conhead

Release 0

Rafe Kaplan

Aug 23, 2022

CONTENTS:

CLI tool for applying and maintaining consistent headers across source files.

- Add headers to files that don’t have them.
- Update fields in files that already have them.
- Maintain different header configurations for different file types

CHAPTER

ONE

INSTALLATION

Conhead is available as the Python `conhead` package.

For example, to install using pipx:

```
$ pipx install conhead
$ conhead --help
Usage: conhead [OPTIONS] SRC
```

CHAPTER TWO

CONFIGURATION

Configure conhead via pyproject.toml. Each header template is configured via a separate [tools.conhead.header.<name>] section. Each section is a header definition and can be applied to one or more file extensions.

Example:

```
[tools.conhead.header.hashhead]
extensions = ['py', 'toml', 'yaml']
template = """
    # Copyright {{YEARS}} Organized Organization
    # SPDX-License-Identifier: Apache-2.0
"""

[tools.conhead.header.slashhead]
extensions = ['c', 'cpp', 'java']
template = """
    // Copyright {{YEARS}} Organized Organization
    // SPDX-License-Identifier: Apache-2.0
"""

[tools.conhead.header.dashhead]
extensions = ['html']
template = """
    <!-- Copyright {{YEARS}} Organized Organization
         SPDX-License-Identifier: Apache-2.0
    -->
"""
```

2.1 Template Definition

A few things to note about the template definition.

Each TOML tools.conhead.header section has a few options:

- **extensions:** A list of extensions that the header definition applies to.
- **template:** The header template for this header definition. This is the text that is applied to files that have the indicated extensions.

2.2 Header Templates

Notice a few things about the header template.

- The text of the template is indented for better readability withing the `pyproject.toml` configuration file, however `conhead` de-indents this text for you.
- The template contains a field that is kept up to date in the target source file. In this case the `{{YEARS}}` field writes the current year into every new template. If a file already contains a header with the year in it, and the year is different from the current year, it is updated to show a range of years. For example, a new template would have the `{{YEARS}}` field replaced with `2020` if it was first written in `2020`. When the header is then updated in `2022`, this field is rewritten as `2020-2022`.
- If you need to write some text that contains certain characters used to describe fields, you must escape them. Examples are `\{`, `\}` and `\\`. These characters will appear in the rendered header without the preceding slash.

CHAPTER

THREE

USAGE

Let's say there is a python file without a header at `hello.py`:

```
def hello():
    print("Greetings!")
```

You can apply the `hashhead` header template defined in `pyproject.toml` and view the results by:

```
$ conhead hello.py
WARNING: missing header: hello.py

$ cat hello.py
# Copyright 2022 Organized Organization
# SPDX-License-Identifier: Apache-2.0


def hello():
    print("Greetings!")
```

`conhead` will recognize the header if you apply it to `hello.py` again and will not write a second header.

```
$ conhead hello.py

$ cat hello.py
# Copyright 2022 Organized Organization
# SPDX-License-Identifier: Apache-2.0


def hello():
    print("Greetings!")
```

CHAPTER FOUR

PRE-COMMIT

conhead is pre-commit ready. To use with pre-commit, add the repo to your .pre-commit-config.yaml.

For example:

```yaml
- repo: https://github.com/slobberchops/conhead
  rev: v0.4.0
  hooks:

    - id: conhead
```

CHAPTER FIVE

LINKS

- Changes: https://github.com/slobberchops/conhead/blob/main/CHANGES.rst
- PyPI Releases: https://pypi.org/project/conhead/
- Source Code: https://github.com/slobberchops/conhead
- Issue Tracker: https://github.com/slobberchops/conhead/issues