

pre-commit-crocodile

Author: Adrian DC

Description: Git hooks intended for developers using pre-commit

Date: 02/02/2025

Version: 4.0.0-7-g582f84f

► Usage

pypi v4.0.0 python 3.9 | 3.10 | 3.11 | 3.12 downloads 313/month license Apache License 2.0

pipeline manual bugs 0 code smells 0 coverage 58.6% lines of code 2.2k

quality gate passed

pre-commit enabled commitizen friendly gcil enabled pre-commit-crocodile enabled

radiandevcore guidelines

Git hooks intended for developers using [pre-commit](#) and [commitizen](#)

Documentation: <https://radiandevcore.gitlab.io/tools/pre-commit-crocodile>

Package: <https://pypi.org/project/pre-commit-crocodile/>

Table of contents

- [Features](#)
- [Preview](#)
- [Usage](#)
- [Installation](#)
- [Compatibility](#)
- [Projects with configurations |](#)
 - [Install dependencies \(once per user\)](#)
 - [Enable hooks for a project](#)
 - [Manually run hooks of a project](#)
 - [Disable hooks for a project](#)
- [Projects without configurations |](#)
 - [Import or refresh configurations](#)
- [Projects maintenance |](#)
 - [Update hooks automatically](#)
 - [Cleanup hooks cache](#)
- [Dependencies](#)
- [References](#)

Features

pre-commit-crocodile uses the following features:

- **CLI - pre-commit:** Automated Git hooks before commits and upon pushes
- **CLI - commitizen:** Commits tools and validation based upon [conventional commits](#)
- **Hooks - pre-commit-hooks:** Common pre-commit hooks useful for developers
- **Hooks - prepare-commit-msg:** Prepare commit message automatically based on changes

Preview

```
--update-check      Check for newer package updates
--settings          Show the current settings path and contents
--set GROUP KEY VAL Set settings specific 'VAL' value to [GROUP] > KEY
                   or unset by using 'UNSET' as 'VAL'

modes arguments:
-l, --list          List Git hooks installed in sources
-i, --install       Install dependencies for pre-commit hooks
-c, --configure    Update sources with hooks configurations
-e, --enable       Enable pre-commit hooks
-d, --disable      Disable pre-commit hooks
-a, --autoupdate   Autoupdate pre-commit hooks
-r, --run          Run pre-commit hooks

positional arguments:
--                  Positional arguments separator (recommended)

adriandc@preview:~/pre-commit-crocodile$ pre-commit-crocodile --install

pre-commit-crocodile (2.0.1)
- Install dependencies: pre-commit

⚙️ upgrading shared libraries
```

Usage

```
usage: pre-commit-crocodile [-h] [--version] [--no-color] [--update-check] [--settings] [--set GROUP KEY VAL]
                             [-l | -i | -c | -b | -e | -d | -a | -C | -r] [--config FOLDER | -D] [--stage STAGE] [--]
```

pre-commit-crocodile: Git hooks intended for developers using pre-commit

internal arguments:

```
-h, --help           # Show this help message
--version           # Show the current version
--no-color          # Disable colors outputs with 'NO_COLOR=1'
                    # (or default settings: [themes] > no_color)
--update-check      # Check for newer package updates
--settings          # Show the current settings path and contents
--set GROUP KEY VAL # Set settings specific 'VAL' value to [GROUP] > KEY
                    # or unset by using 'UNSET' as 'VAL'
```

modes arguments:

```
-l, --list           # List Git hooks installed in sources
-i, --install        # Install dependencies for pre-commit hooks
-c, --configure      # Update sources with hooks configurations
-b, --badges         # Update documentation with badges configurations
-e, --enable         # Enable pre-commit hooks
-d, --disable        # Disable pre-commit hooks
-a, --autoupdate     # Autoupdate pre-commit hooks
-C, --clean          # Clean pre-commit cached hooks
-r, --run            # Run pre-commit hooks
```

configurations arguments:

```
--config FOLDER     # Use configurations from a specific folder
-D, --default        # Use global default configurations instead of sources
--stage STAGE        # Run a specific pre-commit stage with --run
                    # (use 'list' to list supported stages)
```

positional arguments:

```
--                 # Positional arguments separator (recommended)
```

Installation

```
{
# Option 1: If using pipx
if type pipx >/dev/null 2>&1; then
  pipx ensurepath
  pipx install pre-commit-crocodile
  pipx upgrade pre-commit-crocodile

# Option 2: If using pip
else
  sudo pip3 install pre-commit-crocodile
fi
}
```

Compatibility

Projects compatible with `pre-commit-crocodile` can use this badge to ease things for developers, both as an indicator and a documentation shortcut button :



Badge in Markdown

```
[![pre-commit-crocodile](https://img.shields.io/badge/pre--commit--crocodile-enabled-brightgreen?logo=gitlab)]
(https://radiandevcore.gitlab.io/tools/pre-commit-crocodile)
```

Badge in HTML

```
<a href="https://radiandevcore.gitlab.io/tools/pre-commit-crocodile"></a>
```

Projects with configurations | pre-commit-crocodile enabled

Install dependencies (once per user)

```
pre-commit-crocodile --install
```

Enable hooks for a project

```
pre-commit-crocodile --enable
```

Manually run hooks of a project

```
pre-commit-crocodile --run
```

Disable hooks for a project

```
pre-commit-crocodile --disable
```

Projects without configurations | pre-commit missing

Import or refresh configurations

```
pre-commit-crocodile --configure
```

Projects maintenance | pre-commit-crocodile enabled

Update hooks automatically

```
pre-commit-crocodile --autoupdate
```

Cleanup hooks cache

```
pre-commit-crocodile --clean
```


Dependencies

- [colored](#): Terminal colors and styles
 - [commitizen](#): Simple commit conventions for internet citizens
 - [pre-commit](#): A framework for managing and maintaining pre-commit hooks
 - [pre-commit-crocodile](#): Git hooks intended for developers using pre-commit
 - [setuptools](#): Build and manage Python packages
 - [update-checker](#): Check for package updates on PyPI
-

References

- [.gitlab-ci.yml](#): GitLab CI/CD Pipeline Configuration Reference
- [conventionalcommits](#): Conventional Commits specification for commit messages
- [gcil](#): Launch .gitlab-ci.yml jobs locally
- [git-cliff](#): CHANGELOG generator
- [gitlab-release](#): Utility for publishing on GitLab
- [mkdocs](#): Project documentation with Markdown
- [mkdocs-exporter](#): Exporter plugin for mkdocs documentation
- [mkdocs-material](#): Material theme for mkdocs documentation
- [mypy](#): Optional static typing for Python
- [pexpect-executor](#): Automate interactive CLI tools actions
- [PyPI](#): The Python Package Index
- [termtosvg](#): Record terminal sessions as SVG animations
- [twine](#): Utility for publishing on PyPI

pre-commit-crocodile