if release branches names contain a valid release name
- *Branch Type Check* - Check if branch type is allowed

- *Protect Master Check* - Prevents commits to master
- *Specification Check* - Check if commit message contains a valid specification

2.3 Turnstile Subcommands

2.3.1 open-spec

This subcommand opens the specification for a commit in your default browser.

Currently it supports both URI and Github references.

Usage

```
$ turnstile open-spec [OPTIONS] [REFERENCE]
```

2.3.2 specification

This subcommand verifies if the commit messages in a range of revisions have valid specifications.

This command takes the same revision ranges as `git log` to specify the revision ranges.

When using the verbose mode merge commits are printed otherwise they are simply ignored.

Usage

```
$ turnstile specification [<revision range>]
```

2.3.3 upgrade

This subcommand gets the latest versions of turnstile and installed extension from Pypi and allows you to automatically upgrade Turnstile if there is a newer version.

Usage

To upgrade turnstile run:

```
$ turnstile upgrade
```

2.3.4 Usage

To use a turnstile subcommand do:

```
$ turnstile <subcommand>
```

2.3.5 Subcommand list

Subcommand	Description
config	Set turnstile configuration
install	Installs git hooks in repository
open-spec	Opens commit specification in browser
remove	Removes git hooks from repository
specification	Verifies if the commit messages in a range of revisions have valid specifications.
upgrade	Upgrade turnstile from Pypi
version	Prints Turnstile version