requirements-txt Documentation

Release latest

Jul 13, 2022

CONTENTS

to-requirements.txt allows to automatically add and delete modules to requirements.txt installing them using **pip**.

Easy to setup
The installation process include only two steps: install the package using pip and setup up it using script provided by the package. That's it.

Customizable
Customize it the way you like: use it only in git repositories, allow or disallow automated requirements.txt file creation, enable or disable the package itself.

Easy to use
After installing the package, running setup command and (optionally) customizing it the package is ready. There is no additional conditions to use. Just install, uninstall or upgrade packages using *pip* as you usually do.

Always in sync
With *to-requirements.txt* the project's requirements.txt will always stay in sync with packages that you install using *pip*.

CHAPTER

ONE

INSTALLATION

To install the package run the following command:

```
pip install to-requirements.txt
```

And after that run the command below to initialize the package:

```
requirements-txt setup
```

It will update your current *pip* scripts to execute the functionality of this package. Also, if you want to enable all the available functionality of the package you should put the lines below to your .bashrc, .zshrc or other .*rc file:

```
alias rt=". rt"
alias requirements-txt=". requirements-txt"
```

Or simply use the cli command:

```
rt alias
```

It will enable sourced mode of the cli execution and the cli will be able to activate your virtual environment / deactivate it if required.

> The changes made to *pip* scripts will not affect ordinary *pip* workflow after uninstalling *to-requirements.txt*.

1.1 Instructions

In this section you will get started with the package.

1.1.1 Configurations

Configurations of *to-requirements.txt* allows you to customize the way the package deal with requirements.txt the way you like.

There is two sets of configs: local and global. Local settings applies to your current folder. Global settings are used as default as local settings are not specified.

View the config

The following command will output the local config or to-requirements.txt:

```
requirements-txt config
```

The following command will output the global config or to-requirements.txt:

```
requirements-txt config --global
```

Enable / Disable the package

- Name: **disable**
- Values: **0, 1**
- Default: **0**

To disable the package use the following command:

```
requirements-txt config disable 1
```

To enable the package use the following command:

```
requirements-txt config disable 0
```

Only git repository

- Name: **only_git**
- Values: **0, 1**
- Default: **0**

Enable only git repositories

Enable saving to requirements.txt only in git repositories:

```
requirements-txt config only_git 1
```

Enable saving to requirements.txt only in git repositories globally:

```
requirements-txt config --global only_git 1
```

Disable only git repositories

Disable saving to requirements.txt only in git repositories:

```
requirements-txt config only_git 1
```

Disable saving to requirements.txt only in git repositories globally:

```
requirements-txt config --global only_git 1
```

Allow create requirements.txt

- Name: **allow_create**
- Values: **0, 1**
- Default: **0**

Allow requirements.txt creation

To allow the package to create requirements.txt if it does not exist:

```
requirements-txt config allow_create 1
```

To allow the package to create requirements.txt if it does not exist globally:

```
requirements-txt config --global allow_create 1
```

Disallow requirements.txt creation

To disallow the package to create requirements.txt if it does not exist:

```
requirements-txt config allow_create 0
```

To disallow the package to create requirements.txt if it does not exist globally:

```
requirements-txt config --global allow_create 0
```

Config files

Local settings are stored in the current directory.

```
./.to-requirements.txt/default.ini
```

Global settings are stored in the user folder. For Linux:

```
/home/<user>/.to-requirements.txt/default.ini
```

1.1.2 Usage

In this section you can find docs on how to use requirements-txt command line tool.

Basic usage

There are two ways to use requirements-txt command line tool. The first one is:

```
requirements-txt <command> <options>
```

The second one is to use alias:

```
rt <command> <options>
```

Both commands results in the same outcome. It's just an alias.

Alias

To activate full potential of the cli you should add following lines to your .bashrc, .zshrc or other .*rc file:

```
alias rt=". rt"
alias requirements-txt=". requirements-txt"
```

It will allow requirements-txt cli to activate your environment and init command. You can also achieve this by the command:

```
rt alias
```

This command will work for the following shells:

- bash
- zsh

Init

If you want to initialize virtual environment for your project use:

```
rt init
```

Or with alias:

```
rt i
```

This command will:

1. Install virtualenv package with your pip or skip this step if it's already installed.
2. Create requirements.txt file or skip this step if it already exists.
3. Install dependencies from your requirements.txt.
4. Install to-requirements.txt to your newly created environment.
5. Setup installed to-requirements.txt.

After that you can source your environment with

```
source venv/bin/activate
```

and start working on the project.

Install

To override your pip scripts and enable requirements.txt synchronization use:

```
rt install
```

This command is not required if use set up your environment with init command.

Help

To get more information about commands use the following command:

```
rt --help
```

Config

To use configuration util:

```
rt config <configuration options>
```

Show

If you want to see contents of your requirements.txt file type this:

```
rt show
```

Or with alias:

```
rt s
```