DataGorri 1.1

Manual

For Users

Daniel Krieger, Michael Legenc, Julian Hackinger

16th August 2017

Contents

1 DataGorri

DataGorri is an application used to extract data from tables located on websites. To achieve this, two steps are necessary:

- Create a page model to define the content in which one is interested in.
- Collect links of websites that should be scraped.

The first step is handled by the DataGorri's modeler. Here, the user inspects a website's structure to define the contents in which she is interested in. For this, the modeler displays all tables on this page. The user then simply selects the desired contents of one or more tables and creates a page model for this specific page structure. This model then can be used for all pages with similar structure. The second step is collecting one or more URLs that should be scraped with the page model. After both steps, the scraping result can now be generated and saved to a .CSV (Microsoft Excel compatible) file.

2 Installation

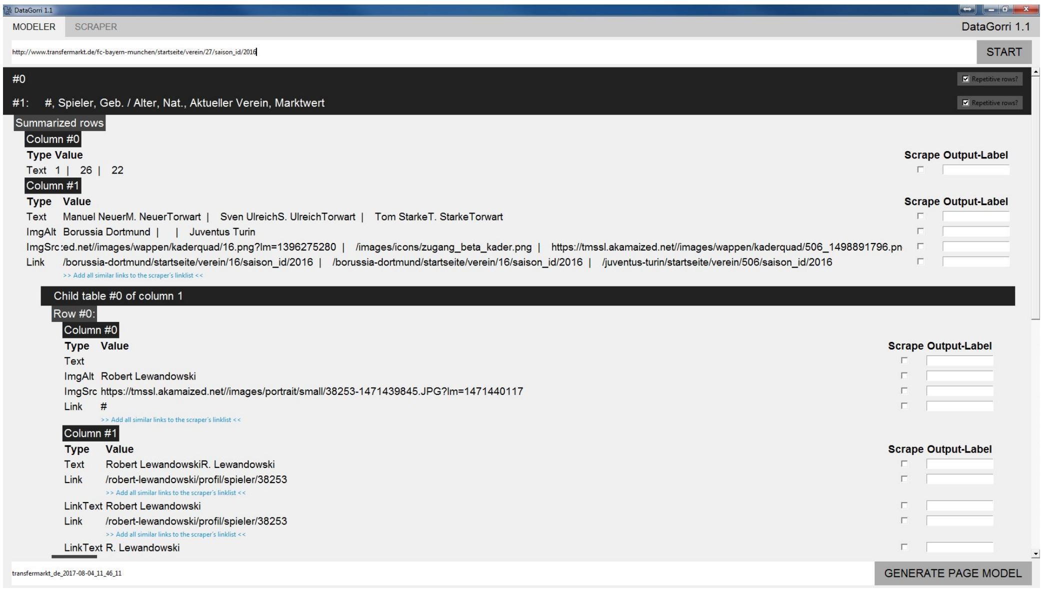
Windows: There is an installer provided for Windows (32 and 64bit). The installation is straightforward and does not need extensive explanation. The user is only required to select a destination directory for the application. Per default this will be the "Program Files (x86)" directory. After the installation, a shortcut to DataGorri can be found on the desktop. The installed files of DataGorri can be found in the respective directory. There is no need to install Python or third-party libraries.

MacOS, Ubuntu: For other operating systems like Mac OS or Ubuntu, DataGorri can be run by installing Python and running the source code by the console/terminal. How to install Python and necessary third-party libraries is described in the documentation for developers. After that, unzip the project files and simply run following commands in the console:

```
cd {PATH_TO_DATAGORRI_DIR}
python DataGorri.py
```

3 The Modeler

The modeler is used to inspect a website's structure and to define the contents in which the user is interested in. The result of this is a page model that can be used for multiple website that share a similar page structure.



http://www.transfermarkt.de/fc-bayern-munchen/startseite/verein/27/saison_id/2016

#0

#1: #, Spieler, Geb. / Alter, Nat., Aktueller Verein, Marktwert

Summarized rows

Column #0

Type	Value	Scrape	Output-Label
Text	1 | 26 | 22	□	

Column #1

Type	Value	Scrape	Output-Label
Text	Manuel NeuerM. NeuerTorwart | Sven UlreichS. UlreichTorwart | Tom StarkeT. StarkeTorwart	□	
ImgAlt	Borussia Dortmund | | Juventus Turin	□	
ImgSrc	:ed.net//images/wappen/kaderquad/16.png?lm=1396275280 | /images/icons/zugang_beta_kader.png | https://tmssl.akamaized.net//images/wappen/kaderquad/506_1498891796.pn	□	
Link	/borussia-dortmund/startseite/verein/16/saison_id/2016 | /borussia-dortmund/startseite/verein/16/saison_id/2016 | /juventus-turin/startseite/verein/506/saison_id/2016	□	

Child table #0 of column 1

Row #0:

Column #0

Type	Value	Scrape	Output-Label
Text		□	
ImgAlt	Robert Lewandowski	□	
ImgSrc	https://tmssl.akamaized.net//images/portrait/small/38253-1471439845.JPG?lm=1471440117	□	
Link	#	□	

Column #1

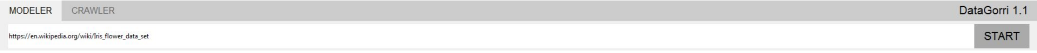
Type	Value	Scrape	Output-Label
Text	Robert LewandowskiR. Lewandowski	□	
Link	/robert-lewandowski/profil/spieler/38253	□	
LinkText	Robert Lewandowski	□	
Link	/robert-lewandowski/profil/spieler/38253	□	
LinkText	R. Lewandowski	□	

transfermarkt_de_2017-08-04_11_46_11

Figure 1: The modeler.

The following describes the individual parts of the scraper and their usage.

3.1 URL bar



https://en.wikipedia.org/wiki/Iris_flower_data_set

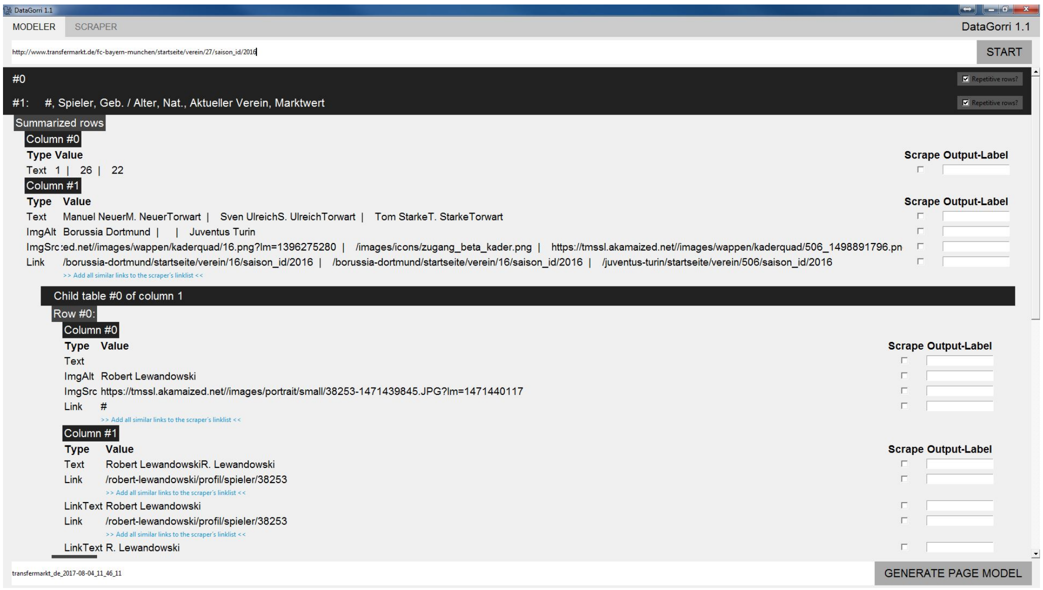
Figure 2: URL bar.

Insert here the URL of the website from which the page model should be created and press the "START" button to analyse the page and to display the tables of the respective website.

3.2 Table view

After starting the modeler, the tables are displayed that are not placed within other ones. First only very basic information is visible like the consecutive numbering and table headers if available. On the right hand side is also a check button next to the "Repetitive" label. By ticking it, the user can define whether the table is repetitive or not. Tables 1 and 2 describe both types of tables.

After defining the type of the table, a click will expand it and display information about what in this table is scrapeable and show further child



http://www.transfermarkt.de/fc-bayern-munchen/startseite/verein/27/saison_id/2016

#0

#1: #, Spieler, Geb. / Alter, Nat., Aktueller Verein, Marktwert

Summarized rows

Column #0

Type	Value	Scrape	Output-Label
Text	1 | 26 | 22	□	

Column #1

Type	Value	Scrape	Output-Label
Text	Manuel NeuerM. NeuerTorwart | Sven UlreichS. UlreichTorwart | Tom StarkeT. StarkeTorwart	□	
ImgAlt	Borussia Dortmund | | Juventus Turin	□	
ImgSrc	:ed.net//images/wappen/kaderquad/16.png?lm=1396275280 | /images/icons/zugang_beta_kader.png | https://tmssl.akamaized.net//images/wappen/kaderquad/506_1498891796.pn	□	
Link	/borussia-dortmund/startseite/verein/16/saison_id/2016 | /borussia-dortmund/startseite/verein/16/saison_id/2016 | /juventus-turin/startseite/verein/506/saison_id/2016 >> Add all similar links to the scraper's linklist <<	□	

Child table #0 of column 1

Row #0:

Column #0

Type	Value	Scrape	Output-Label
Text		□	
ImgAlt	Robert Lewandowski	□	
ImgSrc	https://tmssl.akamaized.net//images/portrait/small/38253-1471439845.JPG?lm=1471440117	□	
Link	# >> Add all similar links to the scraper's linklist <<	□	

Column #1

Type	Value	Scrape	Output-Label
Text	Robert LewandowskiR. Lewandowski	□	
Link	/robert-lewandowski/profil/spieler/38253 >> Add all similar links to the scraper's linklist <<	□	
LinkText	Robert Lewandowski	□	
Link	/robert-lewandowski/profil/spieler/38253 >> Add all similar links to the scraper's linklist <<	□	
LinkText	R. Lewandowski	□	

transfermarkt_de_2017-08-04_11_46_11

Figure 3: Table view.

Table 1: **Repetitive Table:** A table where every column has the same type of content and a row represents one instance of similar information according to its semantic.

Object	**Cost**	**Availability**
table	250	yes
chair	50	yes
bed	150	no

Table 2: **Non-repetitive Table:** A table where no column and row represents the same kind of information according to its semantic.

Object	table
Cost	250
Availability	yes

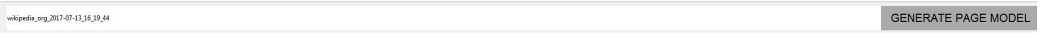
tables. Just mark here with the checkbox the desired contents to scrape and specify the “Output-Label”. This output-label will later be used to label the scraped result values.

To simplify the collecting of URLs, contents of type “Link” can be used to directly add them to the link list of the scraper (see also Section 4.2). Every content of this type has a blue link

“>> Add all similar links to the scraper’s linklist <<”.

All of the URLs in the respective column of the table will directly be added to the scraper’s link list by clicking on it. For repetitive tables multiple URLs are added by clicking at one representative.

3.3 Generate page model bar

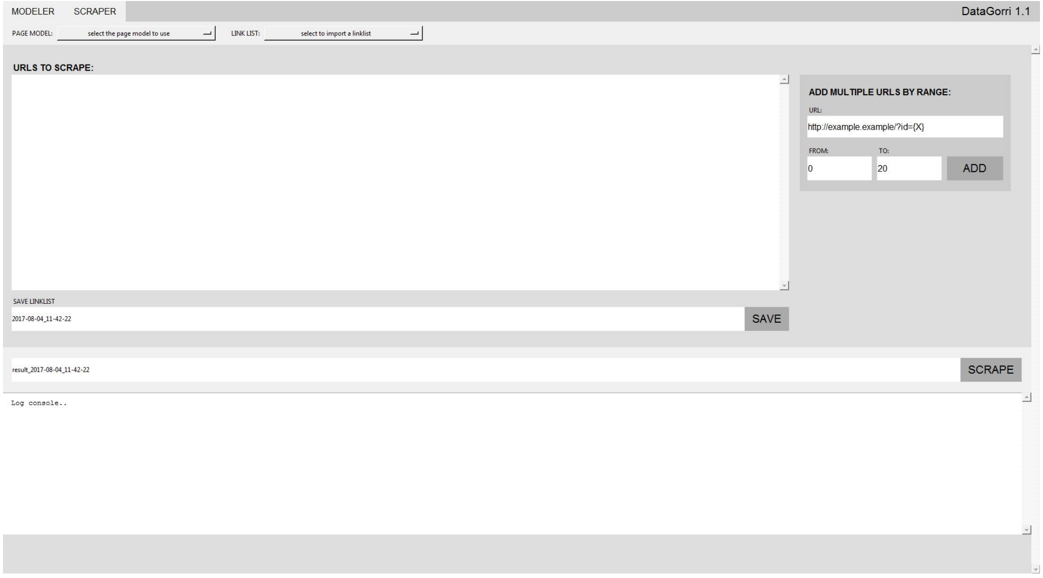


wikipedia_org_2017-07-13_16_19_44

Figure 4: Generate page model bar.

The input in the bar at the bottom of the modeler can be used to give the page model a user-defined name. The default name is the domain name of the URL in the URL bar concatenated with a timestamp. Pressing “GENERATE PAGE MODEL” creates the page model that is from now on available in the scraper. To remove this generated page model, just move to the directory `DataGorri_Output/page_models/` (located on the desktop) and remove the file manually.

4 The Scraper



PAGE MODEL: select the page model to use

LINK LIST: select to import a linklist

URL: http://example.example/?id={X}

FROM: 0

TO: 20

SAVE LINKLIST: 2017-08-04_11-42-22

result_2017-08-04_11-42-22

```
Log console..
```

Figure 5: The scraper.

The scraper needs a page model and a list of URLs to scrape. Figure 5 depicts this part of the application. The following describes the individual parts and their usage.

4.1 Page model selection

This dropdown menu is used to select the page model, which will then be used by the scraper.

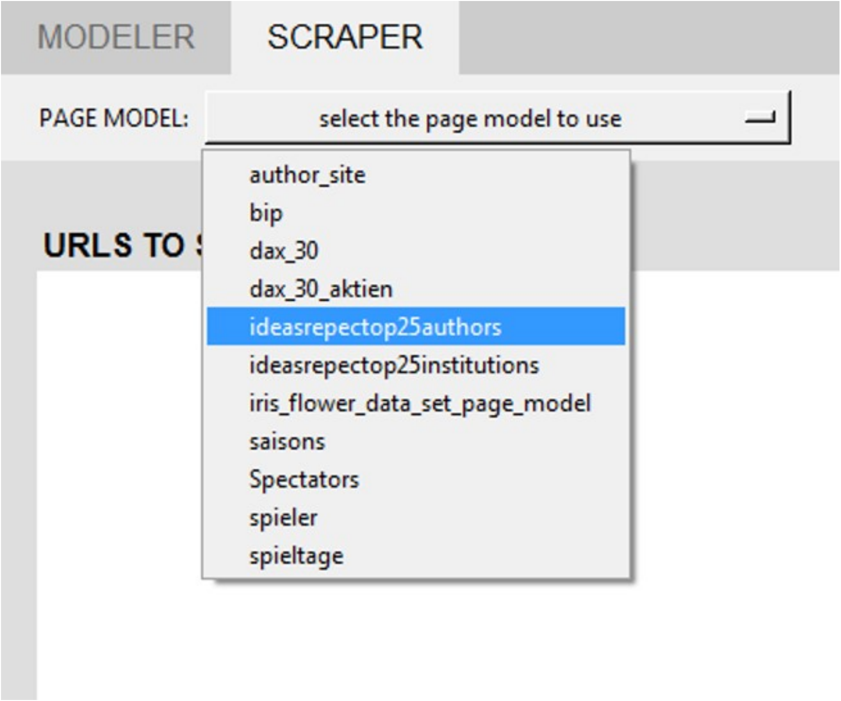
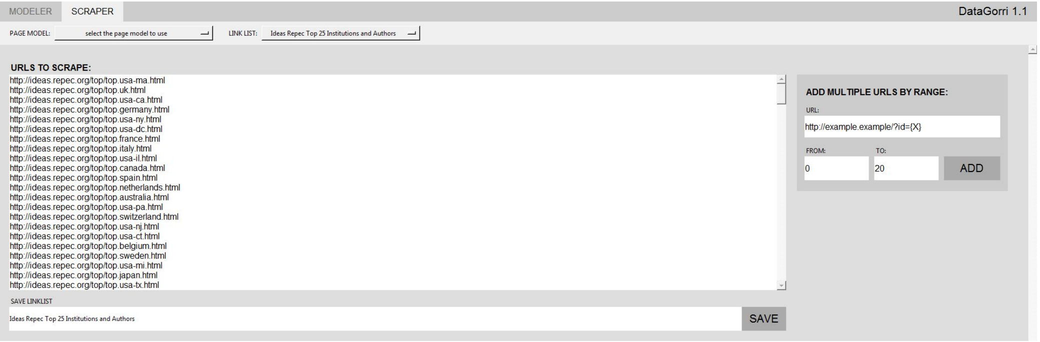


Figure 6: Page Model Selection.

4.2 Link list

This text field contains the list of URLs. Every URL gets an own line. The page model will be applied on these links and the scraping results come from these defined pages.

Link lists do not have to be generated over and over again, because there is a save function. Just enter a name for the link list and press the "SAVE" button. Those lists can then be imported again via the respective dropdown menu. To remove a saved link list, just go to `DataGorri_Output/link_lists/` (located on the desktop) and remove the file manually.



PAGE MODEL: select the page model to use

LINK LIST: Ideas Repec Top 25 Institutions and Authors

URLS TO SCRAPE:

```
http://ideas.repec.org/top/top.usa-ma.html
http://ideas.repec.org/top/top.uk.html
http://ideas.repec.org/top/top.usa-ca.html
http://ideas.repec.org/top/top.germany.html
http://ideas.repec.org/top/top.usa-ny.html
http://ideas.repec.org/top/top.usa-dc.html
http://ideas.repec.org/top/top.france.html
http://ideas.repec.org/top/top.italy.html
http://ideas.repec.org/top/top.usa-il.html
http://ideas.repec.org/top/top.canada.html
http://ideas.repec.org/top/top.spain.html
http://ideas.repec.org/top/top.netherlands.html
http://ideas.repec.org/top/top.australia.html
http://ideas.repec.org/top/top.usa-pa.html
http://ideas.repec.org/top/top.switzerland.html
http://ideas.repec.org/top/top.usa-nj.html
http://ideas.repec.org/top/top.usa-ct.html
http://ideas.repec.org/top/top.belgium.html
http://ideas.repec.org/top/top.sweden.html
http://ideas.repec.org/top/top.usa-mi.html
http://ideas.repec.org/top/top.japan.html
http://ideas.repec.org/top/top.usa-tx.html
```

SAVE LINKLIST: Ideas Repec Top 25 Institutions and Authors

ADD MULTIPLE URLS BY RANGE:

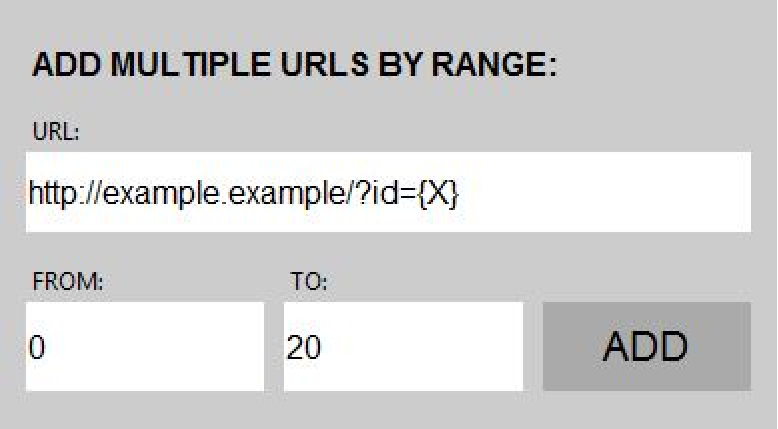
URL: http://example.example/?id={X}

FROM: 0 TO: 20

Figure 7: Page Model Selection.

4.3 Range adder

This feature can be used to create similar URLs, in which just one numerical index changes. This is often the case for very similar pages and thereby very useful for the page model approach.



ADD MULTIPLE URLS BY RANGE:

URL: http://example.example/?id={X}

FROM: 0

TO: 20

Figure 8: Range adder.

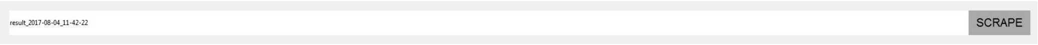
The placeholder for this feature is {X}, so Figure 8 as an example would create the following links:

```
http://example.example/?id=0
http://example.example/?id=1
http://example.example/?id=2
...
http://example.example/?id=20
```

Pressing “ADD” will then create the URLs and add them to the link list.

4.4 Scrape bar

This bar can be used to give the result a user-defined name. The result will be saved to a .CSV file, which is Microsoft Excel compatible. The default name is “resul” concatenated with a timestamp.

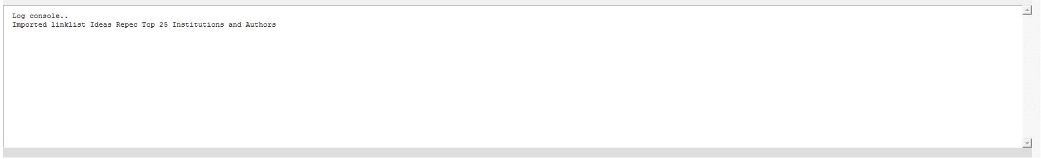


result_2017-08-04_11-42-22

Figure 9: Scrape bar.

4.5 Log console

This field is logging actions in chronological order. It will help the user to keep track of the scraping process and to see possible problems. At the end a report with warnings and failures is printed. Warnings in this report are common and do not mean that the scraping somehow failed. It is only useful to detect URLs to which the page model is not appropriate, because a lot of warnings will refer to this URL. Nonetheless, failures should be extensively inspected.



```
Log console..
Imported linklist Ideas Repec Top 25 Institutions and Authors
```

Figure 10: Log.

5 Page Model-, Link List- and Result Files

After starting the DataGorri for the first time, there will be a folder named `DataGorri_Output` on the desktop:

This folder contains the following three subfolders:

- `link_lists`: Contains the saved link lists.
- `models`: Contains the generated page models.
- `results`: Contains the scraping results in .CSV format.

If you cannot find the folder on your desktop, it might have been created on another path. In some systems several "Desktops" exist, particularly in systems with managed (corporate) PCs. In these cases, the folder might have been created on the one that is not shown on the homescreen. You can find it by browsing directly to `C:\Users\[Username]\Desktop\DataGorri_Output`. We recommend to create a shortcut to this folder and save it to an easily accessible place. The subfolders contained in `DataGorri_Output` are empty at first. The page models, link lists, as well as your results will be stored there. If you would like to start with some sample files first, you find some in the folder `DataGorri\samples` in your installation.