Table of Contents

[Anita scraper design 2](#_Toc44260391)

[Overview design 2](#_Toc44260392)

[Phase 1: Import 3](#_Toc44260393)

[Phase 2: Scrape 3](#_Toc44260394)

[Phase 3: Merge 3](#_Toc44260395)

[Phase 4: Export 4](#_Toc44260396)

[Scraped features 5](#_Toc44260397)

[Product 5](#_Toc44260398)

[Vendor 7](#_Toc44260399)

[Added features 9](#_Toc44260400)

[Product 9](#_Toc44260401)

[Vendor 10](#_Toc44260402)

[Feature overview per Market 11](#_Toc44260403)

[Product features 11](#_Toc44260404)

[Product feedback features 11](#_Toc44260405)

[Vendor features 12](#_Toc44260406)

[Vendor feedback features 12](#_Toc44260407)

[Appendix A 13](#_Toc44260408)

[Appendix A1 – Phase 1: Import 13](#_Toc44260409)

[Appendix A2 – Phase 2: Scraper 14](#_Toc44260410)

[Appendix A3 – Phase 3: Merge 15](#_Toc44260411)

[Appendix A4 – Phase 4: Export 15](#_Toc44260412)

[Appendix A – Keywords 19](#_Toc44260413)

# Anita scraper design

## Overview design

The scraping tool consists of four consecutive steps. These steps will globally be discussed to give an overview of the design of the tool. In the next chapter will discuss each part in a little more depth. The overview is given in figure 1.

The goal of the complete tool is to extract features from darknet market and to structurally store this information to be able to find changes and trends in the pages about vendors or products on these markets. This can be done by manually looking at the information for specific vendors or products or using the accompanying visualization tool.

A screenshot of a social media post

Description automatically generated

*Figure 1: General overview of tool.*

The input of the tool consists of a dump of a market to be scraped in a ZIP format or as a simple folder. In the filename the date of the dump is provided in order to be able to distinguish the different dumps for a market.

1. **Import**

The first phase imports and filters the useful files out of the provided dump of the market. The tool takes the html files and checks whether the html file is a page about a vendor or product. If so, the files are moved to a given folder. Redundant pages are not copied and removed afterwards. The original html files will be stored and could be inspected manually.

1. **Scrape**

The second phase is focused on retrieving the features out of the html files. While every market is different, every market has a separate file that includes the information what to scrape from the page. In this phase the file simply scrapes this information into JSON format, and the information is linked to the file name of the page.

1. **Merge**

The third phase focusses on merging the scraped information. While markets usually contain out of different pages that might contain the same information, there will be duplicate values in the data. For example, the profile page of a vendor might have information about the score of the vendor, while the page about feedback also contains this information. The merge step merges the information per page into information per vendor or product.

1. **Export**

The last phase exports the information into json files. Every dump per market has their own json file. The export phase checks possible duplicates in the dumps that might already exist and exports new or complements the existing json files. The exports are the results of the tool.

## Phase 1: Import

The import phase starts with providing a path to the input files and a folder where the html files are stored. The import path can be a ZIP file containing html files or a folder containing html files or folders containing the html files. Necessary is that one of the parent folders of the html files start with the dump date in the format: yyyy\_mm\_dd. Per file the name of the market, the type of the page (product / vendor) will be scraped. In the target folder a folder for the market and within this market folder, a folder with the dump date will be created. The files will be moved to the market and dump date folder they belong to. Lastly the remaining files will be removed, and this phase will return a list of paths to the moved files. An overview of the process is given in figure 2. A longer description might be introduced in the appendix.

A screenshot of a social media post

Description automatically generated

*Figure 2: Overview phase 1: Import*

## Phase 2: Scrape

The third phase starts with either the result from phase 1, the list with paths to html files or a path to where all the files from phase 1 are stored. If only the folder is provided, a list of html files will be created. The html files will be scraped one-by-one, first the market will be determined as in phase 1. With the name of the market, the appropriate scraper module file for that market will be used to determine the type of the page, product or vendor. These different types of pages consist of different features as will be mentioned in the next chapter. These features will be scraped and together with the main information about the page (dump date, market name and type of page) this information will be added to a list. Lastly all these data will be converted to json and returned.

A screenshot of a social media post

Description automatically generated

*Figure 3: Overview phase 2: Scrape*

## Phase 3: Merge

The third phase starts with the result of the second phase, a list of JSON per page. This phase merges all the page information into information per vendor or product. While doing so, it removes all duplicates. The idea is that it creates a dictionary where the key is an ID for the product or vendor. For every file (or page) in the JSON file the tool will check whether it was already seen, if not the page will completely be added to the new dictionary. If the ID, and thus the vendor or product has already been seen the tool will check whether the new page contains information that is not already in the values for that ID in the merged dictionary. New information can be added to the value in for that ID in the merged dictionary. Similar, reviews that are not yet in in the dictionary for a product or vendor are added. When all files are checked, the merged dictionary will be returned.

A picture containing clock

Description automatically generated

*Figure 4: Overview phase 3: Merge*

## Phase 4: Export

The last phase is to export the scraped and merged files into JSON files that can be used for visualization and trend analysis afterwards. This phase is very important and extensive, because products and vendors need to be linked to each other to be able to create a trend analysis. For example, you don’t want to lose track of a product, because the vendor adds the words ‘sale’ to the product name. Figure 5 provides a simple overview of the process. In the appendix you can find a more elaborate flowchart.

The idea behind the export phase is that a JSON file is created for the product and vendor pages for every dump date for a specific market. In this way we limit the file size per JSON file and are the files nicely structured. See figure 5 for an illustration. Next to the JSON files, two files per market are created to keep track of the names of the vendors and the product ID’s used in the data. Products will get a product ID that will be used to refer to products, the name of the product is not used, because it can be easily changed. To match product IDs the products of a specific vendor are marched based on the reviews written. Do the products for two different dump dates have a different name, but have the same vendor and same reviews, the products are considered the same product and will receive the same product id.

In this phase, the merged dictionary from phase 3 is used. Every different vendor or product will be sequentially looked at. If a JSON file for the current dump date already exist, it is checked whether the (vendor) page is already in the dump. If not, the information is added to the JSON file. For products the process is more complicated, but the main idea is that if the product is not yet in a current dump JSON file the product ID needs to be found or created and the product will be added to the JSON file for the dump date. If no JSON file exists yet a new JSON file will be created.

A screenshot of a computer screen

Description automatically generated

*Figure 5: Overview of JSON file structure*

A screenshot of a cell phone

Description automatically generated

*Figure 6: Overview phase 4: Export*

# Scraped features

## Product

**name**

Name of the product in string format.

**vendor**

Name of the vendor in string format.

**ships\_from**

Where the product is shipped to. This name can be any country or continent. If the location is an abbreviation it is automatically converted to full names.

The feature is given in strings, except when there are multiple countries to ship from, then this feature becomes a list of strings.

Note that if the product is digital, this value will be missing.

**ships\_to**

Where the product is shipped to. This name can be any country or continent. If the location is an abbreviation it is automatically converted to full names.

This feature is given as a list of strings.

Note that if the product is digital, this value will be missing.

Example: [Russian Federation, Namibia]

**price**

This feature contains the price of the product. There are broadly two types of price: (1) A fixed price for the product and (2) Different prices for different amounts of weights.

*Fixed price*

When the product has a fixed price, the price is given as a string. It is important that the currency can be found in this string. If it is missing, add manually to the individual market scraper.

Example: $2,100

*Variable price*

When there are multiple prices, the prices are given as a dictionary. The key is the label of the amount or weight, the value is the price. It is important that the currency can be found in this string. If it is missing, add manually to the individual market scraper.

Examples:

{'0.53$ per 1 Pill, for at least 1000 products ': '0.53$',

'0.66$ per 1 Pill, for at least 500 products ': '0.66$',

'0.79$ per 1 Pill, for at least 250 products ': '0.79$',

'0.92$ per 1 Pill, for at least 100 products ': '0.92$',

'1.05$ per 1 Pill, for at least 50 products ': '1.05$',

'1.18$ per 1 Pill, for at least 25 products ': '1.18$',

'1.31$ per 1 Pill, for at least 10 products ': '1.31$'}

**info**

This part contains the bulk of the information on the page. Most markets do have a field where the vendors can provide information. This information is included in this variable; as a string.

**macro\_category**

There are two category variables. Micro- and Macro. The macro category contains the broader category as much as possible. Every market describes their categories differently, therefore these values will differ a lot and cannot be considered categorical values without further processing.

When a market only consists of for example Drugs or Weapons, this is manually added to the individual market scraper.

**micro\_category**

The less broad categories of the product. Every market describes their categories differently, therefore these values will differ a lot and cannot be considered categorical values without further processing.

**feedback**

Feedback are the reviews of users on the profile or the product. The different reviews will be stored in a list of dictionaries. This consists of different **sub features:**

***score***

The score given by the reviewer. This can be a numerical score or a string.

The numerical scores are given as a tuple: (score, scale). Thus for 4 out of 5, the tuple will be (4,5). For 92% the tuple will be (92/100).

For a thumbs-up or a ‘positive’ score the string will be: ‘positive’. The same holds for ‘negative’ and ‘neutral’.

***message***

The message is a string with textual comments by the reviewer.

***date***

This is the date when the review is given. If the review is an exact date, the review is given as a datetime object in the individual market scraper. If the date is given as ‘2 days ago’, the value is given as a string.

***user***

The username of the reviewer as a string.

Examples:

[{'score': [5, 5],

'message': '\nI ALWAYS DO MY RESEARCH AND FE ALL VENDORS WHO HAVE THE STREET CREDIT AND TRUST CREDIT/REVIEWSif\n they end up sucking then I change the review. If theyre real good it \nleaves a good impression to a good working relationship. Ive followed \nand met many-a-vendor thru many market up and downsUPDATE - Got product super FAST and every cart (i got multiple brands thru this vendor) is 100% legit and im def coming back. ',

'date': 1578956400.0,

'user': None,

'date\_deviation': 'exact date'},

{'score': [5, 5],

'message': '\nWill be back for more! Thanks for always coming through VH! ',

'date': 1583103600.0,

'user': None,

'date\_deviation': 'exact date'}]

## Vendor

**name**

Name of the vendor in string format.

**score**

The average score for this vendor. The score per website differs a lot. There are two formats the score is given: numerical scores and sums of positive and negative scores.

The numerical scores are given as a tuple: (score, scale). Thus for 4 out of 5, the tuple will be (4,5). For 92% the tuple will be (92/100).

The sums of positive and negatives are given in a list: [positive, negative, neutral]. Thus for 96 negatives, no neutrals and 20 positives the list will be: [20, 96, 0].

Example: [4.96, 5]

**registration**

The date when the vendor has created its account. If the date is an exact date, the date is given as a datetime object in the individual market scraper. If the date is given as ‘2 days ago’, the value is given as a string.

The date will be converted to a datetime object. Information about time will be lost and only the date will be kept. Then the date will be converted to a unix time.

Examples: 1382133600

**last\_login**

The date when the vendor has logged in for the last time. If the date is an exact date, the date is given as a datetime object in the individual market scraper. If the date is given as ‘2 days ago’, the value is given as a string.

Examples: 1566252000

**sales**

The number of sales a vendor has made. This value is given as a numerical.

**info**

This part contains the bulk of the information on the page. Most markets do have a field where the vendors can provide information. This information is included in this variable; as a string.

**pgp**

The pgp of the vendor is given as a string.

**feedback**

Feedback are the reviews of users on the profile or the product. The different reviews will be stored in a list of dictionaries. This consists of different **sub features:**

***score***

The score given by the reviewer. This can be a numerical score or a string.

The numerical scores are given as a tuple: (sc qore, scale). Thus for 4 out of 5, the tuple will be (4,5). For 92% the tuple will be (92/100).

For a thumbs-up or a ‘positive’ score the string will be: ‘positive’. The same holds for ‘negative’ and ‘neutral’.

***message***

The message is a string with textual comments by the reviewer.

***date***

This is the date when the review is given. If the review is an exact date, the review is given as a datetime object in the individual market scraper. If the date is given as ‘2 days ago’, the value is given as a string.

***product***

If the review contains a reference to a product, as some markets provide. The product name will be extracted as a string.

***user***

The username of the reviewer as a string.

***deals***

The number of deals a user has made as a numerical value.

Examples:

[{'score': [5, 5],

'message': 'Fast Shipping, very good',

'date': 1562968800.0,

'product': None,

'user': None,

'deals': None,

'date\_deviation': 'week'}]

# Added features

## Product

**price\_eur**

This feature contains the price of the product in euro’s as a numerical value.

The price of the different markets is not always given in euro. Most website use dollars and/or bitcoins. If the price is given in euro’s, the price is scraped as euro’s and converted to a numerical in this feature.

If the price is in dollars it is converted to euros, with the conversion rate picked from the date the page was extracted in an API.

If the price is in bitcoin it is converted to dollars, with the conversion rate picked from the date the page was extracted in an API. Following this the dollar rate is converted to euros.

Example:

{'0.53$ per 1 Pill, for at least 1000 products ': 0.49,

'0.66$ per 1 Pill, for at least 500 products ': 0.61,

'0.79$ per 1 Pill, for at least 250 products ': 0.72,

'0.92$ per 1 Pill, for at least 100 products ': 0.84,

'1.05$ per 1 Pill, for at least 50 products ': 0.96,

'1.18$ per 1 Pill, for at least 25 products ': 1.08,

'1.31$ per 1 Pill, for at least 10 products ': 1.2}

**feedback: date\_deviation**

Dates that are not given as an exact date on the page will get a date assigned to it. For example, ‘2 months ago’ will get the date from 2 months ago. However, ‘2 months ago’ can also be 2 months and a few days ago. There is thus a deviation possible, this is what this feature explains. It can be ‘exact date’, ‘day’, ‘week’, ‘month’ or even ‘year’.

## Vendor

**score\_normalized**

Scores differ on the different markets; therefore, each market uses different scores. This feature normalizes all values, so that scores can be compared. The scale will be 0-1. Thus a 99% score will be .99 and a 4/5 will be 0.8.

**registration\_date\_deviation**

**last\_login\_deviation**

**Feedback: date\_deviation**

Dates that are not given as an exact date on the page will get a date assigned to it. For example, ‘2 months ago’ will get the date from 2 months ago. However, ‘2 months ago’ can also be 2 months and a few days ago. There is thus a deviation possible, this is what this feature explains. It can be ‘exact date’, ‘day’, ‘week’, ‘month’ or even ‘year’.

# Feature overview per Market

## Product features

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Market | name | vendor | ships\_from | ships\_to | price | info | macro\_category | micro\_category | feedback |
| Agartha | X | X | X | X | X | X | X | X | X |
| Apollon | X | X | X | X | X | X |  |  | X |
| Berlusconi | X | X | x | X | X | X | X | X | X |
| Cannahome | X | X | X |  | X | X | XD |  | X |
| Cannazon | X | X | X | X | X | X | XD |  |  |
| Darkmarket | X | X |  |  | X | X | X | X | X |
| Directdrugs | X |  |  |  | X | X | XD | X | X |
| Drugscenter | X |  | XF | XF | X | X | XD | X | X |
| Drugsmedicine | X | X | X | X | X | X | XD | X | X |
| EmpireMarket | No data | | | | | | | | |
| Palmetto | X |  |  |  | X | X | XW | X | X |
| Silkroad3 | X | X | X |  | X | X |  | X | X |
| Tochka | X | X | X | X | X | X |  |  | X |

D Fixed Drugs macro category, this marketplace only sells drugs

F This is a fixed value, because all values are the same

W Fixed Weapons macro category, this marketplace only sells weapons

## Product feedback features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Market | score | message | date | user |
| Agartha | X | X | X | X |
| Apollon | X | X | X | X |
| Berlusconi | X | X | X | X |
| Cannahome | X | X | X |  |
| Cannazon | No product feedback | | | |
| Darkmarket | X | X |  |  |
| Directdrugs | X | X | X | X |
| Drugscenter | X | X | X | X |
| Drugsmedicine | X | X | X |  |
| EmpireMarket | No product feedback | | | |
| Palmetto | X | X | X | X |
| Silkroad3 | X | X | X | X |
| Tochka | X | X | X | X |

## Vendor features

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Market | name | score | registration | last\_login | sales | info | pgp | feedback |
| Agartha | X | X | X | X | XB | X | X | X |
| Apollon | X | X | X | X | X | X | X | X |
| Berlusconi | X | X | X | X | X | X | X | X |
| Cannahome | X | X |  | X |  | X | X | X |
| Cannazon | X | X | X |  | X | X | X | X |
| Darkmarket | X | X | X | X |  | X | X | A |
| Directdrugs | No vendors on this market | | | | | | | |
| Drugscenter | No vendors on this market | | | | | | | |
| Drugsmedicine | X | X | X | X | X | X | X | X |
| EmpireMarket | X | X | X | X | X | X | X |  |
| Palmetto | No vendors on this market | | | | | | | |
| Silkroad3 | X | X |  | X |  | X |  | X |
| Tochka | X | X | X | X |  | X |  | X |

A There exists vendor feedback, but it is difficult to scrape and therefore left out

B The Agartha market uses categories as sales. Therefore more extreme increases and longer steady periods can be expected.

## Vendor feedback features

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Market | score | message | date | product | user | deals |
| Agartha | X | X | X | X | X | X |
| Apollon | X | X | X | X | X |  |
| Berlusconi | X | X | X |  | X | X |
| Cannahome | X | X | X | X |  |  |
| Cannazon | X | X | X | X |  |  |
| Darkmarket | Hard to scrape | | | | | |
| Directdrugs | No vendors on this market | | | | | |
| Drugscenter | No vendors on this market | | | | | |
| Drugsmedicine | X | X | X |  |  |  |
| EmpireMarket | No vendor feedback | | | | | |
| Palmetto | No vendors on this market | | | | | |
| Silkroad3 | X | X | X | X | X |  |
| Tochka | X | X | X |  | X |  |

# Appendix A

In this document simpler flowcharts are provided to give an overview of the phases in scraper tool. In this appendix more extended flowcharts are provided.

## Appendix A1 – Phase 1: Import

A close up of a logo

Description automatically generated

*Figure 7: Flowchart phase 1: Import*

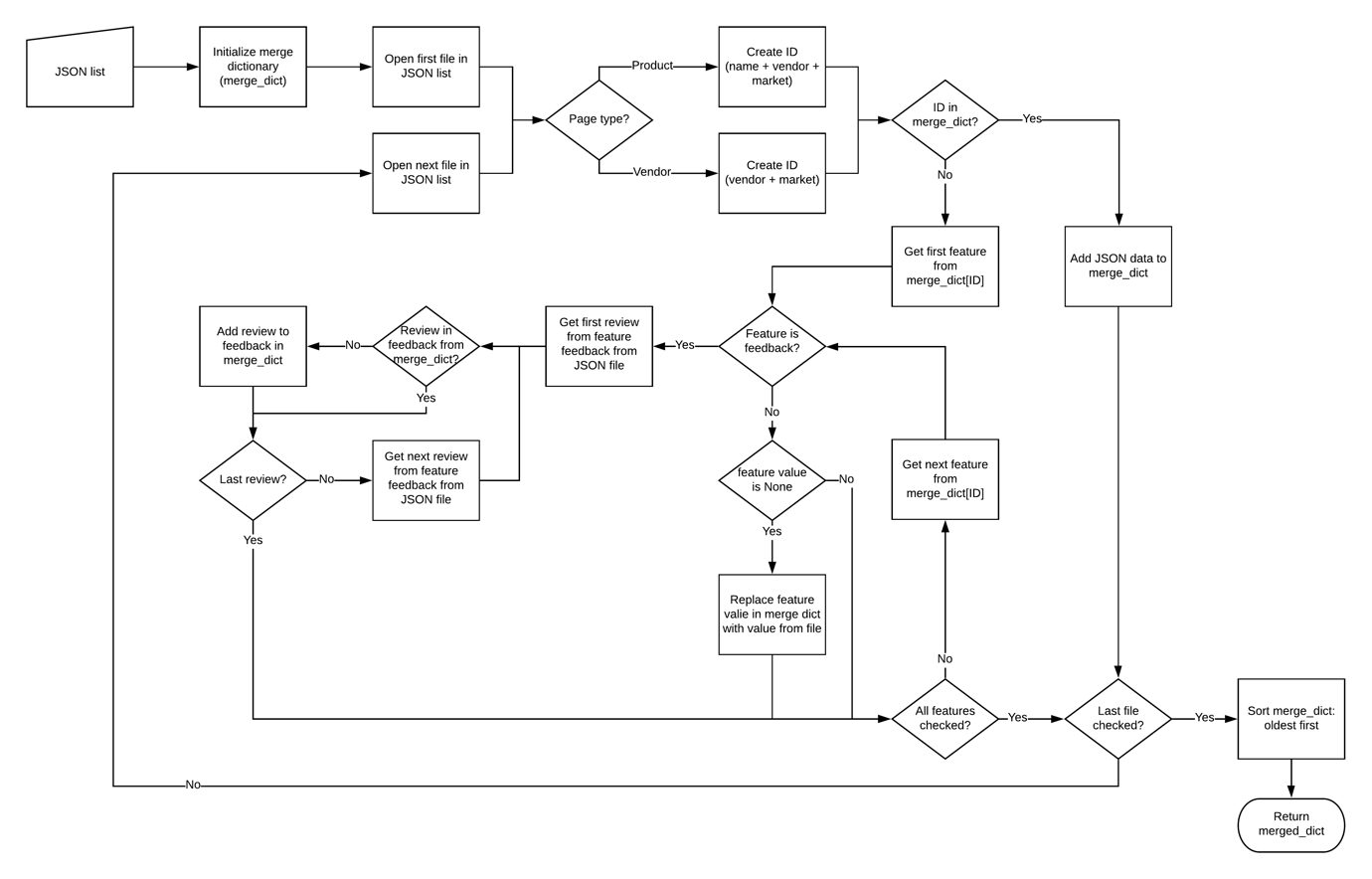
## Appendix A2 – Phase 2: Scraper

A close up of text on a white background

Description automatically generated

*Figure 8: Flowchart phase 2: Scraper*

## Appendix A3 – Phase 3: Merge



*Figure 9: Flowchart phase 3: Merge*

## Appendix A4 – Phase 4: Export

The flowchart for this phase is large. Therefore, the flowchart is cut into four parts: (1) The first is the main part which provides the overview of the process. Parts a, b and c are provided in a separate flowchart below. The circles in the main flowchart explain where this part belongs. Duplicate parts, indicated with a dotted line, are used to show overlap between the different parts and the main flowchart.

A close up of a logo

Description automatically generated

*Figure 9: Flowchart phase 4: Export - main*

A close up of a piece of paper

Description automatically generated

*Figure 10: Flowchart phase 4: Export – part A*

A close up of a logo

Description automatically generated

*Figure 11: Flowchart phase 4: Export – part B*

A close up of a logo

Description automatically generated

*Figure 12: Flowchart phase 4: Export – part C*

# Appendix A – Keywords

**Dump:** A dump is a collection of pages that is exported from the market on a certain date.

**Individual market scraper:** The scraper that is specific to the market

**Market**: Market is an individual website where products are sold

**Product:** The sold goods on a market page.

**Product\_id:** The ID for a product. While vendors might change the name of the product, the product ID is used to refer to products.

**Soup:** Soup is the name for the structured html file that is scraped. A soup can be searched through for specific parts and information.

**Vendor:** The seller of a product on a market page.