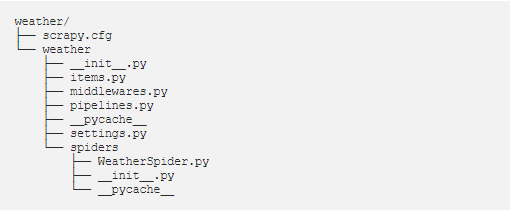
|  |  |  |
| --- | --- | --- |
| Tool | Tool Description | Tested Version |
| [Python](https://www.python.org/downloads/release/python-378/) | Python ( Tested version with below tools : 3.7.8) | 3.7.8 |
| [Elasticsearch](https://www.elastic.co/start) | Distributed, search and analytics engine | 7.9.3 |
| [Kibana](https://www.elastic.co/start) | Elasticsearch Index visualization | 7.9.3 |
| [Longstash](https://www.elastic.co/guide/en/logstash/current/installing-logstash.html) \*optional | Load Bulk data into elasticsearch |  |
| [Scrapy](https://docs.scrapy.org/en/latest/intro/install.html) | Scrape framework | 2.4.1 |
| [ScrapyElasticSearch](https://pypi.org/project/ScrapyElasticSearch/) | Scrapy pipeline which allows you to store scrapy items in Elastic Search. | 0.9.2 |

Documentation for Scrapy Spiders :

* <https://towardsdatascience.com/web-scraping-with-scrapy-practical-understanding-2fbdae337a3b>
* <https://devhints.io/xpath>
* <https://www.simplified.guide/scrapy/scrape-table>

Scrapy framework – Project build Path :



Let’s say we need to create a new scrapy project for weathersite.gr . Then we should follow the steps below:

* Create the buildpath with the command: scrapy startproject weathersite

Now we have create the above buildpath of our project

* Inside spiders folder generate the Spider with the command :

scrapy genspider -t basic weathersite\_spider weathersite.com

* The configuration is ready, we can run the spider with the command (under spider folder):

scrapy crawl weather\_spider

Example:

Let’s say we need to scrape data for [meteo.gr](http://penteli.meteo.gr/stations/tripoli/) (weather station) and load them into an elasticsearch index (we assume that we have already create an index at elasticsearch with name “weather”).

We create a scrapy project with: scrapy startproject meteo

We move into spiders folder and run : scrapy genspider -t basic meteo\_spider meteo.gr

First of all we need to describe the fields of the item we scrape. So we have to configure the **items.py**

import scrapy

class MeteoItem(scrapy.Item):

id = scrapy.Field()

source = scrapy.Field()

time = scrapy.Field()

timecrawl = scrapy.Field()

temperature = scrapy.Field()

humidity = scrapy.Field()

wind = scrapy.Field()

barometer = scrapy.Field()

yetos = scrapy.Field()

direction = scrapy.Field()

city = scrapy.Field()

Then we should configure the **setting.py**

ITEM\_PIPELINES = {

'meteo.scrapyelasticsearch.ElasticSearchPipeline': 500

}

ELASTICSEARCH\_SERVERS = ['localhost']

ELASTICSEARCH\_INDEX = 'weather'

ELASTICSEARCH\_INDEX\_DATE\_FORMAT = ''

ELASTICSEARCH\_TYPE = 'items'

ELASTICSEARCH\_UNIQ\_KEY = 'id' # Custom unique key

The scrapyelasticsearch.ElasticSearchPipeline is a class which needs the classes-files below (you should keep them at the same folder with settings.py) :

* scrapyelasticsearch.py
* transportNTLM.py

The final step is to configure the **meteo\_spider.py** under spiders folder (the name of the file comes from the name we state at :

scrapy genspider -t basic **meteo\_spider** meteo.gr at previous step.

**meteo\_spider.py file:**

import scrapy

import re

import scrapy.spiders

from ..items import MeteoItem

import datetime

class MeteoSpiderSpider(scrapy.Spider):

name = 'meteo\_spider'

allowed\_domains = ['http://penteli.meteo.gr/stations/tripoli/']

start\_urls = ['http://penteli.meteo.gr/stations/tripoli/']

def parse(self, response):

i=0

table = response.xpath('//\*[@id="table1"]')

rows = table.xpath('//tr')

source = 'meteo.gr'

city = 'Tripoli'

crawldate = datetime.datetime.now()

timestr = rows[2].xpath('td//text()')[3].extract()

datepart = timestr[-9:].strip()

timepart = timestr[2:-9].strip()

datetimep = datepart+' '+timepart

time = datetime.datetime(int('20'+datetimep[6:8]), int(datetimep[3:5]), int(datetimep[0:2]),int(datetimep[-5:-3]),int(datetimep[-2:])) - datetime.timedelta(hours=3, minutes=0)

temperature = float(rows[3].xpath('td//text()')[4].extract()[0:-2])

humidity = float(rows[4].xpath('td//text()')[4].extract()[:-1])

windends = (rows[6].xpath('td//text()')[4].extract()).find(" ")

winddire = (rows[6].xpath('td//text()')[4].extract()).find("at")

wind = float(rows[6].xpath('td//text()')[4].extract()[0:windends])

barends = rows[7].xpath('td//text()')[4].extract().find(" ")

barometer = float(rows[7].xpath('td//text()')[4].extract()[:barends])

yetos = float(rows[8].xpath('td//text()')[3].extract()[:-3])

direction = rows[6].xpath('td//text()')[4].extract()[winddire+3:]

id = source+' '+datetimep

item = MeteoItem()

item["id"] = id

item["source"] = source

item["time"] = time

item["timecrawl"] = crawldate

item["temperature"] = temperature

item["humidity"] = humidity

item["wind"] = wind

item["barometer"] = barometer

item["yetos"] = yetos

item["direction"] = direction

item["city"] = city

yield item

def start\_requests(self):

yield scrapy.Request('http://penteli.meteo.gr/stations/tripoli/', self.parse)

Now if we run the command : scrapy crawl meteo\_spider the crawl data transform into items and the items move into our index