# Master Project Koen Matas Manual

This manual describes the steps that have to be taken, in order to be able to run the Python Scrapy scripts, that are found in the documents sent alongside the master project handed in by Koen Matas. These scripts include the web scraping scripts for the following websites: Billboard top 200, Billboard top 200 Global, Grammy, and Metacritic. Additionally a settings.py file sent that contains the settings used for scraping without overloading the servers of the targeted sites.

## Installation

In order to run the web scrapers the following installations are required:

Python

Python is required to run the scripts, please check for the latest version of python on <https://www.python.org/downloads/>

Virtual environment

It is recommended that you run the scripts from a virtual environment. If pipenv is not installed yet on your device, open a command-line or terminal of your choice and write the following command:

**pip install** **pipenv**

This will install pipenv which can then be used to create a virtual environment in your project folder.

Scrapy

After installing Python, scrapy needs to be installed. Scrapy is used as the web-scraping tool of choice in the master project. To install scrapy, open a command-line or terminal of your choice and write the following command:

**pip install** **scrapy**

This in turn will start the download of scrapy.

## Running the scripts

In order to run the scripts on your device please open a command-line or terminal of your choice. Navigate to the folder which you like to use for running the scripts. Here, you will first need to create a virtual environment by running the following command in your terminal:

**Virtualenv**

This will take a few second but after completion it will have created a virtual environment. Next you will need to active the virtual environment by running the following command in your terminal:

**Source venv/bin/activate**

After creating and activating the virtual environment you are ready to use scrapy. Start using scrapy by writing the following command in your terminal:

**Scrapy startproject (write your project name here)**

By running his command a project folder will be created with all the necessary items to run the scrapy web scrapers. These items include:

* A folder called “spiders”
* \_\_init\_\_.pu
* Items.py
* Middlewares.py
* Pipelines.py
* Settings.py

Here you will need to replace the settings.py file with the settings.py file received in the documents of the master project by Koen Matas. Next, you can add all the spiders received in the master project of Koen Matas in the folder named “spiders” these include the following items:

Billboard.py

Used to run the billboard top 200 scraper. In order to run this scraper write the following command in your terminal:

**Scrapy crawl billboard**

This command will start to run the scraper starting from the first recorded items at 17-7-1963. If you would like to run a short test-run instead of scraping the full database, please place a different starting date in the code as seen below:



Billboard\_top\_global.py

Used to run the billboard top 200 global scraper. In order to run this scraper write the following command in your terminal:

**Scrapy crawl billboardglobal**

This command will start to run the scraper . If you would like to run a short test-run instead of scraping the full database, please place a different starting date in the same code as seen above.

Grammy.py

Used to run the Grammy scraper. In order to run this scraper write the following command in your terminal:

**Scrapy crawl grammy**

Metacritic2.py

The final file in the documents contains the Metacritic scraper. To run this scraper write the following command in your terminal:

**Scrapy crawl metacritic1**

Because this scraper is designed to scraper more than 128 pages containing more than 300.000 data entries, it is recommended that this scraper is run in test mode. This is done by adjusting the page number range in the code to maximum (0, 2) found here:

Text

Description automatically generated