sphinxcontrib-eval

Wu Zhenyu

Jan 12, 2023

RESOURCES

You can override the default `eval_funcs` in your `docs/conf.py`:

CHAPTER

ONE

ONLY EVALUATE SHELL

```
from sphinxcontrib.eval import eval_sh

eval_funcs = {"sh": eval_sh}
```

CHAPTER

TWO

EVALUATE SHELL, PYTHON, VIM SCRIPT

```
from sphinxcontrib.eval import eval_sh, eval_python
from subprocess import check_output


def eval_vim(input: str) -> str:
    try:
        output = check_output(  # nosec B603
            ["nvim", "--headless", "-c", input, "-c", "q"],
            universal_newlines=True,
        )
    except Exception:
        output = ""
    return output


eval_funcs = {"sh": eval_sh, "python": eval_python, "vim": eval_vim}
```

CHAPTER

THREE

EVALUATE CMD

```
def eval_cmd(input: str) -> str:
    try:
        output = check_output(  # nosec B603
            ["cmd", "/c", input], universal_newlines=True
        )
    except Exception:
        output = ""
    return output


eval_funcs = {"cmd": eval_cmd}
```

CHAPTER

FOUR

FAQ

4.1 eval-sh does not work for Windows

POSIX shell is not installed in Windows by default. See *evaluate cmd* or install a POSIX shell

4.2 Why not Makefile

Save your time to write Makefile.

4.3 Why this name

Comes from `eval-rst` of MyST.

4.4 What is translate shell

My another project. The intention of this project is to serve its document generation.

4.5 Eval is evil

I don't guarantee any consequence like:

```
rm -rf your_import_file
```

4.6 Difference from other projects

- sphinx-execute-code Display code execute result in a code fence, not inject generated markdown/rst to the source code.
- sphinxcontrib-programoutput Same as above.

CHAPTER

FIVE

INSTALL & UNINSTALL

5.1 PYPI

```
pip install sphinxcontrib-eval
```

```
pip uninstall sphinxcontrib-eval
```

5.2 AUR

```
yay -S python-sphinxcontrib-eval
```

```
sudo pacman -R python-sphinxcontrib-eval
```

CHAPTER

SIX

REQUIREMENTS

6.1 dev

For unit test and code coverage rate test.

- myst-parser
- pre-commit
- pytest-cov
- tomli

6.2 myst

For markdown. Recommend `pip install 'sphinxcontrib-eval[myst]'`.

- myst-parser

CHAPTER

SEVEN

EVAL

Provide `setup()` to sphinx.

sphinxcontrib.eval.**eval_bash**(*input: str*) → str

Eval bash.

Parameters
input (*str*) –

Return type
str

sphinxcontrib.eval.**eval_python**(*input: str*) → str

Eval python.

Parameters
input (*str*) –

Return type
str

sphinxcontrib.eval.**eval_sh**(*input: str*) → str

Eval sh.

Parameters
input (*str*) –

Return type
str

sphinxcontrib.eval.**setup**(*app: Sphinx*) → dict[str, Any]

Set up.

Parameters
app (*Sphinx*) –

Return type
dict[str, Any]

7.1 MyST

Provide `MystEvalParser`.

7.2 RST

Provide `RSTEvalParser`.

7.3 Utilities

sphinxcontrib.eval.utils.**get_lang_map**(*template: str*, *eval_funcs: dict[str, Callable[[str], str]]*) → dict[str, tuple[re.Pattern, Callable[[str], str]]]

Get lang map.

Parameters

- **template** (*str*) –
- **eval_funcs** (*dict[str, Callable[[str], str]]*) –

Return type
dict[str, tuple[re.Pattern, Callable[[str], str]]]

sphinxcontrib.eval.utils.**patch_parser**(*template: str*, *parser: Type[Parser]*) → Type[Parser]

Patch parser.

Parameters

- **template** (*str*) –
- **parser** (*Type[Parser]*) –

Return type
Type[Parser]

sphinxcontrib.eval.utils.**replace**(*inputstring: str*, *pat: Pattern*, *eval_func: Callable[[str], str]*) → str

Replace.

Parameters

- **inputstring** (*str*) –
- **pat** (*re.Pattern*) –
- **eval_func** (*Callable[[str], str]*) –

Return type
str

CHAPTER

EIGHT

SPHINXCONTRIB-EVAL

Evaluate shell command or python code in sphinx and myst.

8.1 Usage

8.1.1 Enable

docs/conf.py

```
extensions = [
    "sphinxcontrib.eval",
]
```

Or

```
extensions = [
    "myst_parser",
    "sphinxcontrib.eval",  # must be after myst_parser
]
```

8.1.2 Demonstration

For myst:

````
```{eval-sh}
echo My OS is $OSTYPE.
```
````

For rst:

```
.. eval-sh::
    echo My OS is $OSTYPE.
```

Then build:

```
sphinx-build docs docs/_build/html
```

Result:

```
My OS is linux-gnu.
```

NOTE: the current working directory depends on you. That is, if you run `cd docs && sphinx-build . _build/html && cd -`, CWD will be `docs`, which is the default setting of https://readthedocs.org. So if your code structure is like

```
$ tree --level 1
.
├── docs
├── scripts
├── src
└── tests
```

And you want to run `scripts/*.sh`, you need `cd ..` firstly from `docs` to `.` else you have to run `../scripts/*.sh`.

8.1.3 Advanced Usages

All of the following examples are myst. The corresponding examples of rst are similar. Click the hyperlinks of the titles and scripts to see the actual examples.

Generate API Document

Before:

````
# API of Translate Shell

```{eval-rst}
.. automodule:: translate_shell
    :members:
.. automodule:: translate_shell.__main__
    :members:
... (More)
```
````

Now

````
# API of Translate Shell

````{eval-rst}
```{eval-sh}
cd ..
scripts/generate-api.md.pl src/*/*.py
```
````
````

Where `scripts/generate-api.md.pl` replaces all `src/translate_shell/XXX.py` to

```
.. automodule:: translate_shell.XXX
    :members:
```

Generate TODO Document

Before:

```
# TODO

- <https://github.com/Freed-Wu/tranlate-shell/tree/main/src/translate_shell/translators/
↪stardict/__init__.py#L4>
  more stardicts.
- <https://github.com/Freed-Wu/tranlate-shell/tree/main/src/translate_shell/translators/
↪stardict/__init__.py#L5>
  Create different subclasses for different dict to get phonetic, explains
- <https://github.com/Freed-Wu/tranlate-shell/tree/main/src/translate_shell/ui/repl.py
↪#L33>
  make the last line gray like ptpython
- ...
```

Now: (notice `eval-bash` because readthedocs uses dash as their default `$SHELL`)

```
# TODO

```{eval-bash}
cd ..
scripts/generate-todo.md.pl src/**/*
```
```

Where `scripts/generate-todo.md.pl` searches all `TODO`s in code then convert them to correct hyperlinks.

Generate Requirements Document

Before:

```
# Requirements

## completion

Generate shell completion scripts.

- [shtab](https://pypi.org/project/shtab)

...
```

Now

```
# Requirements

```{eval-sh}
cd ..
generate-requirements.md.pl
```
```

Where `scripts/generate-requirements.md.pl` searches all `requirements/*.txt`s and `requirements/completion.txt` is:

```
#!/usr/bin/env -S pip install -r
# Generate shell completion scripts.

shtab
```

See document to know more.

PYTHON MODULE INDEX

S

INDEX