# Scrape and Analytics Workshop

V1.0

# Agenda

Scrape and Analytics

- Preparation
- Scraping
- Data Analysis
- Assignments

# Part 1
# Workshop Preparation

# Environment

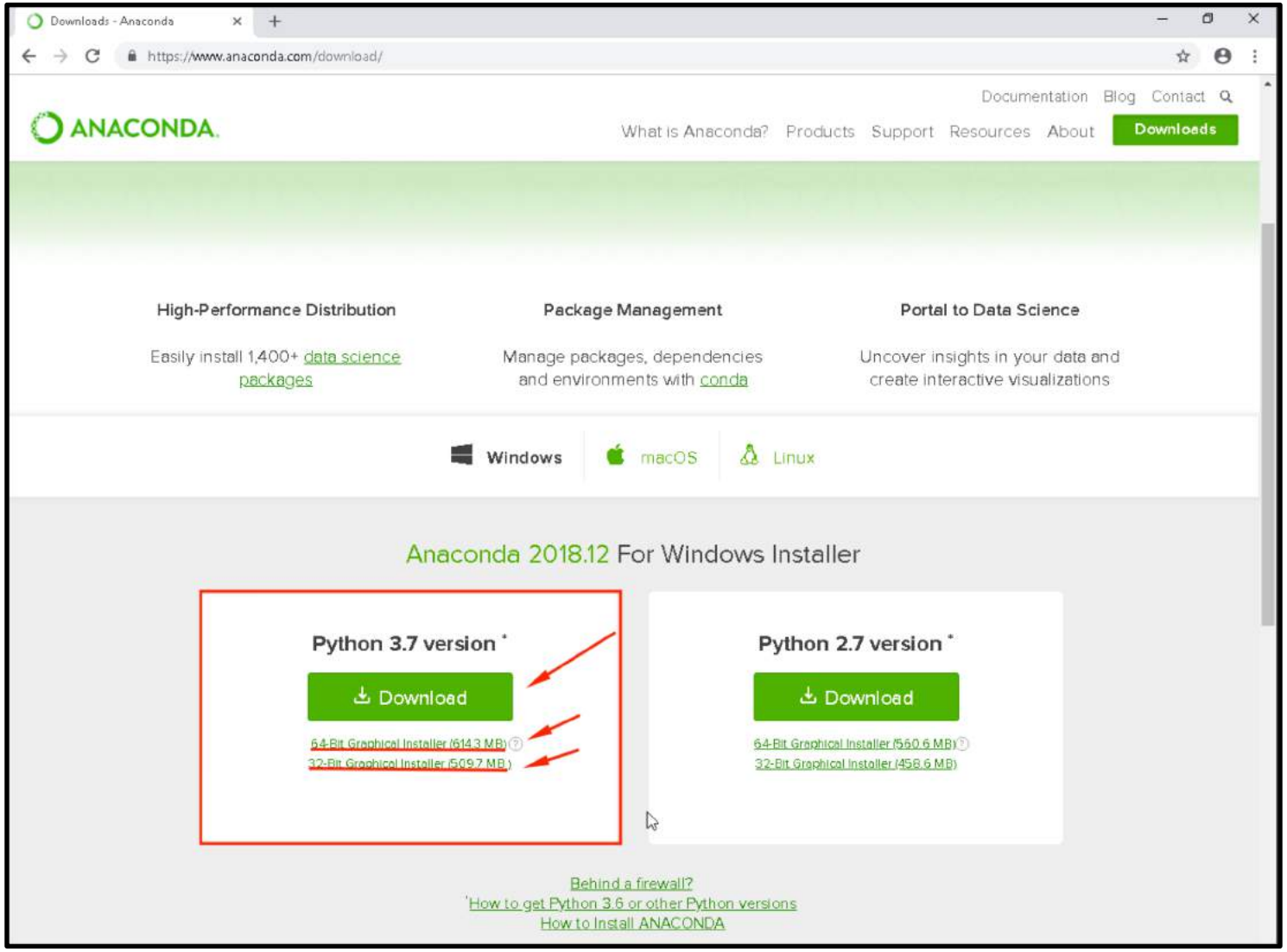
ANACONDA®

python™

Scrapy

jupyter

# www.anaconda.com



Anaconda3 2018.12 (64-bit) Setup

ANACONDA.

## Welcome to Anaconda3 2018.12 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2018.12 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Next > | Cancel

Anaconda3 2018.12 (64-bit) Setup

ANACONDA

## License Agreement

Please review the license terms before installing Anaconda3 2018.12 (64-bit).

Press Page Down to see the rest of the agreement.

```
==========================================
Anaconda End User License Agreement
==========================================

Copyright 2015, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2018.12 (64-bit).

Anaconda, Inc.

< Back | I Agree | Cancel

Anaconda3 2018.12 (64-bit) Setup

ANACONDA

## Choose Install Location

Choose the folder in which to install Anaconda3 2018.12 (64-bit).

Setup will install Anaconda3 2018.12 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Users\Roberto\Anaconda3 | Browse...

Space required: 2.8GB
Space available: 57.2GB

Anaconda, Inc.

< Back | Next > | Cancel

Anaconda3 2018.12 (64-bit) Setup

ANACONDA

## Select Installation Type

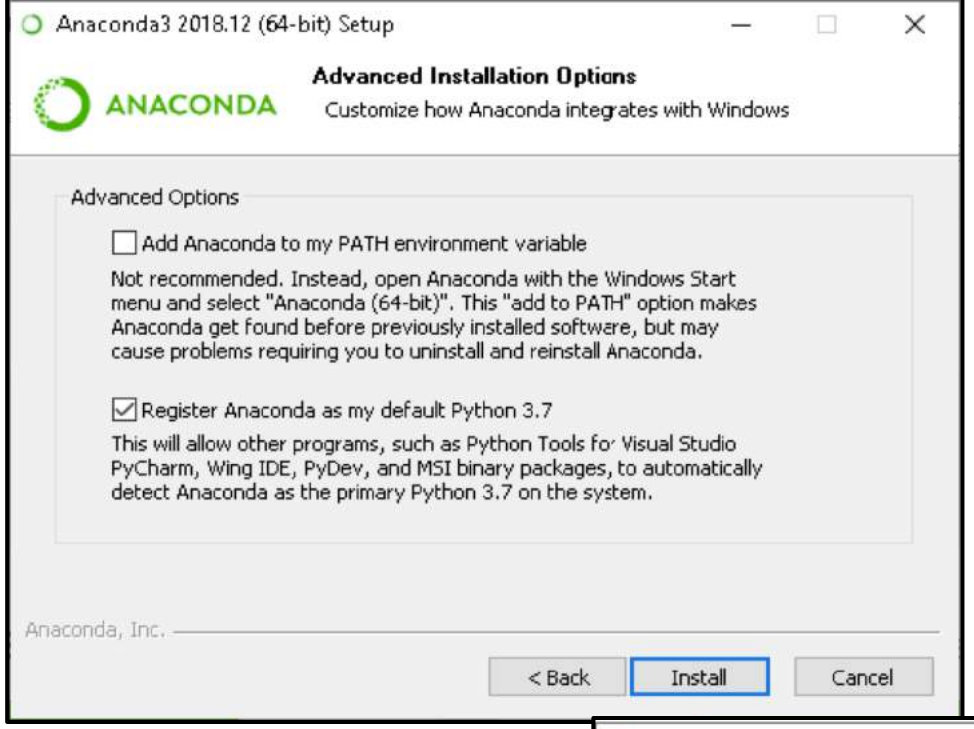
Please select the type of installation you would like to perform for Anaconda3 2018.12 (64-bit).

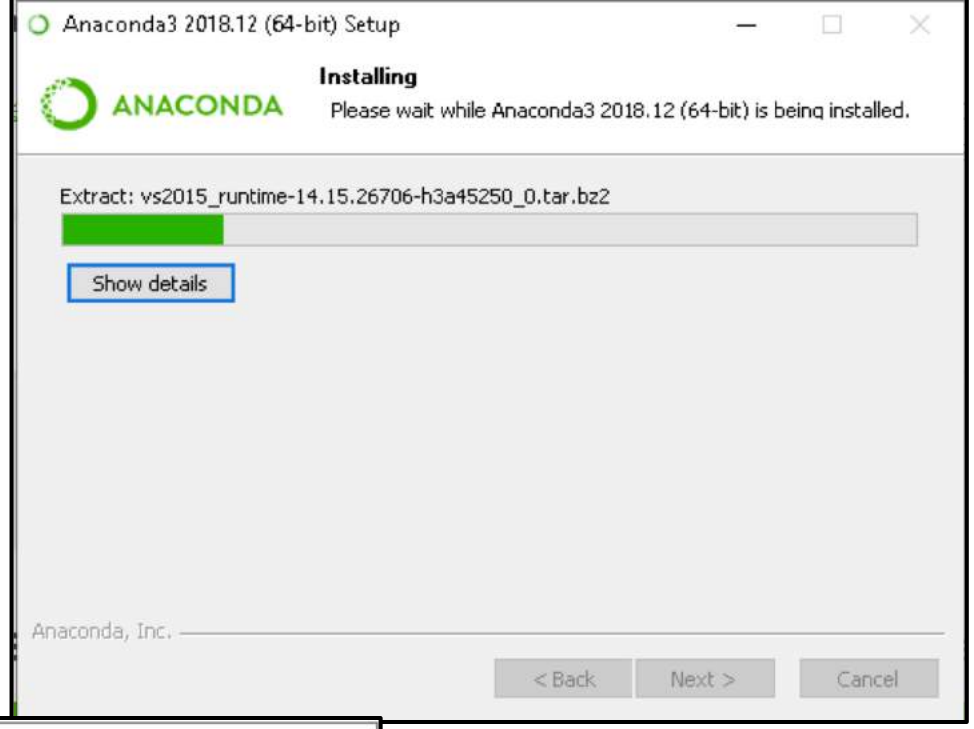
Install for:

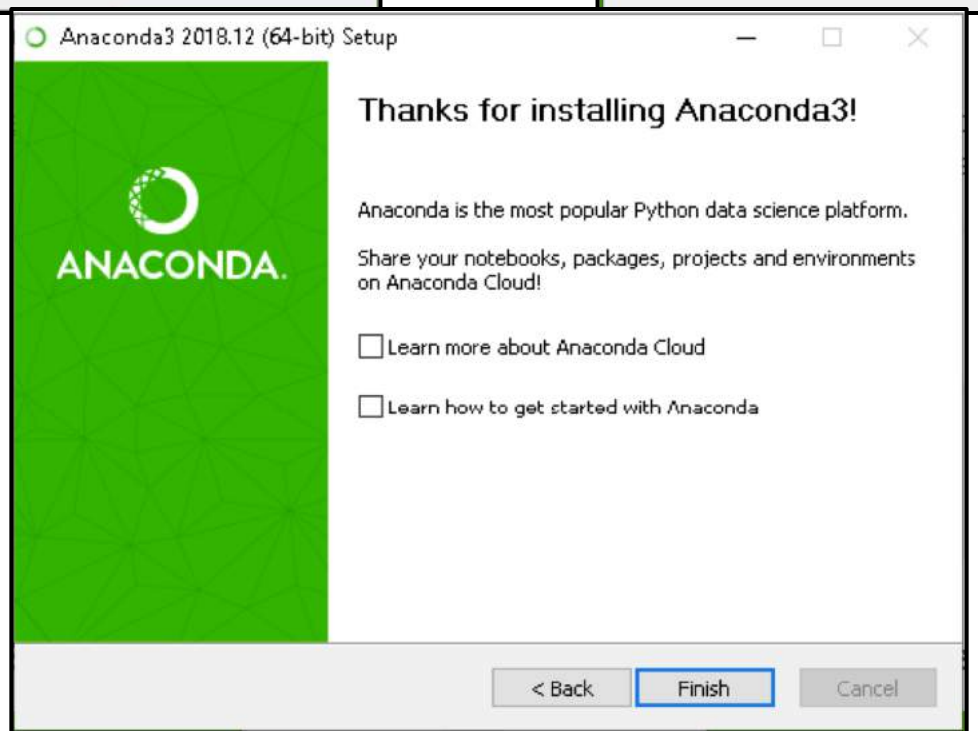
- (●) Just Me (recommended)
- ( ) All Users (requires admin privileges)

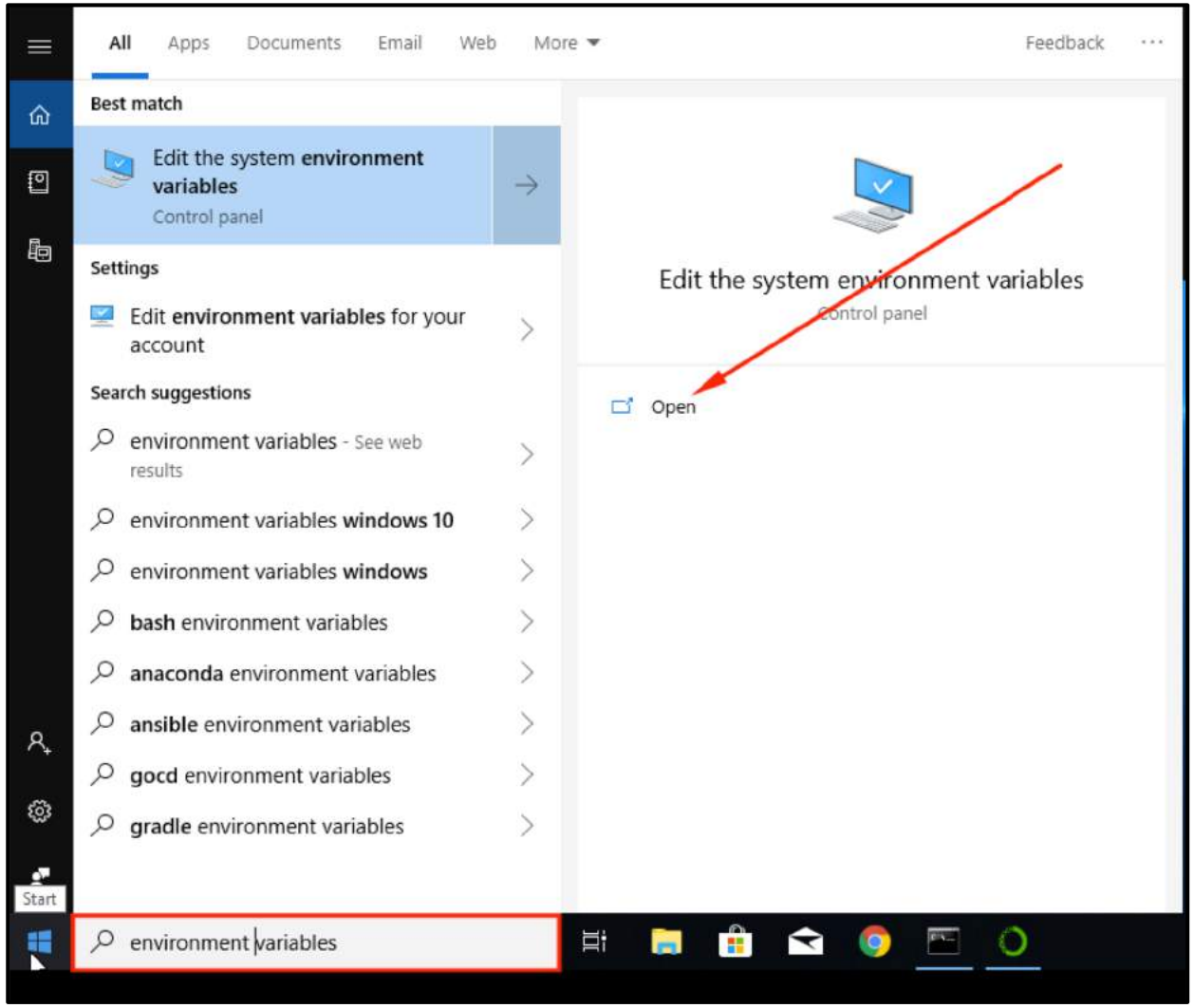
Anaconda, Inc.

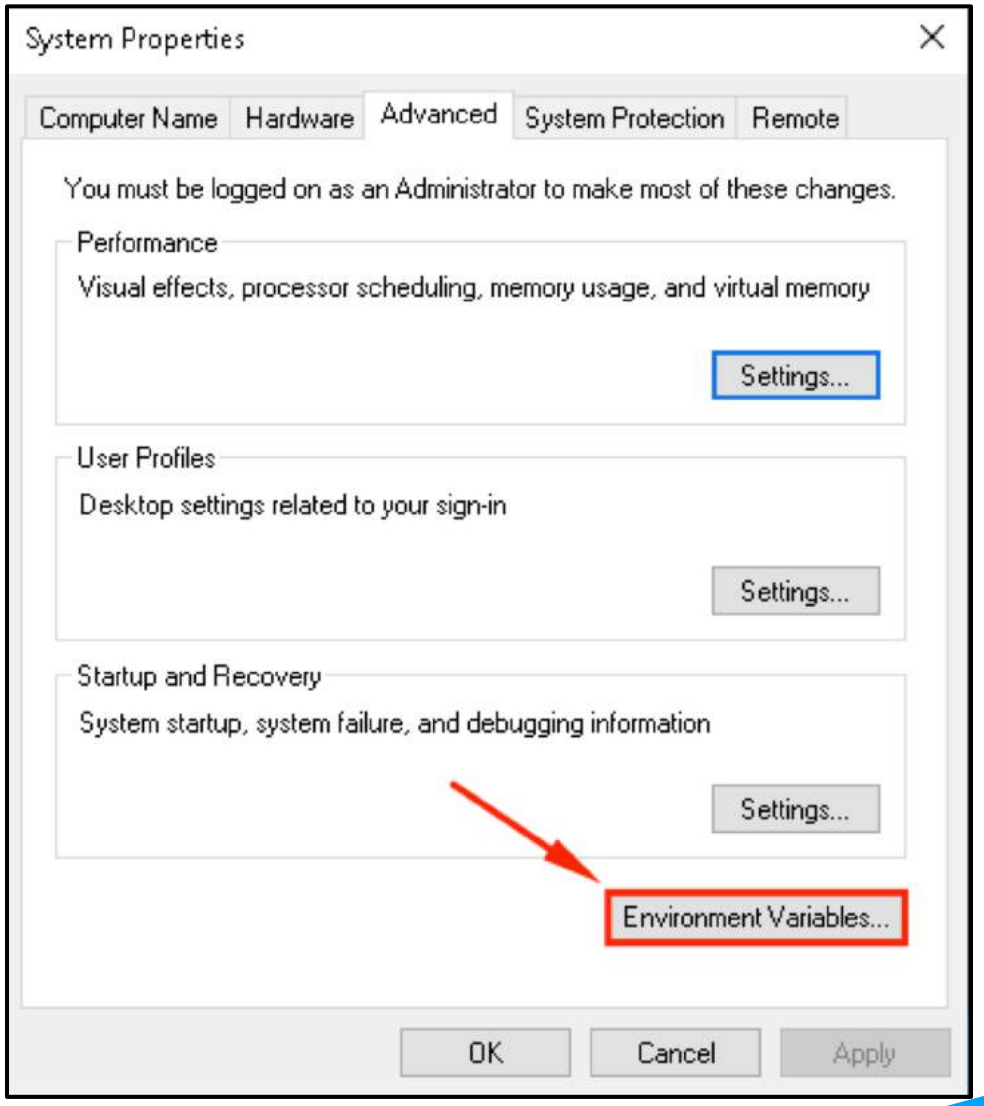
< Back | Next > | Cancel

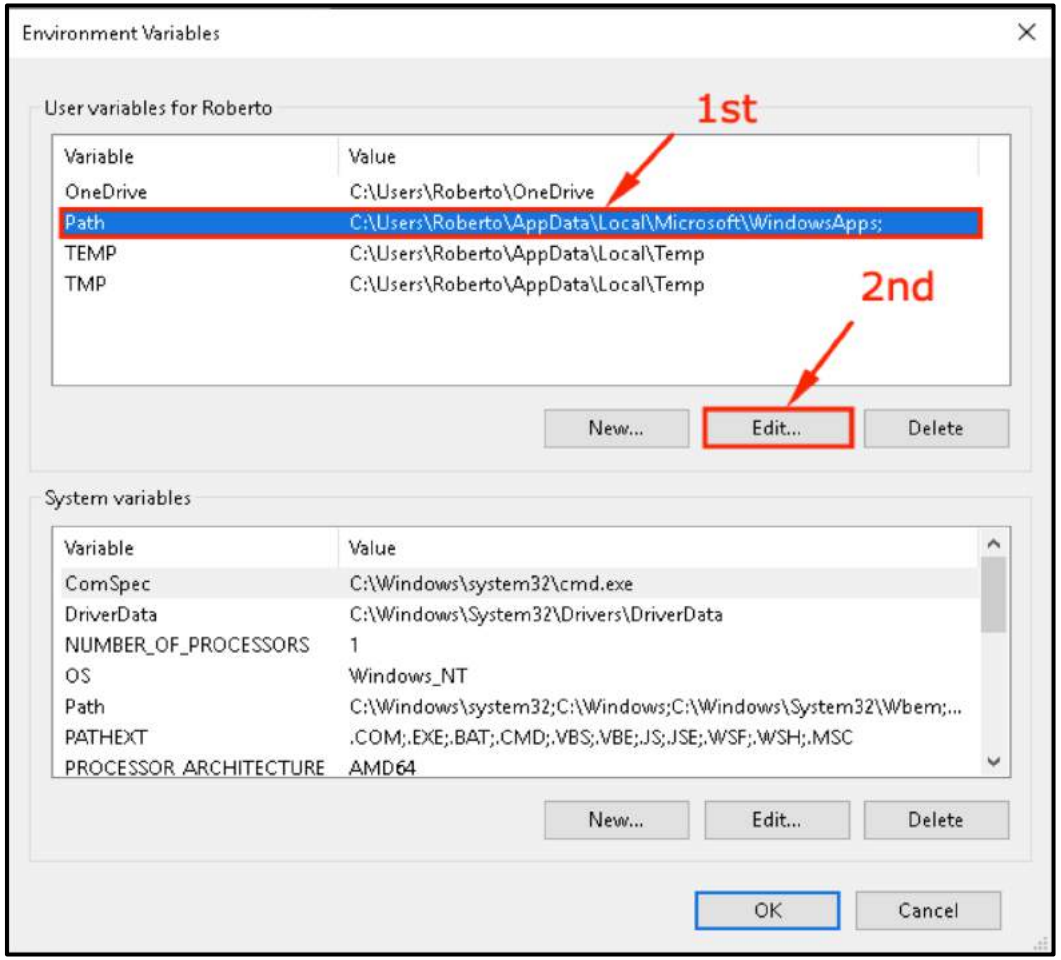

Anaconda3 2018.12 (64-bit) Setup
ANACONDA
Advanced Installation Options
Customize how Anaconda integrates with Windows
Advanced Options
Add Anaconda to my PATH environment variable
Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.
Register Anaconda as my default Python 3.7
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.
Anaconda, Inc.
< Back
Install
Cancel

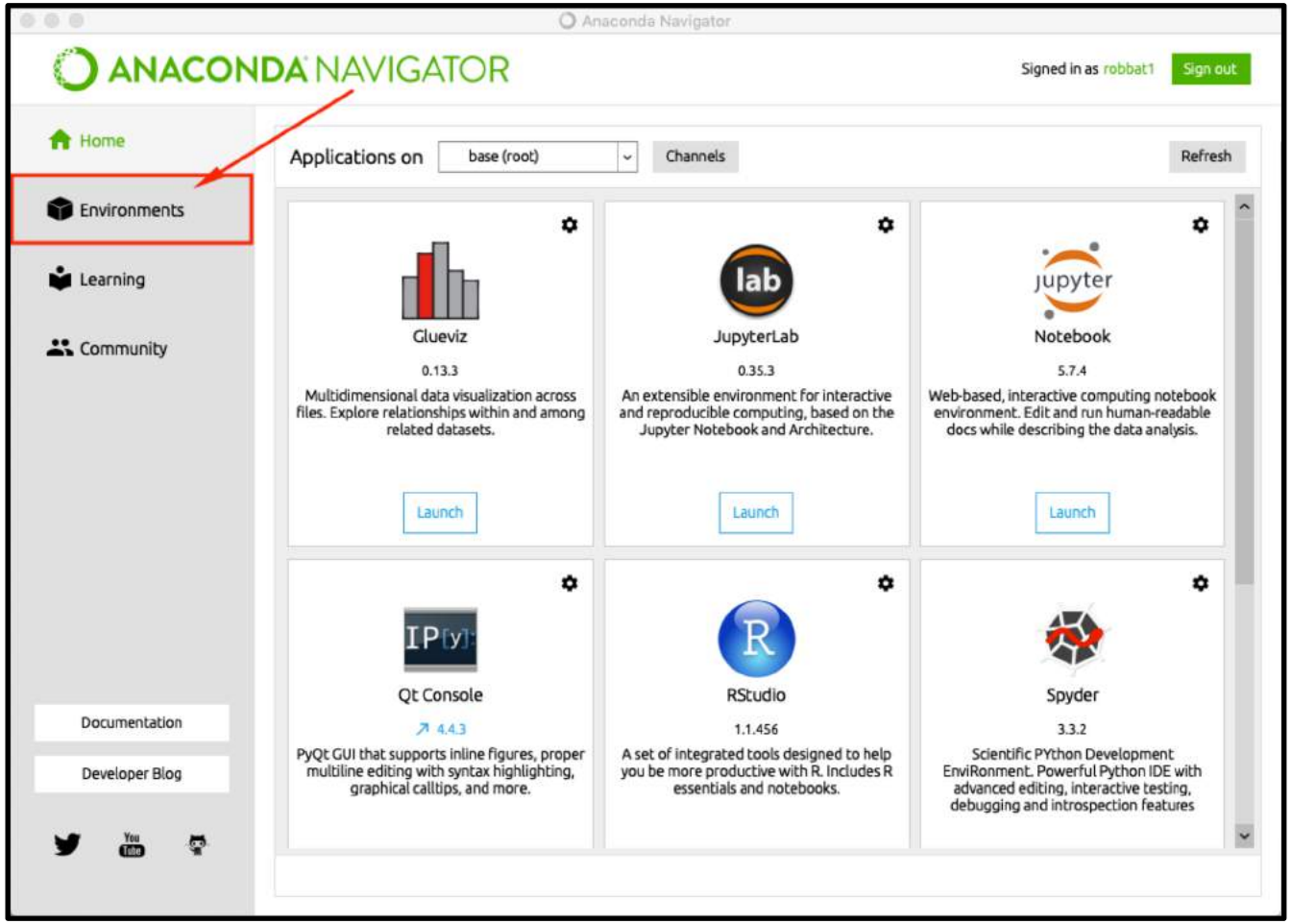

Anaconda3 2018.12 (64-bit) Setup
ANACONDA
Installing
Please wait while Anaconda3 2018.12 (64-bit) is being installed.
Extract: vs2015_runtime-14.15.26706-h3a45250_0.tar.bz2
Show details
Anaconda, Inc.
< Back
Next >
Cancel

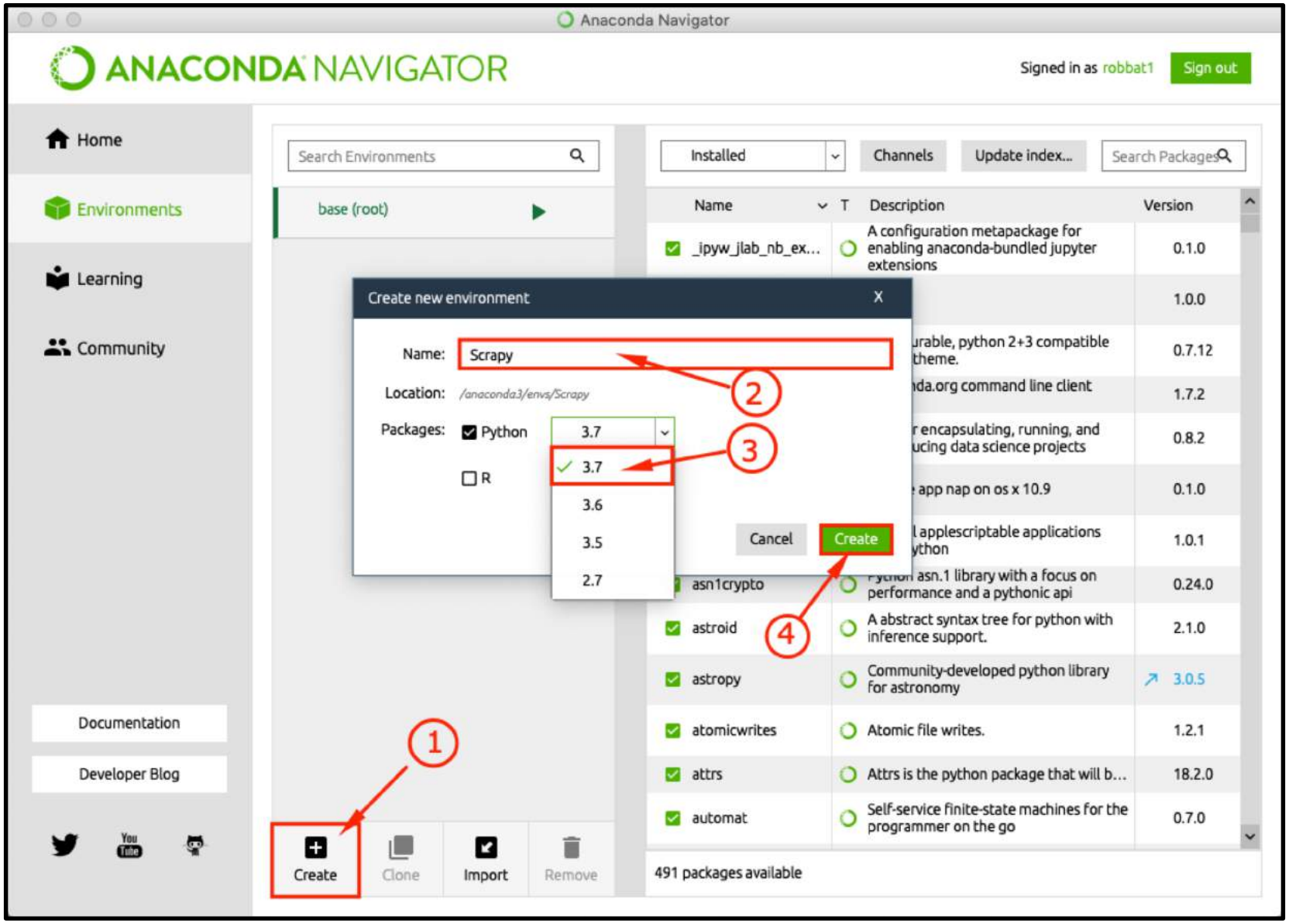

Anaconda3 2018.12 (64-bit) Setup
ANACONDA.
Thanks for installing Anaconda3!
Anaconda is the most popular Python data science platform.
Share your notebooks, packages, projects and environments on Anaconda Cloud!
Learn more about Anaconda Cloud
Learn how to get started with Anaconda
< Back
Finish
Cancel

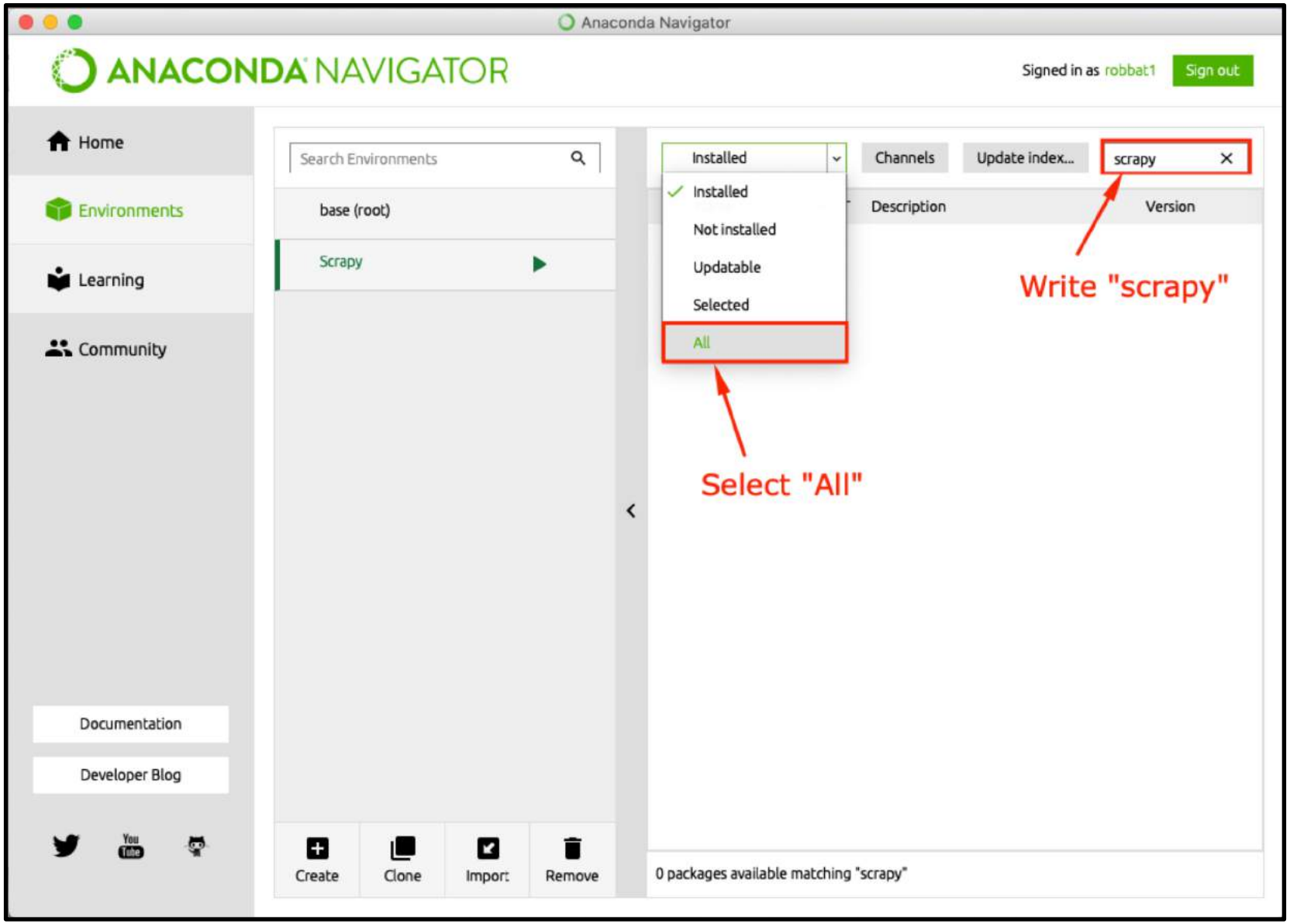

All
Apps
Documents
Email
Web
More
Feedback
Best match
Edit the system environment variables
Control panel
Settings
Edit environment variables for your account
Search suggestions
environment variables - See web results
environment variables windows 10
environment variables windows
bash environment variables
anaconda environment variables
ansible environment variables
gocd environment variables
gradle environment variables
Edit the system environment variables
Control panel
Open
Start
environment variables

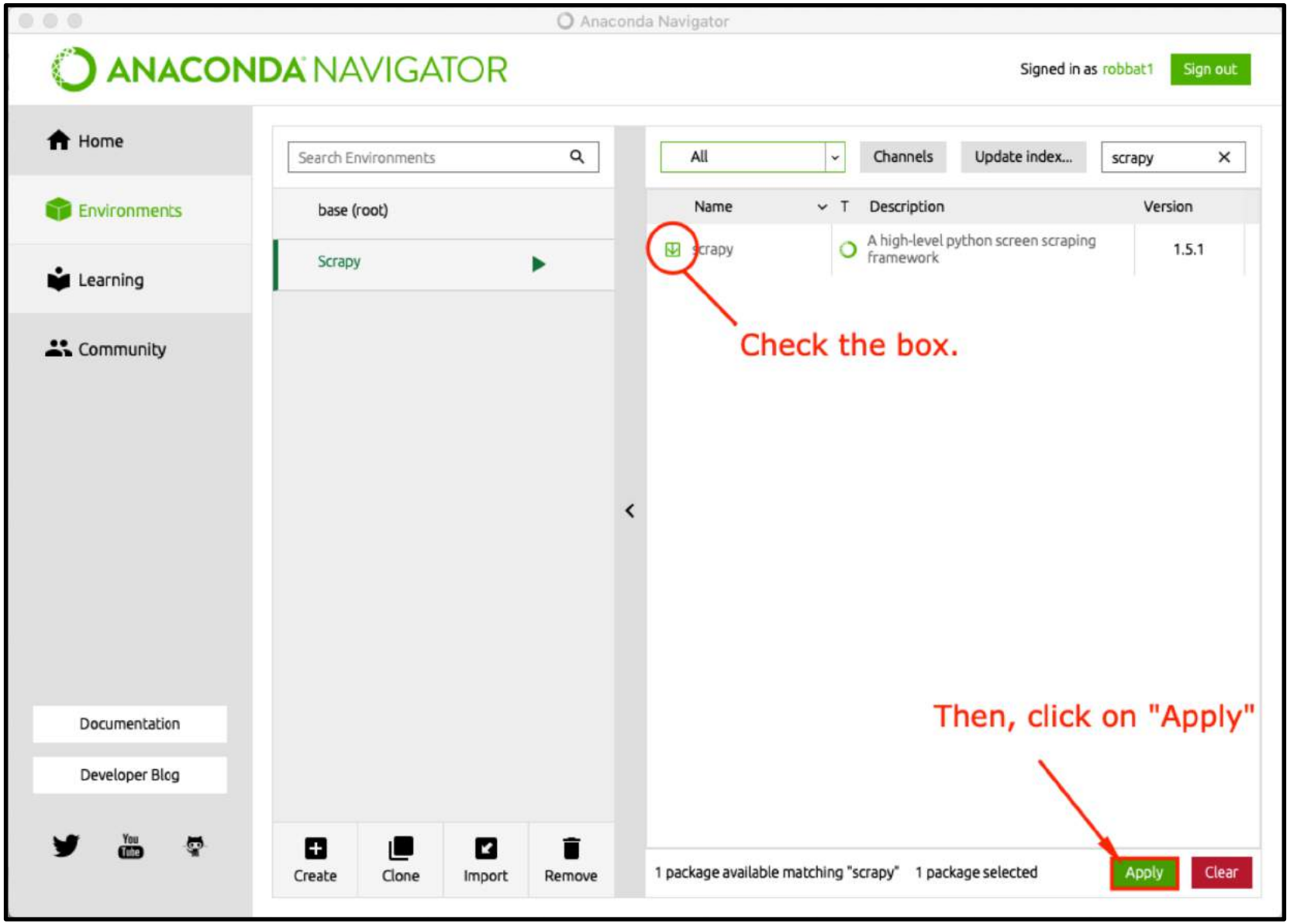

System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply

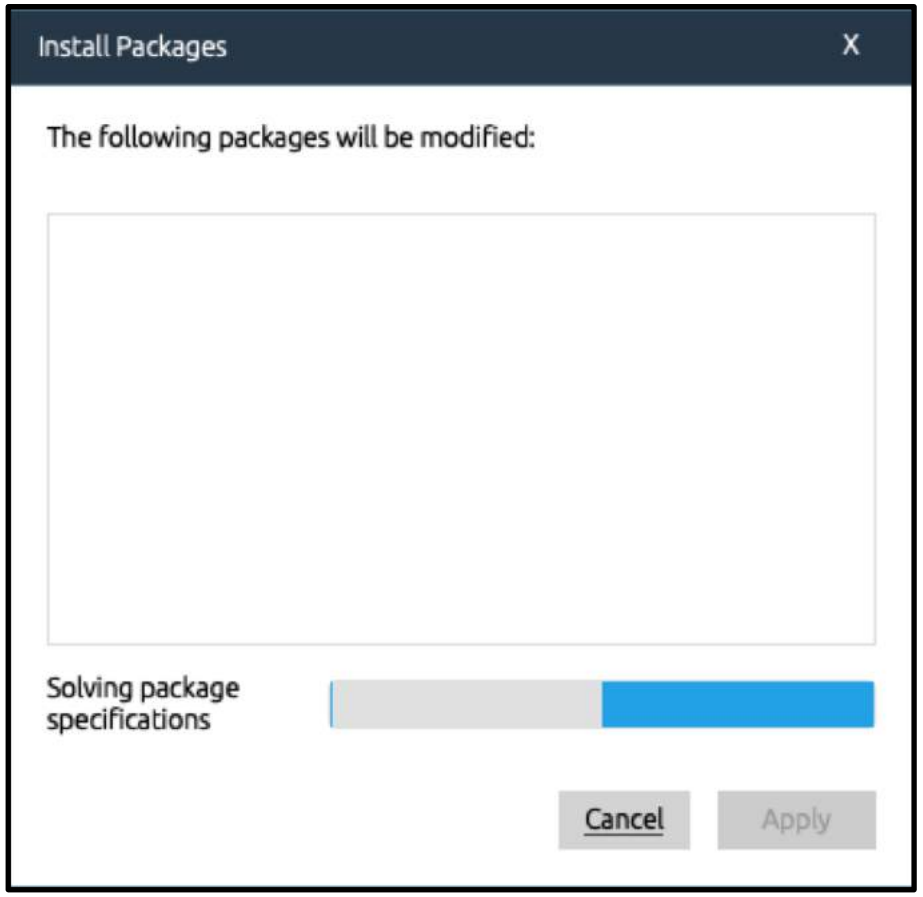

Environment Variables
User variables for Roberto
1st
Variable
Value
OneDrive
C:\Users\Roberto\OneDrive
Path
C:\Users\Roberto\AppData\Local\Microsoft\WindowsApps;
TEMP
C:\Users\Roberto\AppData\Local\Temp
TMP
C:\Users\Roberto\AppData\Local\Temp
2nd
New...
Edit...
Delete
System variables
Variable
Value
ComSpec
C:\Windows\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS
1
OS
Windows_NT
Path
C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE
AMD64
New...
Edit...
Delete
OK
Cancel

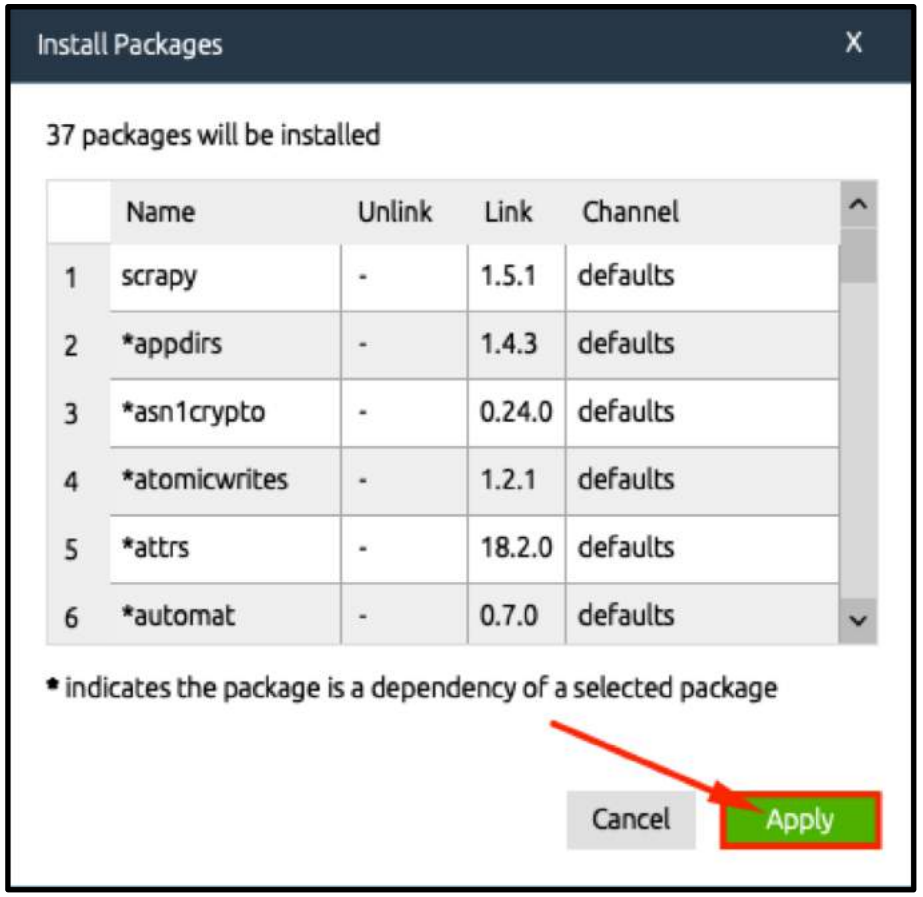

Anaconda Navigator
ANACONDA NAVIGATOR
Signed in as robbat1
Sign out
Home
Environments
Learning
Community
Documentation
Developer Blog
Applications on
base (root)
Channels
Refresh
Glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Launch
JupyterLab
0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
Notebook
5.7.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
Qt Console
4.4.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Spyder
3.3.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

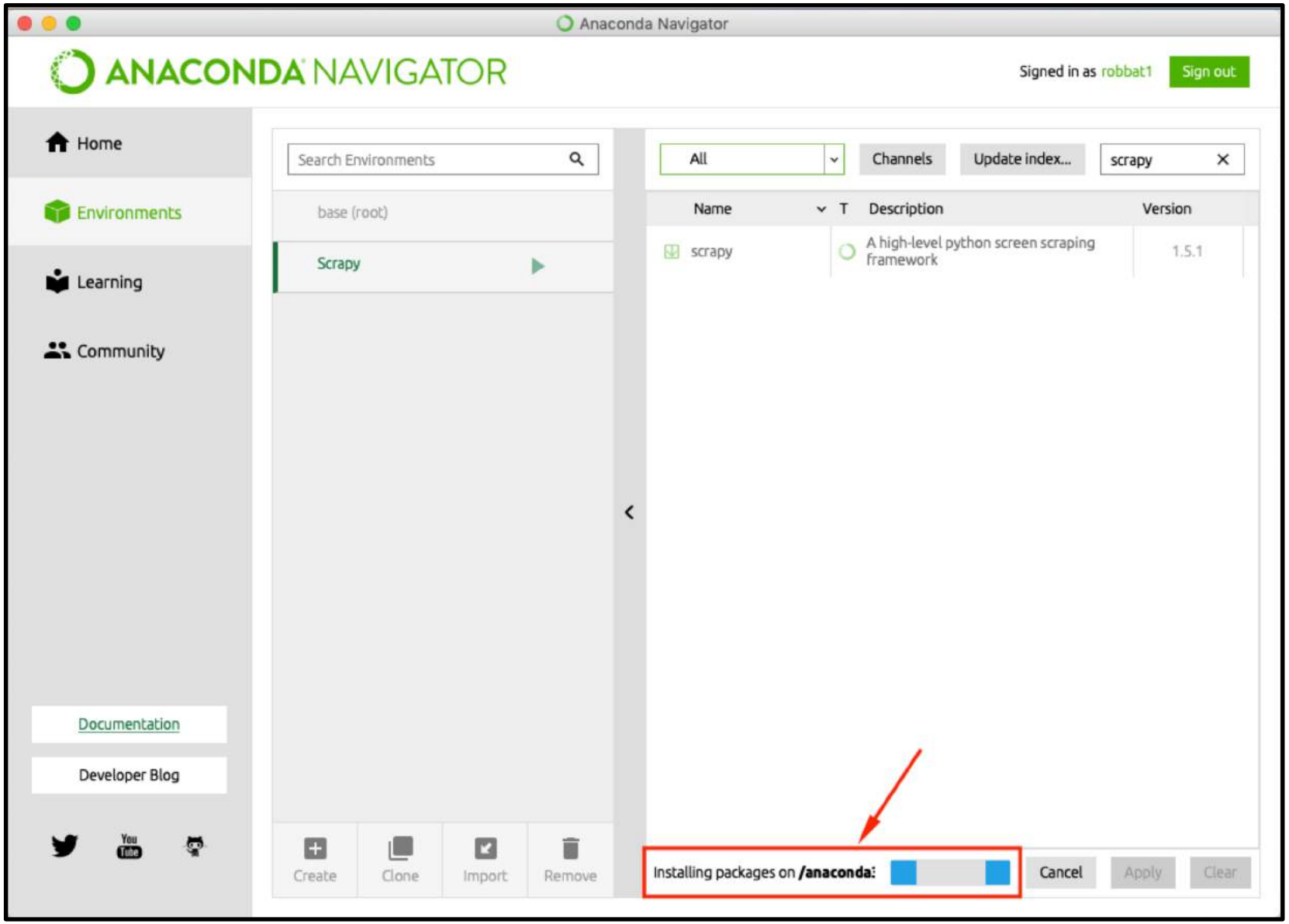

Anaconda Navigator
ANACONDA NAVIGATOR
Signed in as robbat1
Sign out
Home
Environments
Learning
Community
Documentation
Developer Blog
Search Environments
base (root)
Installed
Channels
Update index...
Search Packages
Name
T
Description
Version
_ipyw_jlab_nb_ex...
A configuration metapackage for enabling anaconda-bundled jupyter extensions
0.1.0
1.0.0
0.7.12
1.7.2
0.8.2
0.1.0
1.0.1
asn1crypto
0.24.0
astroid
A abstract syntax tree for python with inference support.
2.1.0
astropy
Community-developed python library for astronomy
3.0.5
atomicwrites
Atomic file writes.
1.2.1
attrs
Attrs is the python package that will b...
18.2.0
automat
Self-service finite-state machines for the programmer on the go
0.7.0
491 packages available
Create new environment
X
Name:
Scrapy
Location:
/anaconda3/envs/Scrapy
Packages:
Python
3.7
3.7
3.6
3.5
2.7
R
Cancel
Create
1
2
3
4
Create
Clone
Import
Remove

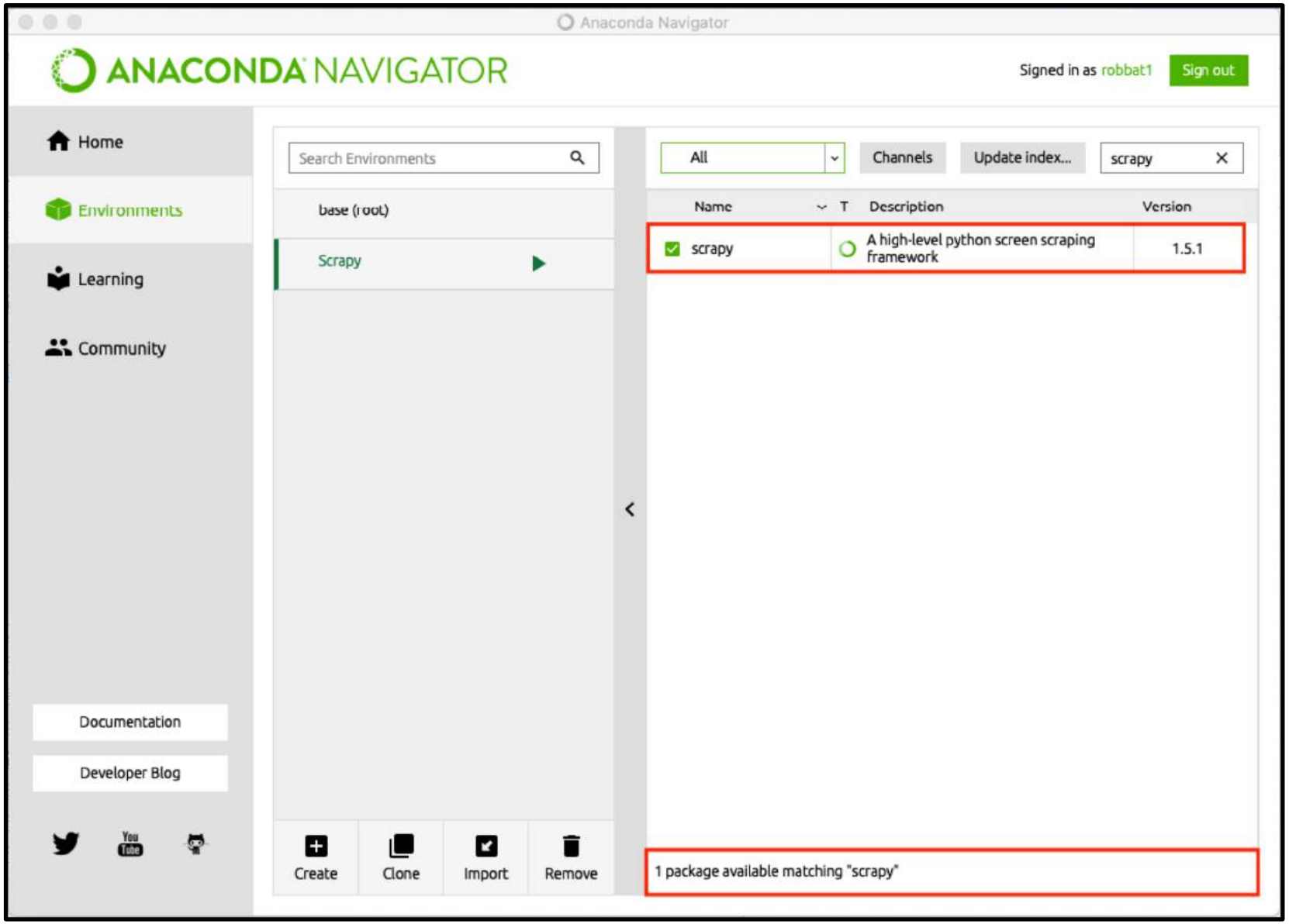

Anaconda Navigator
ANACONDA NAVIGATOR
Signed in as robbat1
Sign out
Home
Environments
Learning
Community
Documentation
Developer Blog
Search Environments
base (root)
Scrapy
Create
Clone
Import
Remove
Installed
Channels
Update index...
scrapy
Installed
Not installed
Updatable
Selected
All
Description
Version
Write "scrapy"
Select "All"
0 packages available matching "scrapy"

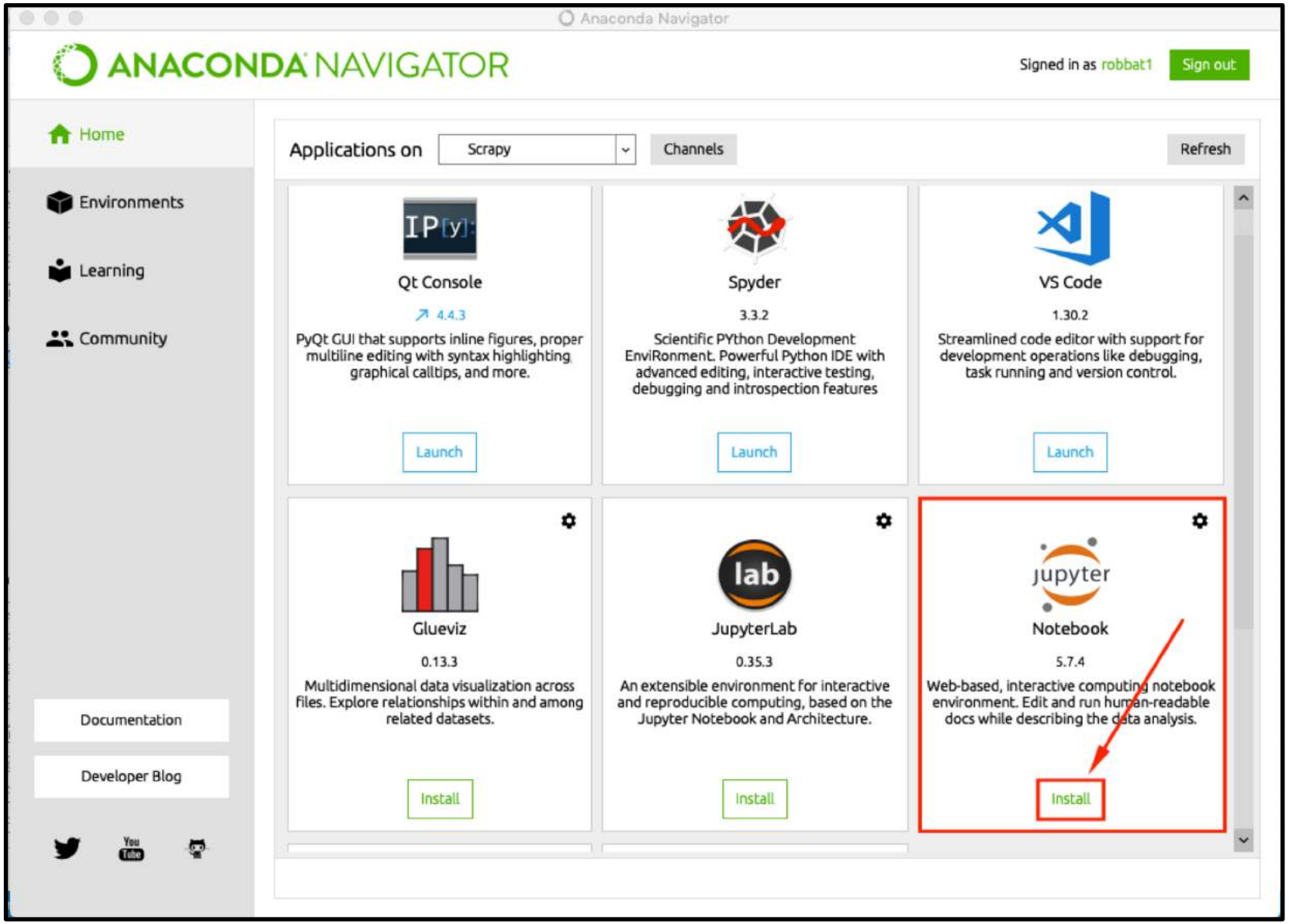

Anaconda Navigator
ANACONDA NAVIGATOR
Signed in as robbat1
Sign out
Home
Environments
Learning
Community
Documentation
Developer Blog
Search Environments
base (root)
Scrapy
Create
Clone
Import
Remove
All
Channels
Update index...
scrapy
Name
T
Description
Version
scrapy
A high-level python screen scraping framework
1.5.1
Check the box.
Then, click on "Apply"
1 package available matching "scrapy"
1 package selected
Apply
Clear

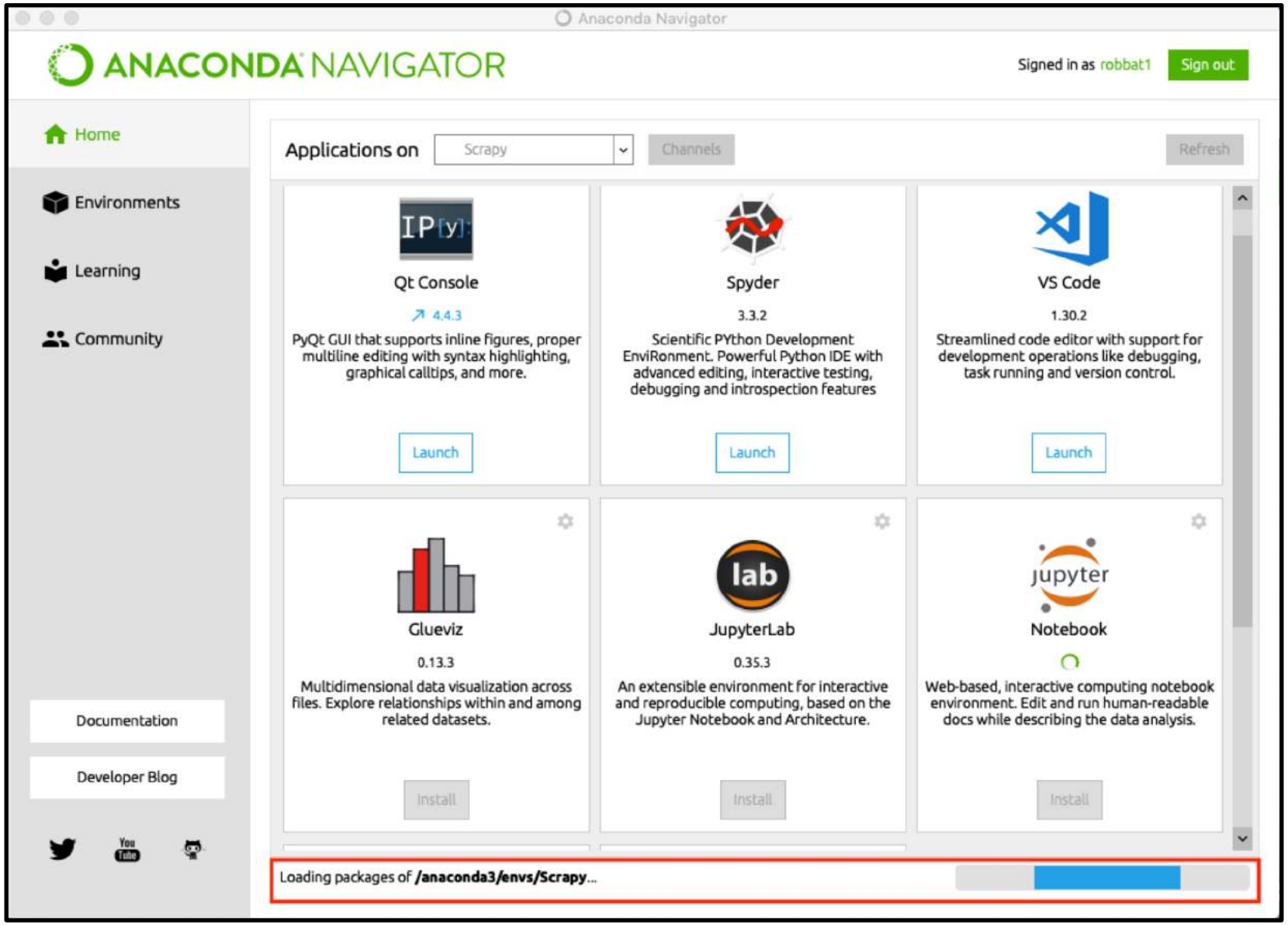

Install Packages
X
The following packages will be modified:
Solving package specifications
Cancel
Apply


Install Packages
X
37 packages will be installed
Name
Unlink
Link
Channel
1 scrapy - 1.5.1 defaults
2 *appdirs - 1.4.3 defaults
3 *asn1crypto - 0.24.0 defaults
4 *atomicwrites - 1.2.1 defaults
5 *attrs - 18.2.0 defaults
6 *automat - 0.7.0 defaults
* indicates the package is a dependency of a selected package
Cancel
Apply

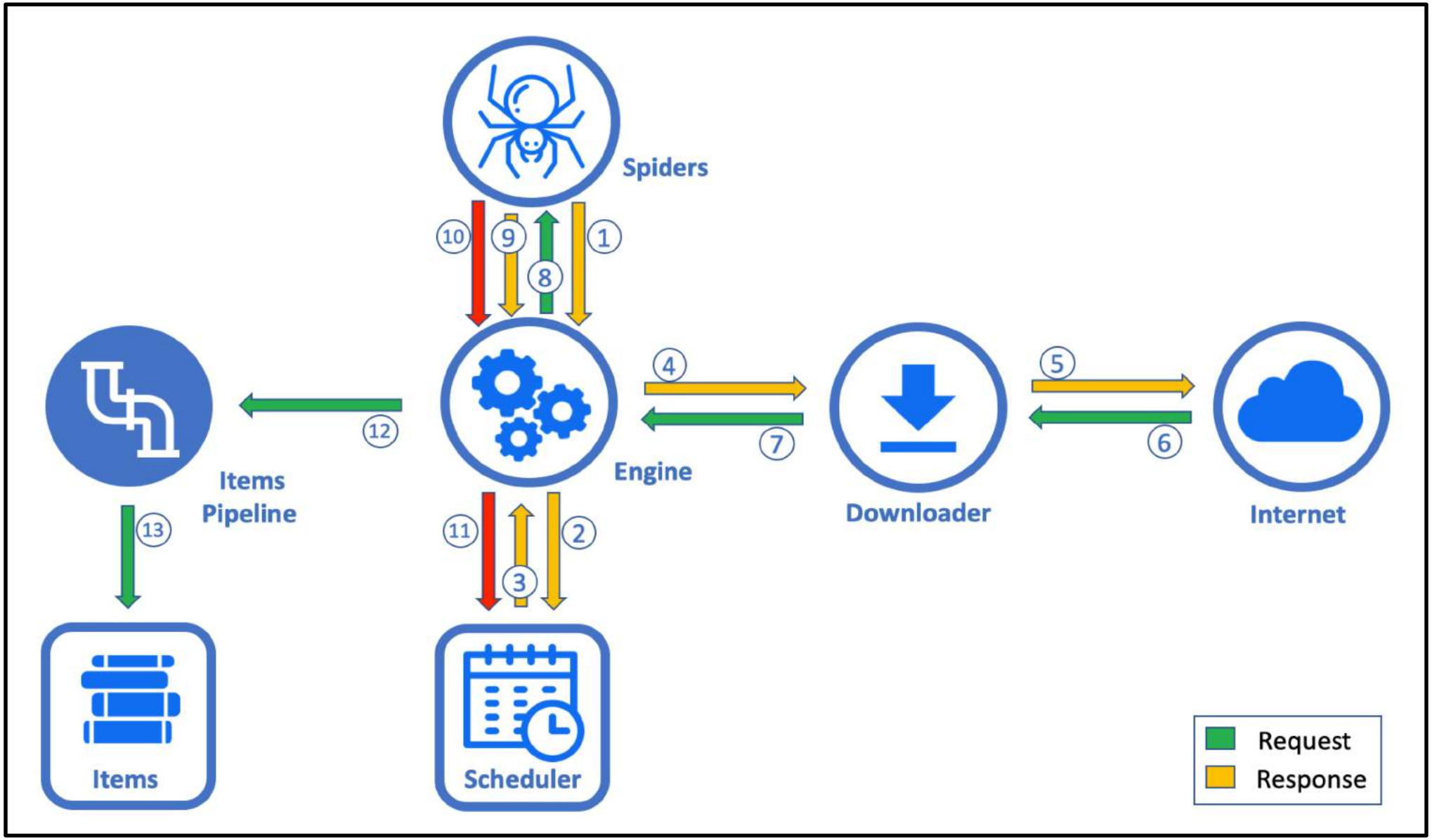

Anaconda Navigator
ANACONDA NAVIGATOR
Signed in as robbat1
Sign out
Home
Environments
Learning
Community
Documentation
Developer Blog
Search Environments
base (root)
Scrapy
Create
Clone
Import
Remove
All
Channels
Update index...
scrapy
Name
T
Description
Version
scrapy
A high-level python screen scraping framework
1.5.1
Installing packages on /anaconda:
Cancel
Apply
Clear


Anaconda Navigator
ANACONDA NAVIGATOR
Signed in as robbat1
Sign out
Home
Environments
Learning
Community
Documentation
Developer Blog
Search Environments
base (root)
Scrapy
Create
Clone
Import
Remove
All
Channels
Update index...
scrapy
Name
T
Description
Version
scrapy
A high-level python screen scraping framework
1.5.1
1 package available matching "scrapy"

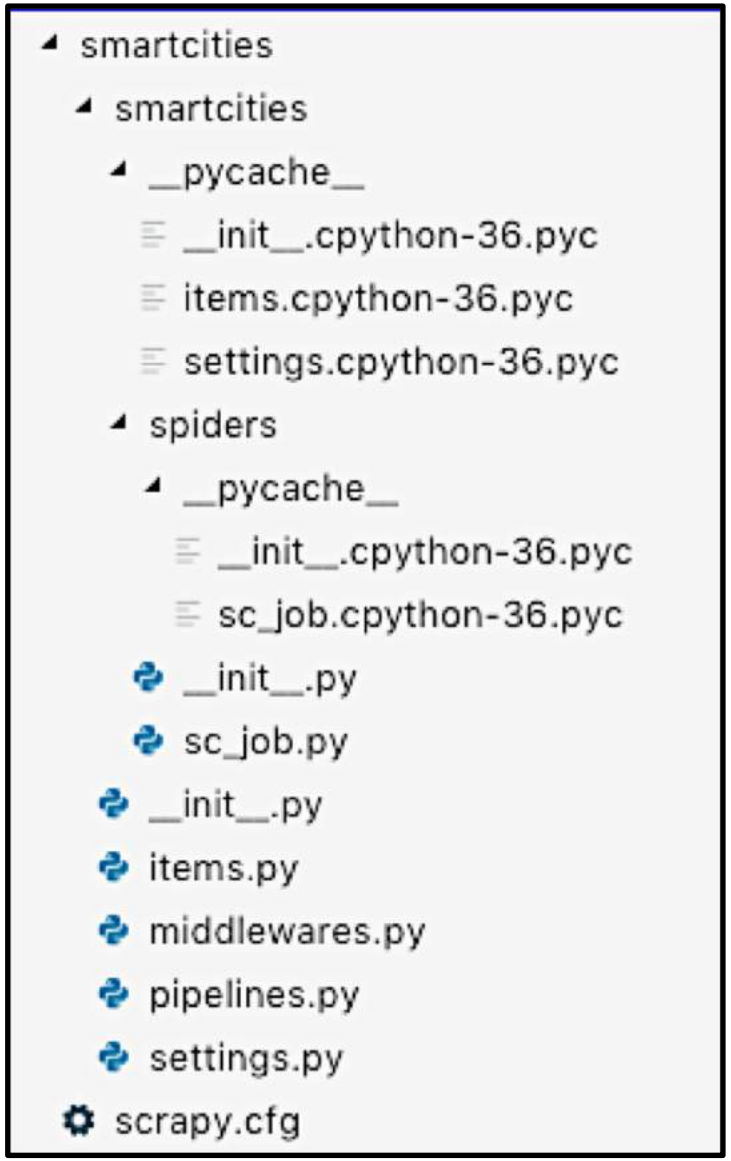

Anaconda Navigator
ANACONDA NAVIGATOR
Signed in as robbat1
Sign out
Home
Environments
Learning
Community
Documentation
Developer Blog
Applications on
Scrapy
Channels
Refresh
Qt Console
4.4.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
Spyder
3.3.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
VS Code
1.30.2
Streamlined code editor with support for development operations like debugging, task running and version control.
Launch
Glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install
JupyterLab
0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Install
Notebook
5.7.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Install


Anaconda Navigator
ANACONDA NAVIGATOR
Signed in as robbat1
Sign out
Home
Environments
Learning
Community
Documentation
Developer Blog
Applications on
Scrapy
Channels
Refresh
Qt Console
4.4.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
Spyder
3.3.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
VS Code
1.30.2
Streamlined code editor with support for development operations like debugging, task running and version control.
Launch
Glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install
JupyterLab
0.35.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Install
Notebook
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Install
Loading packages of /anaconda3/envs/Scrapy...

# Part 2
# Scraping Introduction


Spiders
10
9
8
1
Items
Pipeline
12
Engine
4
7
Downloader
5
6
Internet
13
11
3
2
Items
Scheduler
Request
Response

# Web site investigation


smartcities
smartcities
__pycache__
__init__.cpython-36.pyc
items.cpython-36.pyc
settings.cpython-36.pyc
spiders
__pycache__
__init__.cpython-36.pyc
sc_job.cpython-36.pyc
__init__.py
sc_job.py
__init__.py
items.py
middlewares.py
pipelines.py
settings.py
scrapy.cfg

```
$ scrapy shell
```

After execute the command above the final data showed in the terminal is the available Scrapy objects in the shell environment:

```
[s] Available Scrapy objects:
[s]   scrapy     scrapy module (contains scrapy.Request, scrapy.Selector, etc)
[s]   crawler    <scrapy.crawler.Crawler object at 0x1092933c8>
[s]   item       {}
[s]   settings   <scrapy.settings.Settings object at 0x10b825898>
[s] Useful shortcuts:
[s]     fetch(url[, redirect=True]) Fetch URL and update local objects (by
        default, redirects are followed)
[s]     fetch(req)          Fetch a scrapy.Request and update local objects
[s]     shelp()             Shell help (print this help)
[s]     view(response)      View response in a browser
```

## Step One - Fetching the page

Once on the Scrapy Shell ">>>" we will use the 'fetch' command (short cut) to fetch the target web site. Which in this case will be the following search from www.ebay.com.

```
'https://www.ebay.com/sch/i.html?_from=R40&_nkw=heart+rate+monitor&_sacat=0&L
H_ItemCondition=3&rt=nc&LH_BIN=1&_ipg=200'
```

```
>>>
fetch("https://www.ebay.com/sch/i.html?_from=R40&_nkw=heart+rate+monitor&_sac
at=0&LH_ItemCondition=3&rt=nc&LH_BIN=1&_ipg=200")
2019-01-14 07:56:13 [scrapy.core.engine] INFO: Spider opened
2019-01-14 07:56:15 [scrapy.core.engine] DEBUG: Crawled (200) <GET
https://www.ebay.com/sch/i.html?_from=R40&_nkw=heart+rate+monitor&_sacat=0&LH
_ItemCondition=3&rt=nc&LH_BIN=1&_ipg=200> (referer: None)
```

```
>>> response.url
'https://www.ebay.com/sch/i.html?_from=R40&_nkw=heart+rate+monitor&_sacat=0&L
H_ItemCondition=3&rt=nc&LH_BIN=1&_ipg=200'
```

We could identify that the URL used by fetch was correct. The next verification is to retrieve the web page content.

```
>>> view(response)
True
```

Hi! Sign in My eBay

ebay heart rate monitor

Related: heart rate monitor **watch** heart rate monitor **polar** heart rate monitor **garmi...**

All Listings | Accepts Offers | Auction | **Buy It Now**

Best Match

71,569 results Save this search

New

h3.s-item__title | 300.39×38

New GARMIN HRM1G Run Sport Heart Rate Monitor With Chest Strap ANT+ USA Sent

Brand New

12 product ratings

$14.80

or Best Offer

+$2.80 shipping

638 Sold

2 new & refurbished from $14.80

F1 Fitness Blood Pressure Tracker Heart Rate Monitor Smart Wrist Bracelet Watch

Brand New

Elements Console Sources Network Performance Memory » 33

```
1806-11e9-a19e-74dbd180f094"}}" _sp="p2351460.m1686.l7400">
... <h3 class="s-item__title" role="text">New GARMIN HRM1G Run Sport Heart
    Rate Monitor With Chest Str
</a>
```

... div #srp-river-main #srp-river-results ul #srp-rive .s-item__title

Styles Event Listeners DOM Breakpoints Properties

Filter

```
element.style {
}
.s-item__link:visited .s-item__title {                    search-page-lar
  color: #6a29b9;
}
.s-item__link:visited>.s-item__title {                    search-page-lar
  color: #6a29b9;
}
.s-page .s-item__link>.s-item__title {                    search-page-lar…9-3f1c83.css:1
  color: inherit;
}
@media only screen and (min-width: 480px)
.srp-list .s-item__title {                                search-page-lar…9-3f1c83.css:1
  max-height: none;
```

Add attribute
Edit as HTML
Delete element
Copy
Hide element
Force state
Break on
Expand recursively
Collapse children
Scroll into view
Focus
Store as global variable

Cut element
Copy element
Paste element
Copy outerHTML
Copy selector
Copy XPath

Filter Show all

box-sizing border…
color rgb(…
cursor pointer
display -webki…
font-family "Helvet…

Console What's New

top Filter Default levels 1 hidden

```
Failed to load resource: the server responded with a status of 404 ()   choices.truste.com/c…ext/javascript%22:1
Failed to load resource: the server responded with a status of 404 ()   choices.truste.com/c…ext/javascript%22:1
Failed to load resource: the server responded with a status of 404 ()   choices.truste.com/c…ext/javascript%22:1
Failed to load resource: the server responded                           choices.truste.com/c…ext/javascript%22:1
```

The expression copied will be used in the Scrapy Shell to test if we in fact can extract the content of interest. Expression:

```
#srp-river-results-listing1 > div > div.s-item__info.clearfix > a > h3
```

Using the command response, we will test the expression above:

```
>>> response.css('#srp-river-results-listing1 > div > div.s-
item__info.clearfix > a > h3')
[<Selector xpath="descendant-or-self::*[@id = 'srp-river-results-
listing1']/div/div[@class and contains(concat(' ', normalize-space(@class), '
'), ' s-item__info ') and (@class and contains(concat(' ', normalize-
space(@class), ' '), ' clearfix '))]/a/h3" data='<h3 class="s-item__title"
role="text">Ne'>]
```

The result given is not generalized to all the titles, but to the specific title we copy the selector. In order to make it generalized it is necessary to remove the parent node which is “#srp-river-results-listing1 >”. Let’s execute without this part.

```
>>> response.css('div > div.s-item__info.clearfix > a > h3')
[<Selector xpath="descendant-or-self::div/div[@class and contains(concat(' ',
normalize-space(@class), ' '), ' s-item__info ') and (@class and
contains(concat(' ', normalize-space(@class), ' '), ' clearfix '))]/a/h3"
data='<h3 class="s-item__title" role="text">Ne'>,
[...]
```

```
{" s-item__info ') and (@class and contains(concat(' ', normalize-
space(@class), ' '), ' clearfix '))]/a/h3" data='<h3 class="s-item__title s-
item__title--'>]
```

The result above just show the first and last line of response. We can identify which the response potentially covers all the titles present in the document. The next step is to add the option '::text' in the expression and a function called 'extract()' in order to clean-up the data retrieved.

```
>>> response.css('div > div.s-item__info.clearfix > a > h3::text').extract()
['New GARMIN HRM1G Run Sport Heart Rate Monitor With Chest Strap ANT+ USA
Sent', 'Sports Blood Pressure/Heart Rate Monitor Fitness Smart Watch Wrist
Band Bracelet', 'F1 Fitness Blood Pressure Tracker Heart Rate Monitor Smart
Wrist Bracelet Watch', 'Sports Blood Pressure/Heart Rate Monitor Fitness
Smart Watch Wrist Band Bracelet',
```

The response above which shows few lines that the text titles extracted from the page. The next step is to collect the price from each item in the web page. Similar to the process of get the selector of the title we will proceed with the price.

```
>>> response.css('#srp-river-results-listing1 > div > div.s-
item__info.clearfix > div.s-item__details.clearfix > div:nth-child(1) >
span')
[<Selector xpath="descendant-or-self::*[@id = 'srp-river-results-
listing1']/div/div[@class and contains(concat(' ', normalize-space(@class), '
'), ' s-item__info ') and (@class and contains(concat(' ', normalize-
space(@class), ' '), ' clearfix '))]/div[@class and contains(concat(' ',
normalize-space(@class), ' '), ' s-item__details ') and (@class and
contains(concat(' ', normalize-space(@class), ' '), ' clearfix
'))]/div[count(preceding-sibling::*) = 0]/span" data='<span class="s-
item__price">$14.80</span'>]
```

Once again the response was specific to the selected price, now we will generalize the expression removing the parent node, and add the entire command with '::text' and the function 'extract()'.

```
>>> response.css('div > div.s-item__info.clearfix > div.s-
item__details.clearfix > div:nth-child(1) > span::text').extract()
['$14.80', '$15.99', '$14.81', '$14.92', '$15.99', '$13.45', '$22.95',
'$13.45', '$13.50', '$49.99', '$22.78', '$56.49', '$8.90', '$10.99',
'$13.49', '$16.18', '$13.58', '$14.24',
```

# Scraping

## Step 1 - Create a Project Directory

After we test our selector, we will create our Scrapy project in a new folder that we will create, for this example was created the folder 'SCRAPY_PROJECT' in '/Users/username/'.

## Step 2 - Create a Project

To create a new Scrapy project we will use the command scrapy startproject <project_name> [project_dir], in this case we will use 'ebay' as the 'project_name', and the default directory by omitting the argument 'project_dir'.

```
$ scrapy startproject ebay
New Scrapy project 'ebay', using template directory
'/anaconda3/lib/python3.6/site-packages/scrapy/templates/project', created
in:
    /Users/user_name/SCRAPY_PROJECTS/ebay

You can start your first spider with:
    cd ebay
    scrapy genspider example example.com
```

## Step 3 – Create a Spider

After create the project we will create our first spider inside of the directory defined (/Users/user_name/SCRAPY_PROJECTS/ebay ) using the command scrapy genspider [options] <name> <domain>

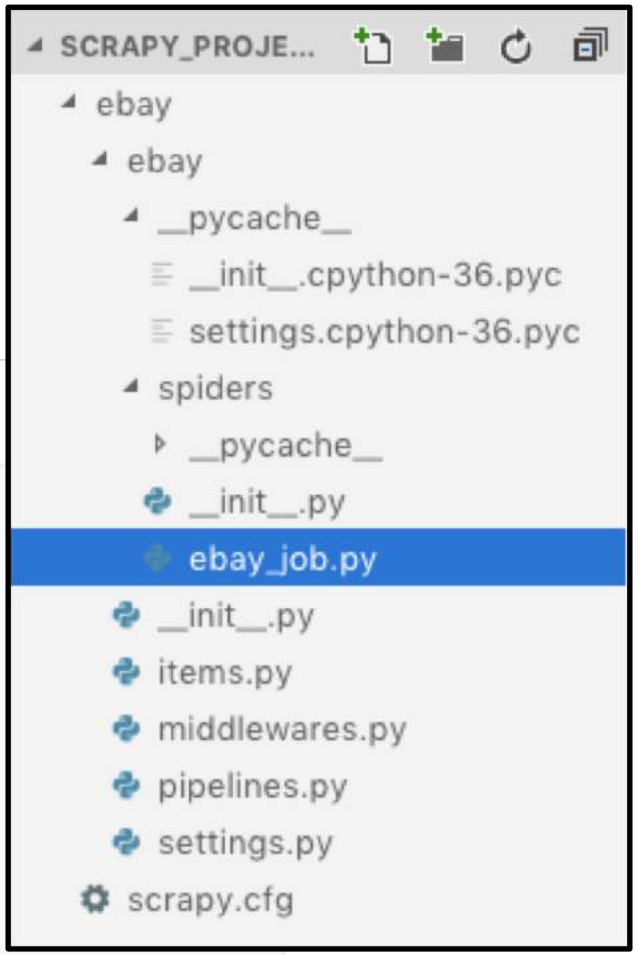
```
$ scrapy genspider ebay_job ebay.com
    Created spider 'ebay_job' using template 'basic' in module:
ebay.spiders.ebay_job
```

Opening the file 'ebay_job.py' we can see the default content created:

```
# -*- coding: utf-8 -*-
import scrapy

class EbayJobSpider(scrapy.Spider):
    name = 'ebay_job'
    allowed_domains = ['ebay.com']
    start_urls = ['http://ebay.com/']

    def parse(self, response):
        pass
```




We will edit few components in this file. First it is necessary to substitute the 'start_url' by the link we investigated previously:

```
'https://www.ebay.com/sch/i.html?_from=R40&_nkw=heart+rate+monitor&_sacat=0&LH_ItemCondition=3&rt=nc&LH_BIN=1&_ipg=200'
```

```
# -*- coding: utf-8 -*-
import scrapy
from ebay.items import EbayItem

class EbayJobSpider(scrapy.Spider):
    name = 'ebay_job'
    allowed_domains = ['ebay.com']
    start_urls =
["https://www.ebay.com/sch/i.html?_from=R40&_nkw=heart+rate+monitor&_sacat=0&
LH_ItemCondition=3&rt=nc&LH_BIN=1&_ipg=200"]

    def parse(self, response):
        titles = response.css("div > div.s-item__info.clearfix > a >
h3::text").extract()
        prices = response.css("div > div.s-item__info.clearfix > div.s-
item__details.clearfix > div:nth-child(1) > span").extract()

        for item in zip(titles, prices):
            new_item = EbayItem()

            new_item['titles'] = item[0]
            new_item['prices'] = item[1]

            yield new_item
```

```
# -*- coding: utf-8 -*-
import scrapy
from ebay.items import EbayItem

class EbayJobSpider(scrapy.Spider):
    name = 'ebay_job'
    allowed_domains = ['ebay.com']
    start_urls =
['https://www.ebay.com/sch/i.html?_from=R40&_nkw=heart+rate+monitor&_sacat=0&
LH_ItemCondition=3&rt=nc&LH_BIN=1&_ipg=200']

    def parse(self, response):

    titles = response.css('div > div.s-item__info.clearfix > a >
h3::text').extract()
    prices = response.css('div > div.s-item__info.clearfix > div.s-
item__details.clearfix > div:nth-child(1) > span').extract()

    for item in zip(titles, prices):
         new_item = EbayItem()

         new_item['titles'] = item[0]
         new_item['prices'] = item[1]

         yield new_item
```

## Step 6 – Crawling

After we set the scrapy framework structure it is time to run the Crawling function. To do this we will execute the following command:

```
$ scrapy crawl ebay_job
```

```
[...]
2019-01-20 20:12:32 [scrapy.core.scraper] DEBUG: Scraped from <200
https://www.ebay.com/sch/i.html?_from=R40&_nkw=heart+rate+monitor&_sacat=0&LH
_ItemCondition=3&rt=nc&LH_BIN=1&_ipg=200>
{'prices': '<span class="s-item__price">$13.39<span class="DEFAULT"> to '
           '</span>$13.79</span>',
 'titles': 'Original Xiaomi Mi Band 2 Smart Wristband Bracelet Heart Rate '
           'Monitor Smartwatch'}
2019-01-20 20:12:32 [scrapy.core.engine] INFO: Closing spider (finished)
2019-01-20 20:12:32 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
{'downloader/request_bytes': 759,
 'downloader/request_count': 2,
 'downloader/request_method_count/GET': 2,
 'downloader/response_bytes': 103540,
 'downloader/response_count': 2,
 'downloader/response_status_count/200': 2,
 'finish_reason': 'finished',
 'finish_time': datetime.datetime(2019, 1, 21, 1, 12, 32, 735435),
 'item_scraped_count': 206,
 'log_count/DEBUG': 209,
 'log_count/INFO': 7,
 'memusage/max': 51384320,
 'memusage/startup': 51376128,
 'response_received_count': 2,
 'scheduler/dequeued': 1,
 'scheduler/dequeued/memory': 1,
 'scheduler/enqueued': 1,
 'scheduler/enqueued/memory': 1,
 'start_time': datetime.datetime(2019, 1, 21, 1, 12, 29, 486444)}
```

**Step 7 – Exporting the extracted data for Data Analytics**

It is not enough the extraction of data, it is necessary to export it in a tabular dataset for further analyses. In order to do it is necessary to add arguments in the crawled command used:

```
$ scrapy crawl ebay_job -o ebay_extraction.csv -t csv
```

# Data Analytics

# Thank you