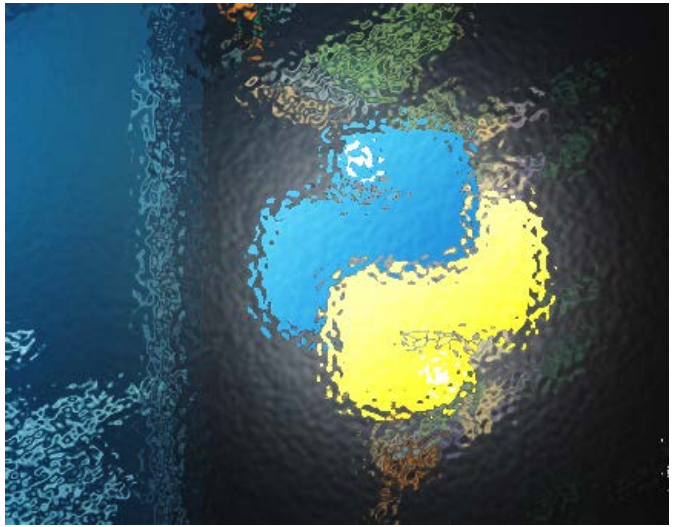
Web Scraping

Computer-Science Projects on BigData

Elias Larbi
Bart Gerritsen

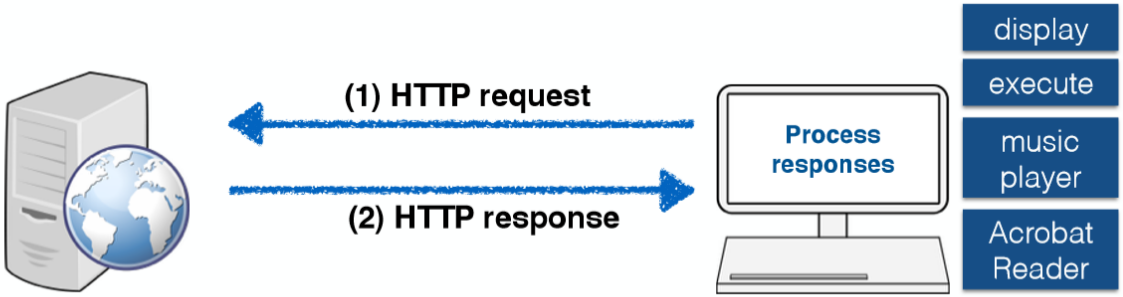
Dec 5, 2018



Bart Gerritsen
Dec 5, 2018

TU Delft

Client-server and the Web



Manages web resources

- Servers wait for data requests
- Answer thousands of clients simultaneously
- Host **web resources**

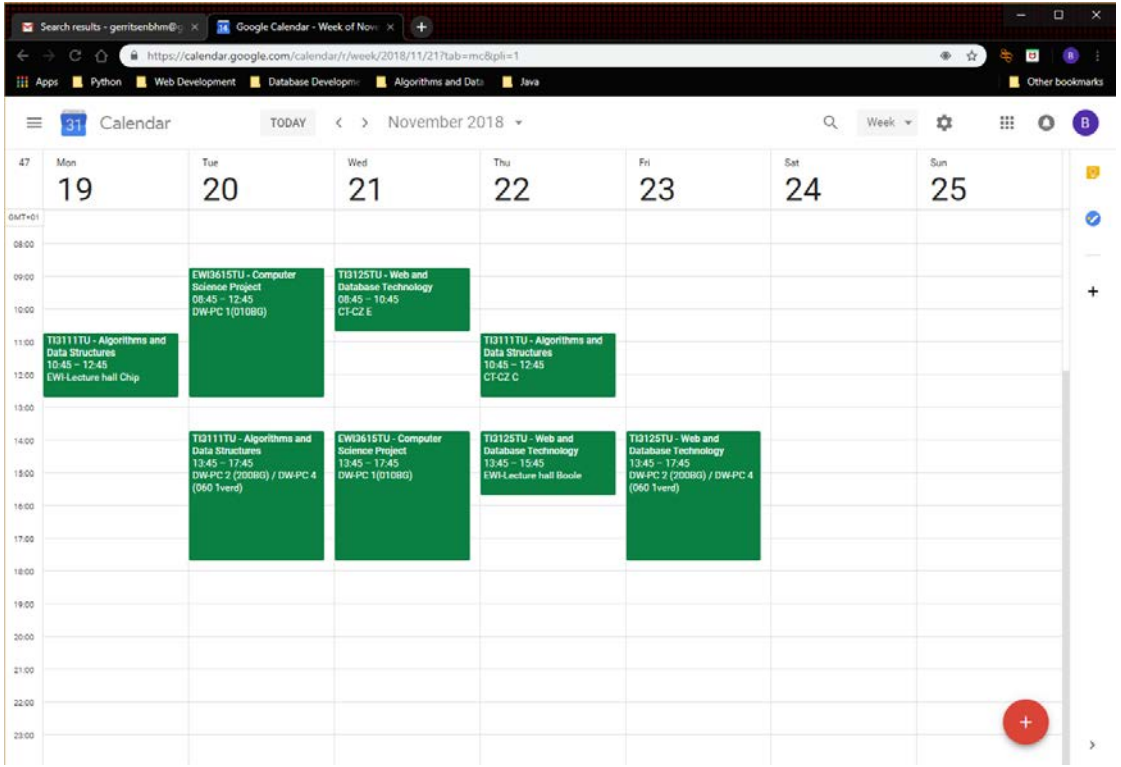
- Clients are most often Web browsers
- Telnet

Consumes / manipulates web resources

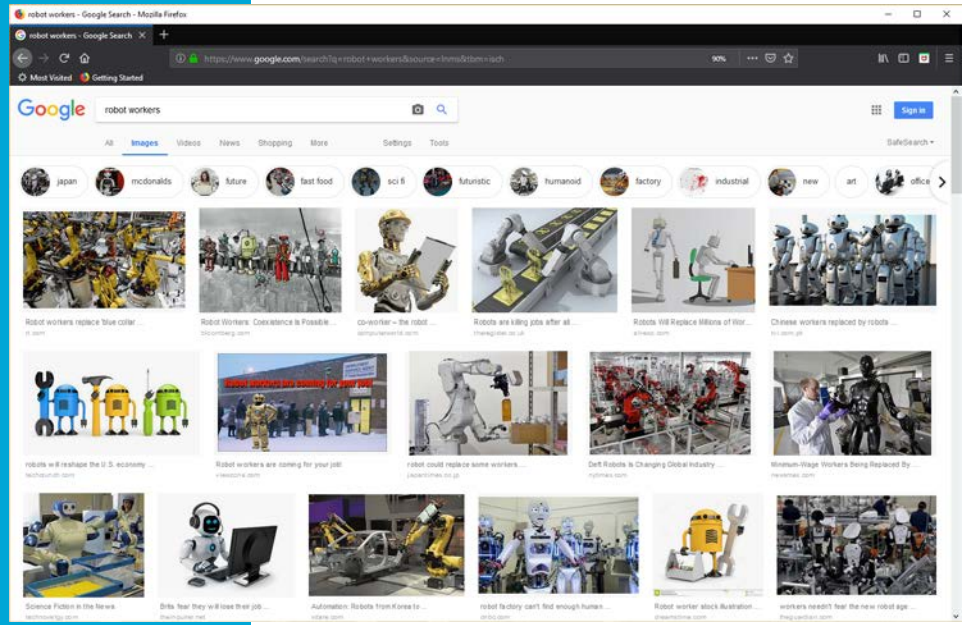
Web resource: any kind of content with an identity, including static files (e.g. text, images, video), software programs, Web cam gateway, etc.

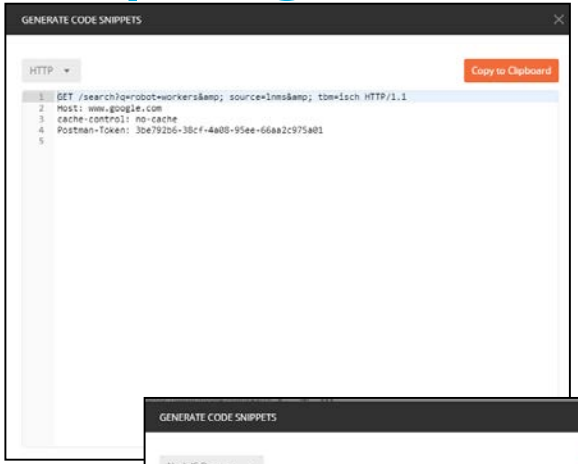
TU Delft

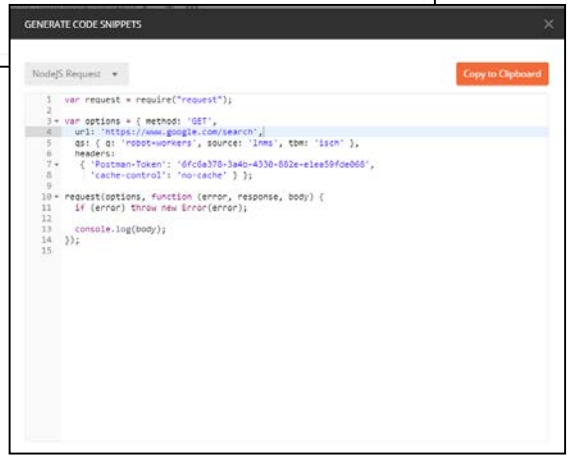
Demo query


Calendar
TODAY
November 2018
Week
Mon 19
Tue 20
Wed 21
Thu 22
Fri 23
Sat 24
Sun 25
TI3111TU - Algorithms and Data Structures 10:45 – 12:45 EWI-Lecture hall Chip
EWI3615TU - Computer Science Project 08:45 – 12:45 DW-PC 1(0108G)
TI3125TU - Web and Database Technology 08:45 – 10:45 CT-CZ E
TI3111TU - Algorithms and Data Structures 10:45 – 12:45 CT-CZ C
TI3111TU - Algorithms and Data Structures 13:45 – 17:45 DW-PC 2 (2008G) / DW-PC 4 (060 1verd)
EWI3615TU - Computer Science Project 13:45 – 17:45 DW-PC 1(0108G)
TI3125TU - Web and Database Technology 13:45 – 15:45 EWI-Lecture hall Boole
TI3125TU - Web and Database Technology 13:45 – 17:45 DW-PC 2 (2008G) / DW-PC 4 (060 1verd)

Requests and web programming

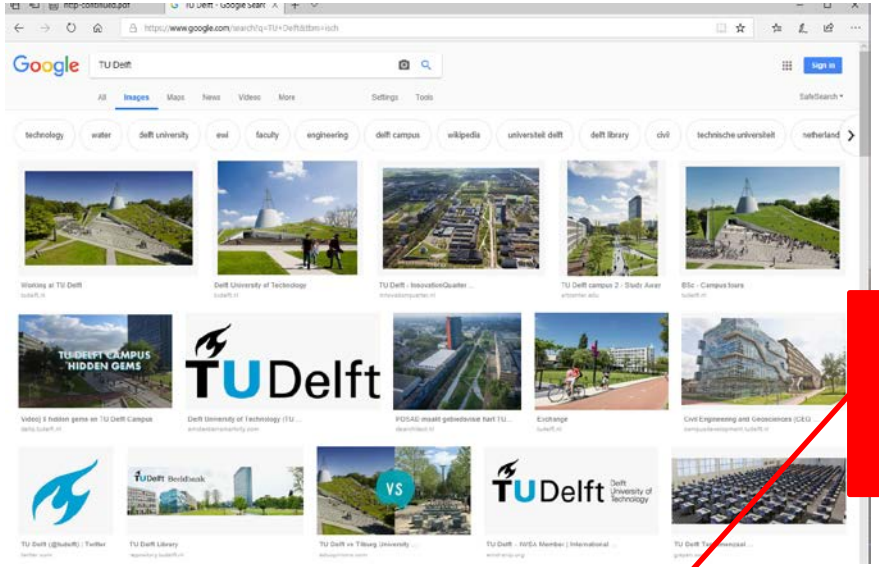






URL – Syntax query

<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>



name1=value1&name2=value2&…

https://www.google.com/search?q=TU+Delft&tbm=isch

TU Delft

Using Python and BeautifulSoup

Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

D:\bhmgerritsen\DUT-EWI\Lecturing-Education\2018-2019\CS-Projects

Editor - D:\bhmgerritsen\DUT-EWI\Lecturing-Education\2018-2019\CS-Projects\bs-demo.py

temp.py | bs-demo.py

```python
# -*- coding: utf-8 -*-
"""
Web Scraping

@author: bhmgerritsen
@date: Fall 2018
@rev: 1.0
"""

from bs4 import BeautifulSoup as bs
import requests as rq
import re

# Ok response from request ...
OK_CODE = 200

KEYWORDS = [
        r'bachelor',
        r'onderwijs',
        r'ewi',
        r'3mE',
        r'wheather model'
]

# collect re for special patterns in the below dictionary ...
RE_PATTERNS = {
        'form'    : r'inputfield',
        'dies'    : r'dies[- ]+natalis',
        'tel_nr'  : r'(([+]31|0)[ ]?\d\d[-]?\d\d\d\d\d\d\d)',
        'user_at' : r'(<)?(\w+@\w+(?:\.\w+)+)(?(1)>|$)'
}

def get_data_from_web(host, path='', q_string=''):
    """compose a full url and send request"""

    # do some checking first ...
    assert host is not None, 'specify valid host'

    # now compose the url ...
    url = r'http'+'://'+host+path

    # query string specified ?
    if q_string:
        url += r'?' + q_string
```

Help

Source Console Object

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
      <button class="button-reset btn--icon i-search js-modal" data-target=".nav-search">
       <span class="text-hide">
        Search
       </span>
      </button>
     </div>
    </div>
   </div>
  </nav>
  <script src="/typo3conf/ext/site_tud/Resources/Public/JavaScript/dist.javascripts.9a020b01ec.min.js" type="text/
javascript">
  </script>
 </body>
</html>
... searching for keyword: bachelor ...
          <input name="q" type="hidden" value="bachelor"/>
         <input autocomplete="off" class="searchForm-input autocomplete-inputField" name="q" type="search"
value="bachelor"/>
        <a class="news-summary" href="/onderwijs/opleidingen/bachelors/" target="_top" title="Bachelors">
            /onderwijs/opleidingen/bachelors/
        <a class="news-summary" href="/evenementen/2017/scholierenlab/open-dag-bachelor-opleidingen-19okt18/"
target="_top" title="OPEN DAG bachelor opleidingen">
           OPEN DAG bachelor opleidingen
            /evenementen/2017/scholierenlab/open-dag-bachelor-opleidingen-19okt18/
        <a class="news-summary" href="/evenementen/2017/scholierenlab/open-dag-bachelor-opleidingen-22okt18/"
target="_top" title="OPEN DAG bachelor opleidingen">
           OPEN DAG bachelor opleidingen
```

IPython console | History log

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 1 Column: 1 Memory: 41 %

TU Delft

HTTP methods

GET	Get a document from the Web server.
HEAD	Get the header of a document from the Web server.
POST	Send data from the client to the server for processing.
PUT	Save the body of the request on the server.
TRACE	Trace the message through proxy servers to the server.
OPTIONS	Determine what methods can operate on a server.
DELETE	Remove a document from a Web server.

Servers may implement more or fewer methods than shown.

TU Delft

Status codes

Also see:

https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

1xx: HTTP Informational Codes	
100	Continue
101	Switching Protocols
102	Processing WebDAV
103	Checkpoint draft POST PUT
122	Request-URI too long IE7

2xx: HTTP Successful Codes	
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information 1.1
204	No Content
205	Reset Content
206	Partial Content
207	Multi-Status WebDAV 4918
208	Already Reported WebDAV 5842
226	IM Used 3229 GET

3xx: HTTP Redirection Codes	
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other 1.1
304	Not Modified
305	Use Proxy 1.1
306	Switch Proxy unused
307	Temporary Redirect 1.1
308	Permanent Redirect 7538

307 and **308** are similar to **302** and **301**, but the new request method after redirect must be the same, as on initial request.

4xx: HTTP Client Error Code	
400	Bad Request
401	Unauthorized
402	Payment Required res
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
416	Requested Range Not Satisfiable
417	Expectation Failed
418	I'm a teapot 2324
422	Unprocessable Entity WebDAV 4918
423	Locked WebDAV 4918
424	Failed Dependency WebDAV 4918
425	Unordered Collection 3648
426	Upgrade Required 2817
428	Precondition Required draft
429	Too Many Requests draft
431	Request Header Fields Too Large draft
444	No Response nginx
449	Retry With MS
450	Blocked By Windows Parental Controls MS
451	Unavailable For Legal Reasons draft
499	Client Closed Request nginx

5xx: HTTP Server Error Codes	
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version Not Supported
506	Variant Also Negotiates 2295
507	Insufficient Storage WebDAV 4918
508	Loop Detected WebDAV 5842
509	Bandwidth Limit Exceeded nostd
510	Not Extended 2774
511	Network Authentication Required draft
598	Network read timeout error nostd
599	Network connect timeout error nostd

HTTP Code Comments	
WebDAV	WebDAV extension
1.1	HTTP/1.1
GET, POST, PUT, POST	For these methods only
IE	IE extension
MS	MS extension
nginx	nginx extension
2518, 2817, 2295, 2774, 3229, 4918, 5842	RFC number
draft	Proposed draft
nostd	Non standard extension
res	Reserved for future use
unused	No more in use, deprecated

Wikipedia was used to produce all HTTP codes content:
http://en.wikipedia.org/wiki/HTTP_status

TU Delft

HTTP headers

Content-Type	Entity type
Content-Length	Length/size of the message
Content-Language	Language of the entity sent (e.g. English)
Content-Encoding	Data transformations applied to the entity
Content-Location	Alternative location of the entity
Content-Range	For partial entities, range defines the pieces sent
Content-MD5	Checksum of the content
Last-Modified	Date on which this entity was created/modified
Expires	Date at which the entity will become stale
Allow	Lists the legal request methods for the entity

Important: Entity bodies only contain **raw** data, **header** information required to **interpret** the data.

See: http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/http_basics.html

User-related HTTP header fields

From	Request	User's email address	mostly Web crawler
User-Agent	Request	User's browser	device customization
Referer	Request	Page the user came from	user interests
Client-IP	Request (Extension)	Client's IP address	
Authorization	Request	Username & password	

TU Delft

Authentication

- Username + password
- HTTP headers **`WWW-Authenticate`** en **`Authorization`**
- HTTP is stateless: once logged in, the client sends the login info with each request
- More in the Security Lecture

TU Delft

URL – Syntax

- Uniform resource locators offer a standardized way to point to any resource on the Internet
- Not restricted to the http scheme, syntax slightly varies from scheme to scheme
- General format (adhered to by most schemes):

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

TU Delft

URL – Syntax details

The name of a piece of a resource. Only used by the client - the fragment is not transmitted to the server.

Parameters passed to gateway resources, i.e. applications [identified by the path] such as search engines.

Additional input parameters applications may require to access a resource on the server correctly. Can be set per path segment.

the local path to the resource

the port on which the server is expecting requests for the resource

domain name (host name) or numeric IP address of the server

the username/password (may be necessary to access a resource)

determines the protocol to use when connecting to the server.

<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>

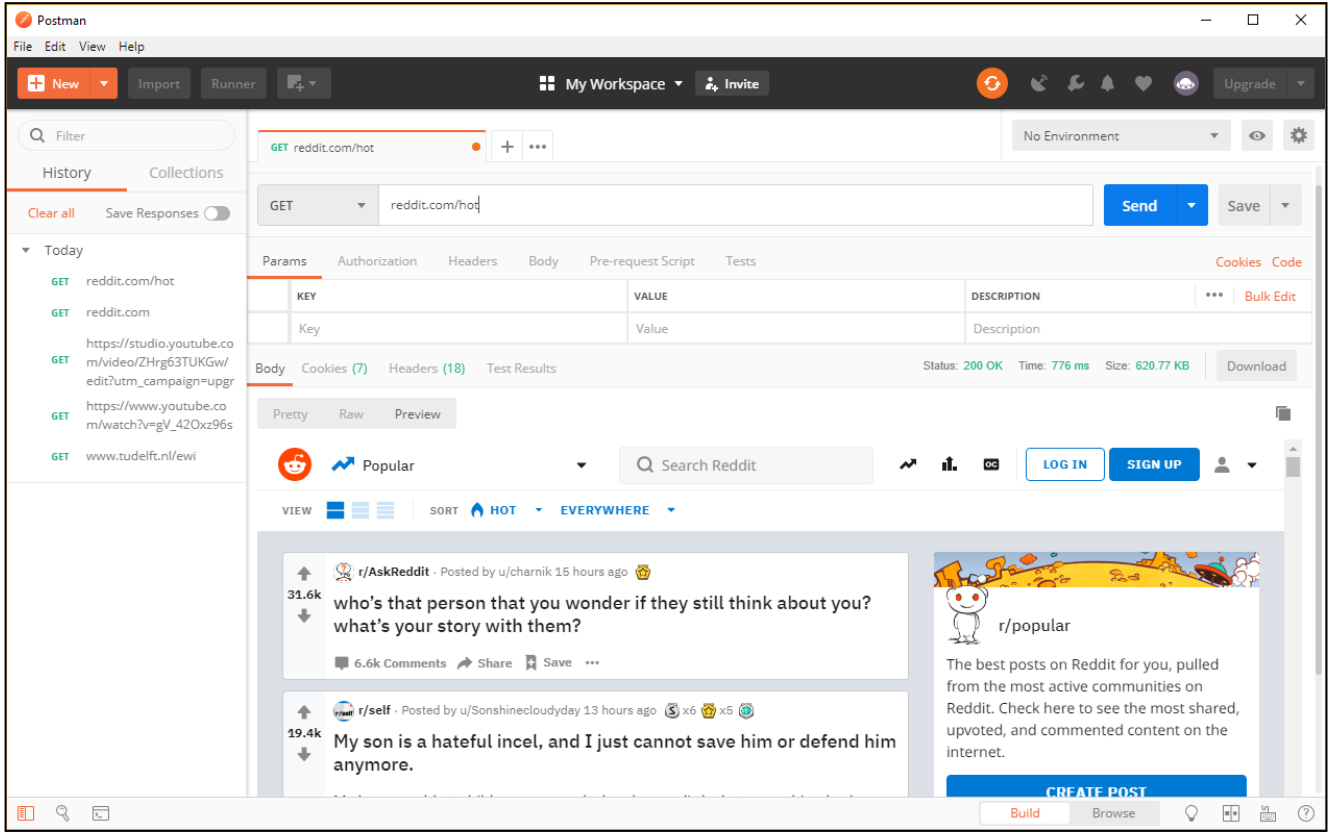
URL– Syntax query (cont'd)

https://www.google.com/search?q=TU+Delft&tbm=isch

- Query component is passed to the application accessed at web server ('gateway resource')
- Enables interactive web applications
- Pattern: name1=value1&name2=value2&…

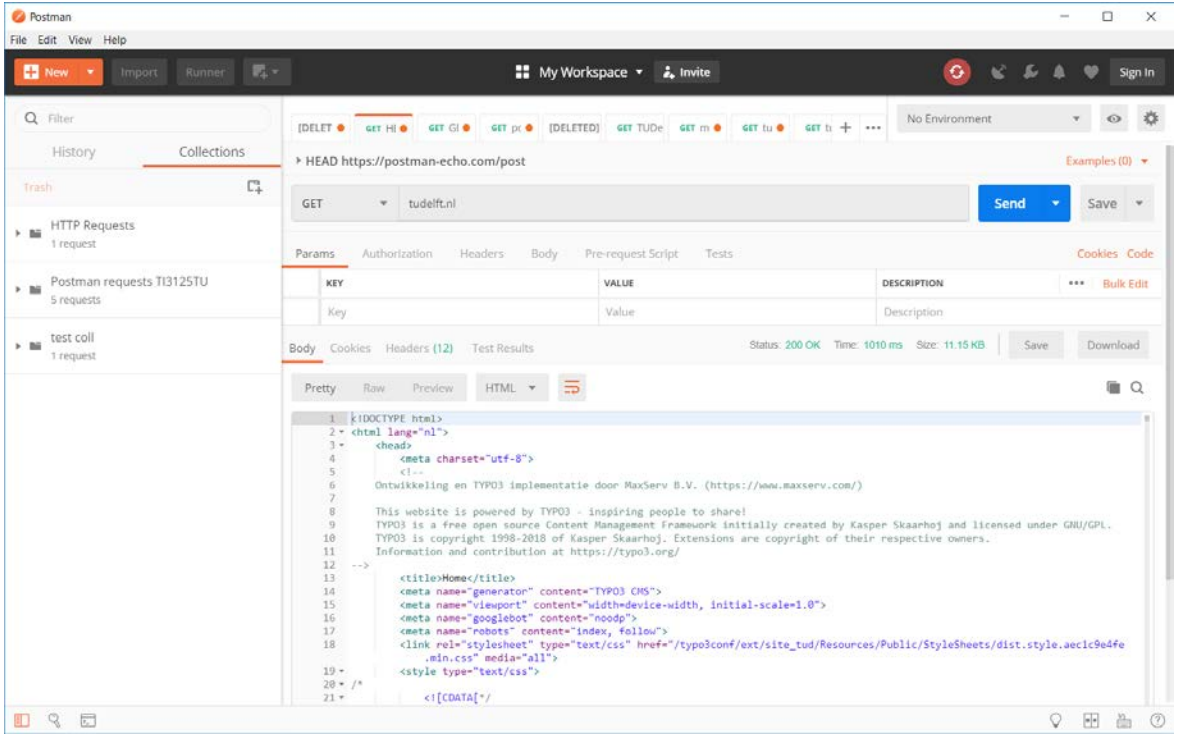
TU Delft

Postman



See: https://steelkiwi.com/blog/api-testing-useful-tools-postman-tutorial-and-hints/

Postman demo



Example

ONLINE ANALYSIS OF ENGINEERING FRAMEWORKS FOR VARIOUS LIFE CYCLE STAGES

ONLINE ANALYSIS OF ENGINEERING FRAMEWORKS FOR VARIOUS LIFE CYCLE STAGES

Conference Paper (PDF Available) · October 2010 *with* 164 Reads

Conference: IRIT, VSST'2010, At Toulouse, F, Volume: 1

Cite this publication

Bart H.M. Gerritsen
15.13 · Delft University of Technology

Bart Gerritsen@planet

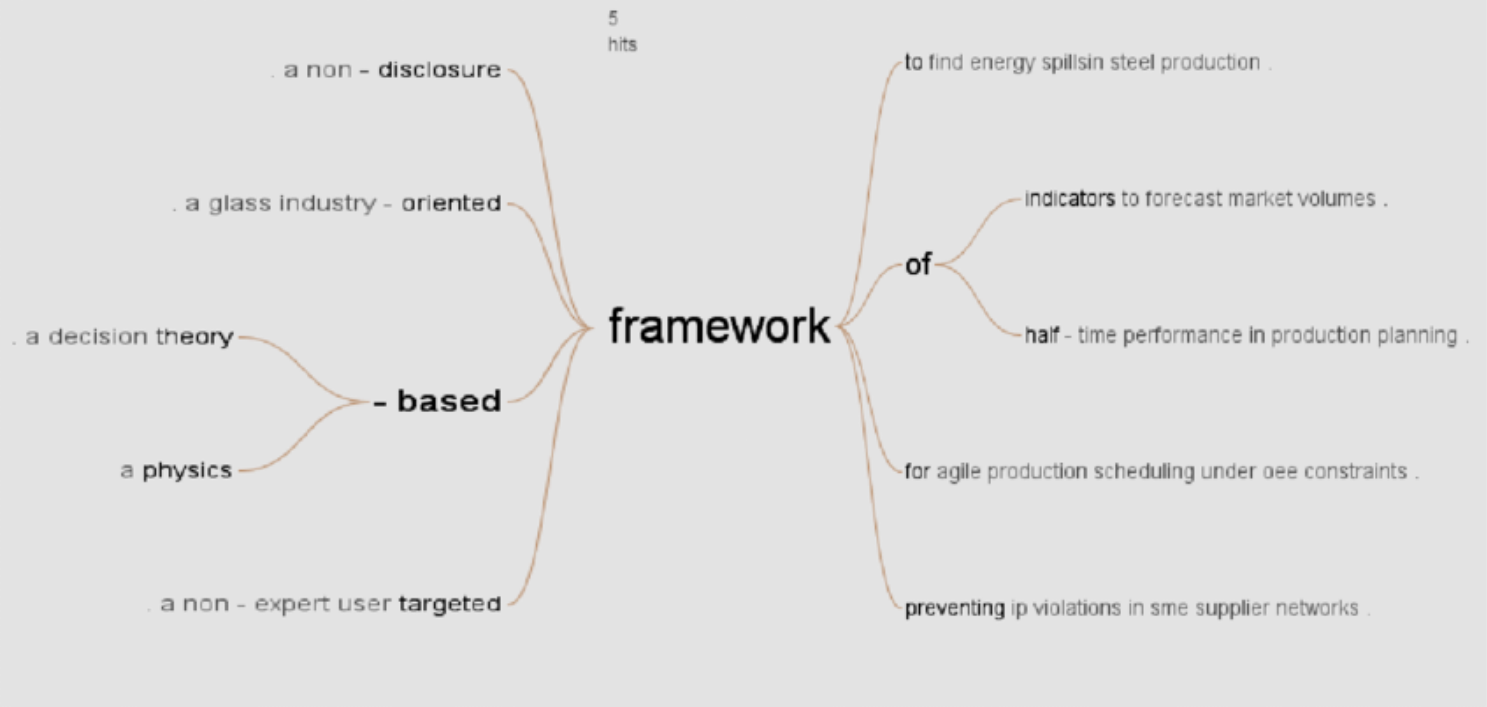
NI

Abstract

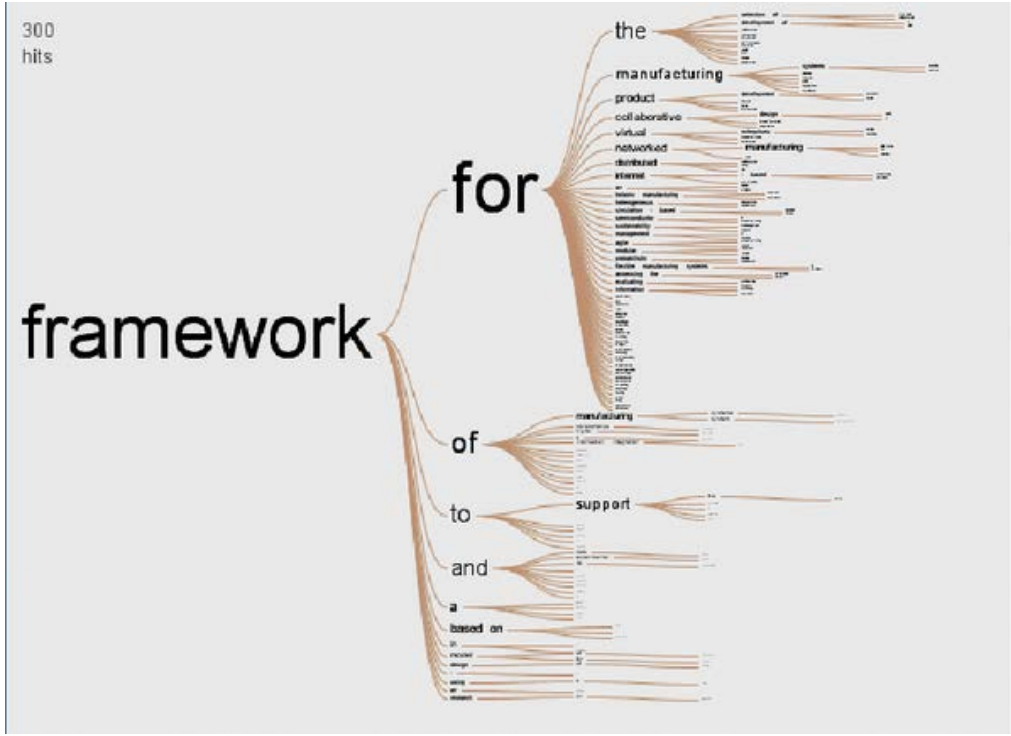
Effective, adaptable and extendable frameworks can be regarded a key element for future sustainable whole-life holistic product approaches, like the development of intelligent products and the internet-of-things, agile manufacturing, smart product bundling, closed-loop life cycle management, asset management, etc. Frameworks also play a vital role in enterprise architecture and business organization. Peculiar enough and in contrast with model, technology and knowledge, framework properties form an oddly neglected engineering research field, despite the fact that frameworks are being proposed by the dozens. The question arises how to apply, reuse and extend all these frameworks in a well-formed manner so as to be able to verify, implement and build upon earlier results in a harsh industrial setting. Moreover, framework analysis, if somehow structured, can complement to standards, protocols, ontology's and other formalizations. We carried out a comprehensive bibliographic study into frameworks and applied part-of-speech-based analysis on framework properties. To that extend, we mined the available meta data of thousands of engineering frameworks disclosed on the internet in the period after 2000 and developed methods and techniques to classify them according to indicative factors for goals, resources, application context, etc. To enable searching, matching, comparison and ranking, we explored measures indicative of similarity in (part of) the framework properties. More specifically, we calculated textual energy-based association strengths to determine cross-context framework applicability. This is believed to be a necessary first step towards the formation of federations of frameworks and contract-based transformation and frameworks. Results shown in this paper are promising, but a supporting (semi-)structure like a framework ontology, is believed to further raise the online search and analysis potential of our method.

TU Delft

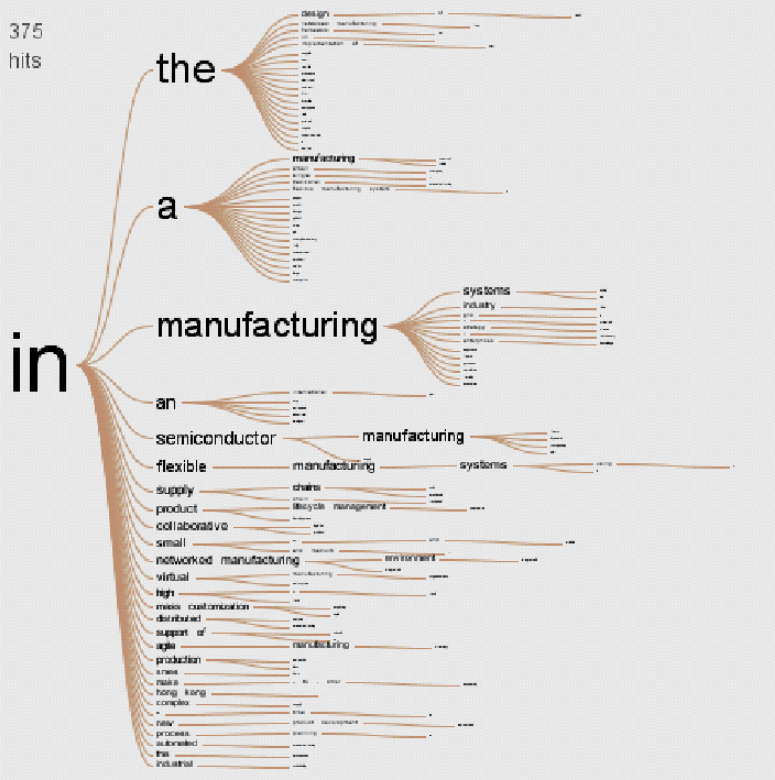
Looking for patterns …



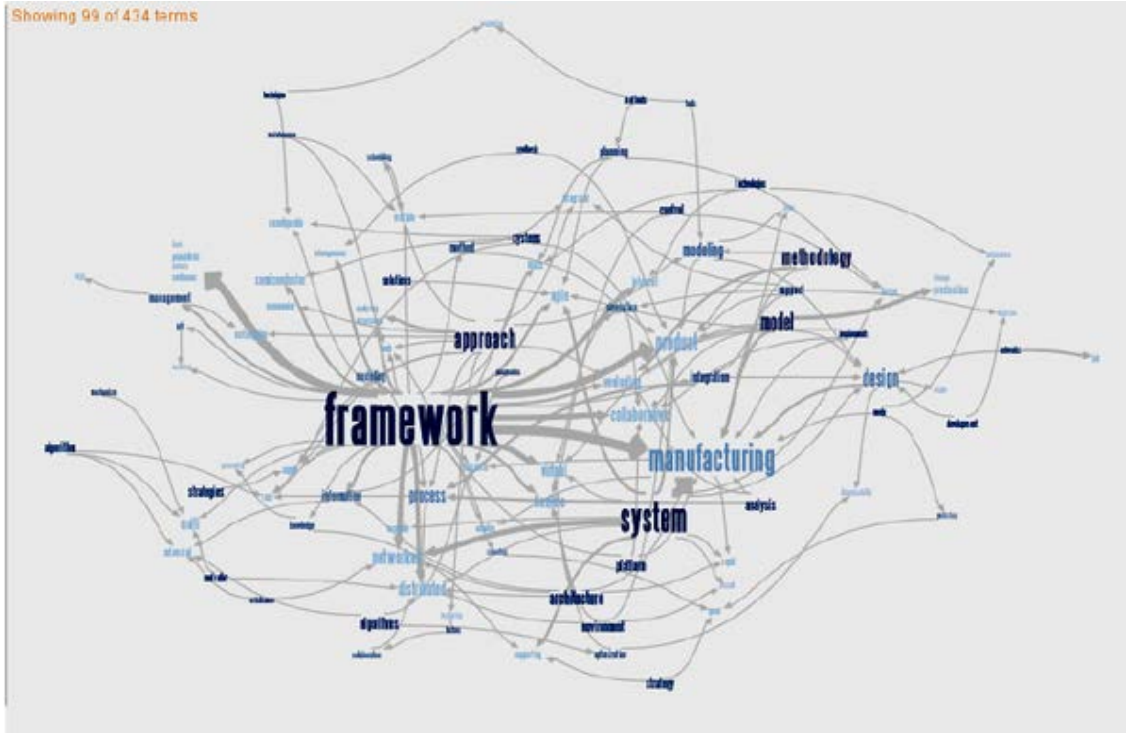
Looking for patterns (2)



Pattern element analysis



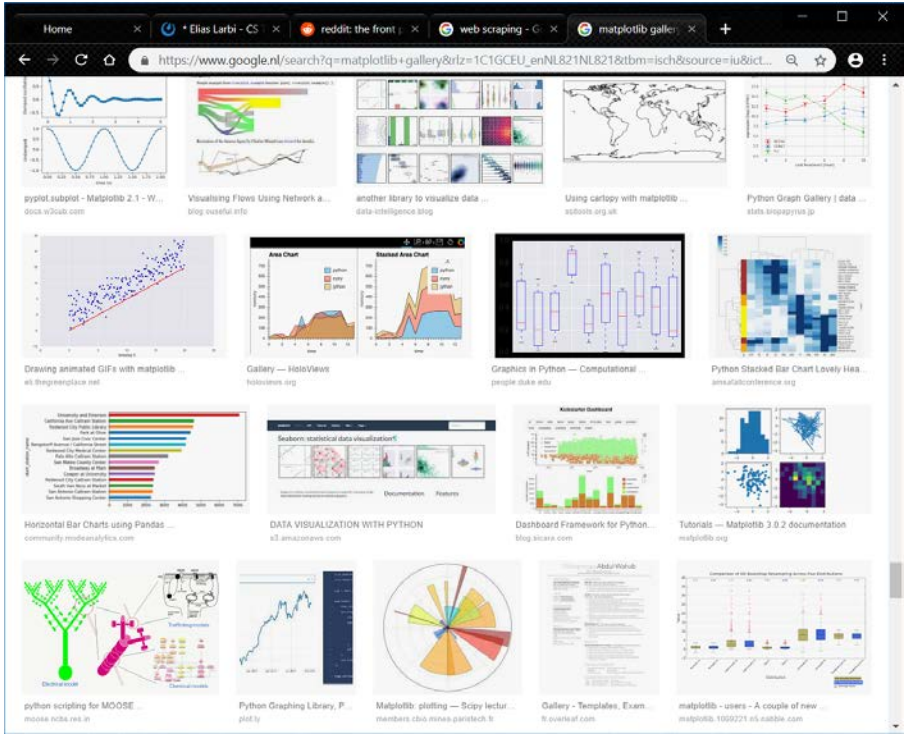
Looking for patterns (3)



Math analysis (Markov model)

d(t0,t)=-3			d(t0,t)=-2			d(t0,t)=-1			t0 = CT	d(t0,t)=1			d(t1,t)=2			d(t2,t)=3			d(t
Term	freq	%	Term	freq	%	Term	freq	%		Term	freq	%	Term	freq	%	Term	freq	%	Term
	13	6.2%	A	27	12.8%	A	34	16.1%	F	for	100	47.4%	the	16	7.6%	design	19	9.0%	of
	7	3.3%	An	10	4.7%	modeling	8	3.8%	F	to	19	9.0%	product	12	5.7%	product	8	3.8%	and
	5	2.4%	and	7	3.3%	integration	8	3.8%	F	of	17	8.1%	support	11	5.2%	and	6	2.8%	design
	5	2.4%	product	7	3.3%	a	7	3.3%	F	and	13	6.2%	on	5	2.4%	the	5	2.4%	information
ign	5	2.4%	an	5	2.4%	design	7	3.3%	F	based	7	3.3%	collaborative	5	2.4%	of	5	2.4%	development
	4	1.9%	design	5	2.4%	modelling	5	2.4%	F	with	4	1.9%	concurrent	3	1.4%	integration	5	2.4%	information
	3	1.4%	the	4	1.9%	system	5	2.4%	F	in	3	1.4%	Web-based	3	1.4%	development	4	1.9%	for
em	3	1.4%	information	4	1.9%	conceptual	5	2.4%	F	model	2	0.9%	evaluation	3	1.4%	a	4	1.9%	product
	3	1.4%	a	4	1.9%	integrated	5	2.4%	F	and	2	0.9%	for	3	1.4%	management	3	1.4%	platform
eric	2	0.9%	-	3	1.4%	theoretical	4	1.9%	F	supporting	2	0.9%	development	2	0.9%	to	2	0.9%	engineering
l	2	0.9%	of	3	1.4%	engineering	4	1.9%	F				to	2	0.9%	function	2	0.9%	concurrent
ly	2	0.9%	on	3	1.4%	software	4	1.9%	F				integrating	2	0.9%	support	2	0.9%	knowledge
duct	2	0.9%	data	3	1.4%	and	3	1.4%	F				its	2	0.9%	communication	2	0.9%	sharing
ributed	2	0.9%	function	2	0.9%	the	3	1.4%	F				virtual	2	0.9%	tool	2	0.9%	integration
nufacturing	2	0.9%	via	2	0.9%	recovery	3	1.4%	F				optimal	2	0.9%	distributed	2	0.9%	data
	2	0.9%	processes	2	0.9%	unified	3	1.4%	F				knowledge	2	0.9%	planning	2	0.9%	environment
			knowledge	2	0.9%	optimization	3	1.4%	F				new	2	0.9%	knowledge	2	0.9%	management
			Part	2	0.9%	management	3	1.4%	F				integrate	2	0.9%	engineering	2	0.9%	
			Research	2	0.9%	development	2	0.9%	F				design	2	0.9%	conceptual	2	0.9%	
			cost	2	0.9%	generic	2	0.9%	F							lifecycle	2	0.9%	
			models	2	0.9%	decision-making	2	0.9%	F							rapid	2	0.9%	
			design-decision	2	0.9%	making	2	0.9%	F							collaborative	2	0.9%	
			decision	2	0.9%	support	2	0.9%	F							products	2	0.9%	
						information	2	0.9%	F							enterprises	2	0.9%	
						on	2	0.9%	F										
						internet-based	2	0.9%	F										
						component	2	0.9%	F										
						reference	2	0.9%	F										
						exchange	2	0.9%	F										
						collaborative	2	0.9%	F										
						general	2	0.9%	F										
									211										
SING	79	37.4%	MISSING	16	7.6%	MISSING	0	0.0%	-	MISSING	33	15.6%	MISSING	33	15.6%	MISSING	38	18.0%	MISSING

Matplotlib


Home
* Elias Larbi - CS
reddit: the front
web scraping - G
matplotlib galler
https://www.google.nl/search?q=matplotlib+gallery&rlz=1C1GCEU_enNL821NL821&tbm=isch&source=iu&ict...
pyplot.subplot - Matplotlib 2.1 - W...
Visualising Flows Using Network a...
another library to visualize data ...
Using cartopy with matplotlib ...
Python Graph Gallery | data ...
Drawing animated GIFs with matplotlib ...
Gallery — HoloViews
Graphics in Python — Computational ...
Python Stacked Bar Chart Lovely Hea...
Horizontal Bar Charts using Pandas ...
DATA VISUALIZATION WITH PYTHON
Dashboard Framework for Python...
Tutorials — Matplotlib 3.0.2 documentation
python scripting for MOOSE ...
Python Graphing Library, P...
Matplotlib: plotting — Scipy lectur...
Gallery - Templates, Exam...
matplotlib - users - A couple of new ...

Some pointers to start …

- http://duspviz.mit.edu/tutorials/python-scraping
- https://www.dataquest.io/blog/web-scraping-tutorial-python/
- https://www.dataquest.io/blog/web-scraping-beautifulsoup/