

5.1 Open-source availability and infrastructure

The corgibrowser framework is an open-source project, and it's expected to be used by the Python community and start getting new features from users integrated into the modules of Web Crawling, Web Scraping, Data Management, and Data Analysis.

Currently on the Data Management module it's only supported one type of storage for data, but with more users on boarding on the project, it's expected to start getting more possibilities to integrate with cloud providers, or use local storage services to manage the tables, queues, or objects.

For new users to add features to the framework they will need to start a new Pull Request, pass all tests from the Pipeline, and get approval of one of the repository owners.

- Framework Tests: <https://github.com/j-enriquez/corgibrowser/tree/main/tests>

Users can use the framework directly installing it from the Python Package Index:

pip install corgibrowser

<https://pypi.org/project/corgibrowser/>

Or accessing the code from the GitHub repository:

gh repo clone j-enriquez/corgibrowser

<https://github.com/j-enriquez/corgibrowser>

When a Pull request is created, it will trigger a Pipeline to run validations for each change made, and at least 1 reviewer will be required to allow pull request to be merged:

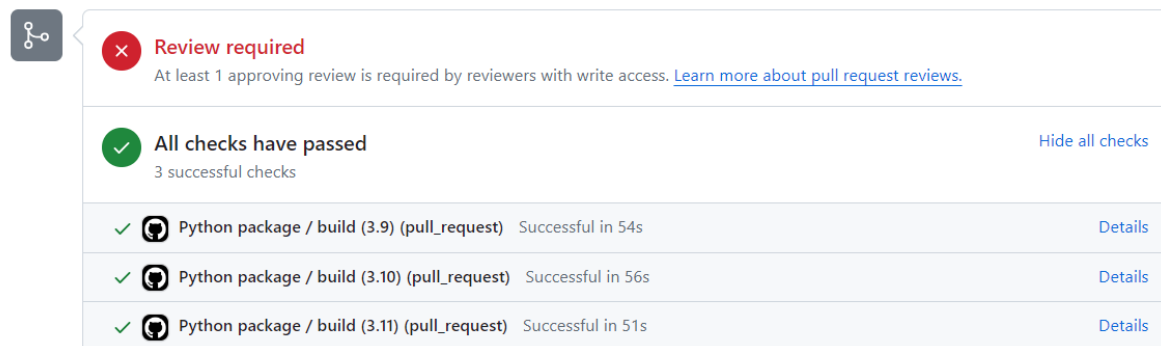


Figure 1: Example of pipeline requirements to merge pull request

And all tests for this specific pull request need to pass for build to pass.

```
build (3.11)
succeeded 2 minutes ago in 51s

✓  Test with pytest

1  ▶ Run pytest
9  ===== test session starts =====
10 platform linux -- Python 3.11.8, pytest-8.1.1, pluggy-1.4.0
11 rootdir: /home/runner/work/corgibrowser/corgibrowser
12 plugins: anyio-4.3.0
13 collected 18 items
14
15 tests/test_cloud_integration/test_add_urls_to_queue.py ..          [ 11%]
16 tests/test_crawler/test_crawler.py ..                            [ 22%]
17 tests/test_crawler/test_robots_txt.py .                           [ 27%]
18 tests/test_scraper/test_scraper.py .                             [ 33%]
19 tests/test_utils/test_names_generator.py .....                  [100%]
20
```

Figure 2: Example of Tests needed by the pipeline

5.2 Build in tools available to end-users for framework to be considered easy to use.

To start using the framework with default configurations, the users need to do the following steps:

- A) Set up Azure Storage Account
- B) Create a Crawler
- C) Create a Scraper

A) Instructions to set up an Azure Storage Account can be found in the Following url:

<https://learn.microsoft.com/en-us/azure/storage/common/storage-account-create?tabs=azure-portal>

A.1 Create storage account

<https://portal.azure.com/#create/Microsoft.StorageAccount-ARM>

Home > Storage accounts >

Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * thesisFramework

Resource group * (New) demo001 [Create new](#)

Instance details

Storage account name * frameworkdemo001

Region * (US) West US [Deploy to an Azure Extended Zone](#)

Performance * ☒ Standard: Recommended for most scenarios (general-purpose v2 account) ☐ Premium: Recommended for scenarios that require low latency.

Redundancy * Locally-redundant storage (LRS)

[Previous](#) [Next](#) [Review + create](#)

Figure 3: How to create a Storage Account

A.2 Review and create Storage Account

Create a storage account ...

Blob anonymous access	Disabled
Allow storage account key access	Enabled
Default to Microsoft Entra authorization in the Azure portal	Disabled
Minimum TLS version	Version 1.2
Permitted scope for copy operations (preview)	From any storage account

Networking

Network connectivity	Public endpoint (all networks)
Default routing tier	Microsoft network routing

Data protection

Point-in-time restore	Disabled
Blob soft delete	Enabled
Blob retainment period in days	7
Container soft delete	Enabled
Container retainment period in days	7
File share soft delete	Enabled
File share retainment period in days	7
Versioning	Disabled
Blob change feed	Disabled
Version-level immutability support	Disabled

Encryption

Encryption type	Microsoft-managed keys (MMK)
Enable support for customer-managed keys	Blobs and files only
Enable infrastructure encryption	Disabled

[Previous](#) [Next](#) [Create](#)

Figure 4: Review and create Storage Account

A.3) Retrieve access keys

<https://learn.microsoft.com/en-us/azure/storage/common/storage-account-keys-manage?tabs=azure-portal>

The account name and key1 will be used to access the Storage Account, this keys will allow the framework to communicate with the tables, queues and object storage.

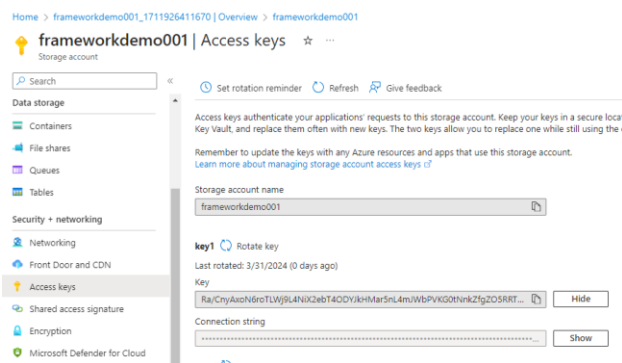


Figure 5: Where to find Access Keys of Storage Account

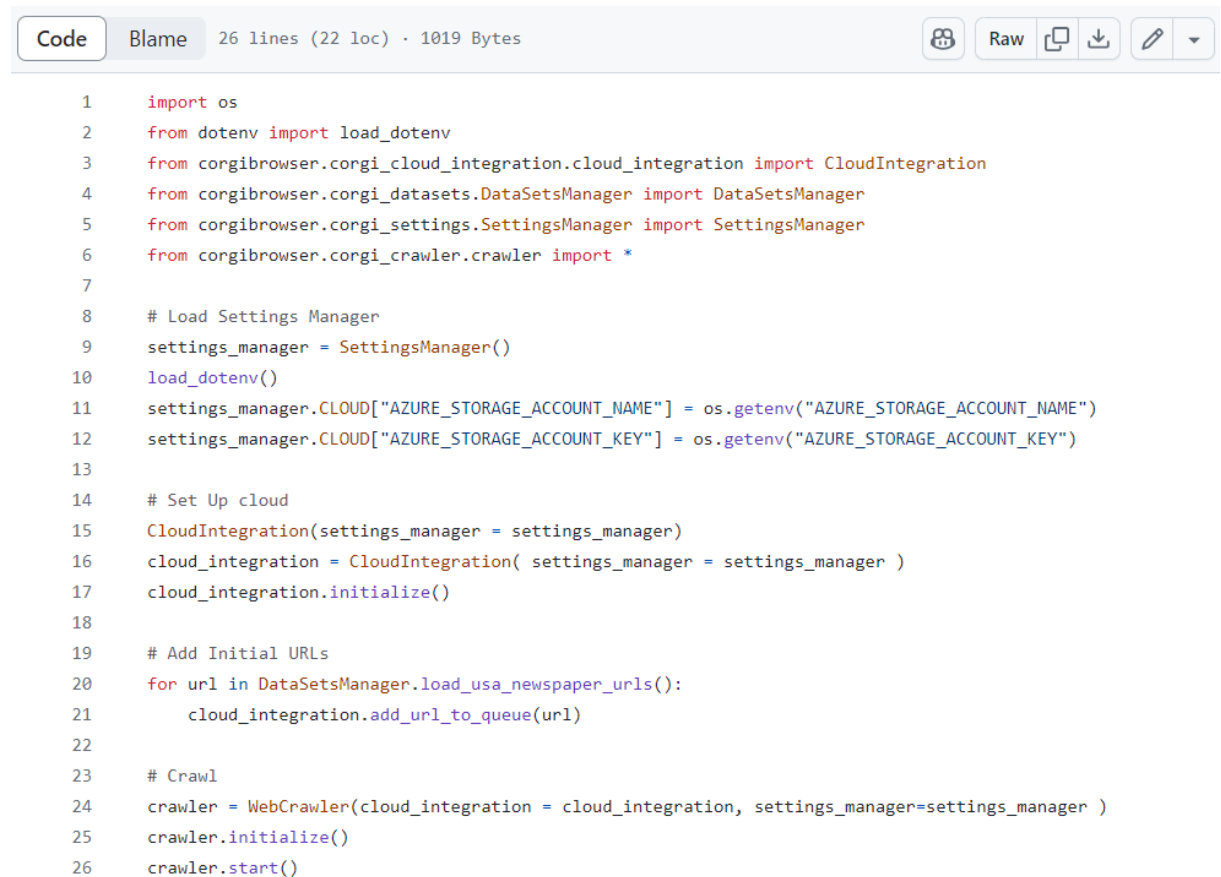
B) Create a Crawler

https://github.com/j-enriquez/corgibrowser/blob/main/user/default/demo_crawler.py

In the following example, from lines 1-17 the user is importing the framework and using the SettingsManager to update the default values to connect with the Storage Account. With the Cloud integration is creating an instance where it will be connecting the Storage Account.

On lines 20-21, is using the DataSetsManager that contains default datasets that are useful to initialize the crawler with certain domain urls.

On the lines 24-26 user is initializing the crawler and this is all the work needed from users to visit websites.



```
Code Blame 26 lines (22 loc) · 1019 Bytes
1 import os
2 from dotenv import load_dotenv
3 from corgibrowser.corgi_cloud_integration.cloud_integration import CloudIntegration
4 from corgibrowser.corgi_datasets.DataSetsManager import DataSetsManager
5 from corgibrowser.corgi_settings.SettingsManager import SettingsManager
6 from corgibrowser.corgi_crawler.crawler import *
7
8 # Load Settings Manager
9 settings_manager = SettingsManager()
10 load_dotenv()
11 settings_manager.CLOUD["AZURE_STORAGE_ACCOUNT_NAME"] = os.getenv("AZURE_STORAGE_ACCOUNT_NAME")
12 settings_manager.CLOUD["AZURE_STORAGE_ACCOUNT_KEY"] = os.getenv("AZURE_STORAGE_ACCOUNT_KEY")
13
14 # Set Up cloud
15 CloudIntegration(settings_manager = settings_manager)
16 cloud_integration = CloudIntegration( settings_manager = settings_manager )
17 cloud_integration.initialize()
18
19 # Add Initial URLs
20 for url in DataSetsManager.load_usa_newspaper_urls():
21     cloud_integration.add_url_to_queue(url)
22
23 # Crawl
24 crawler = WebCrawler(cloud_integration = cloud_integration, settings_manager=settings_manager )
25 crawler.initialize()
26 crawler.start()
```

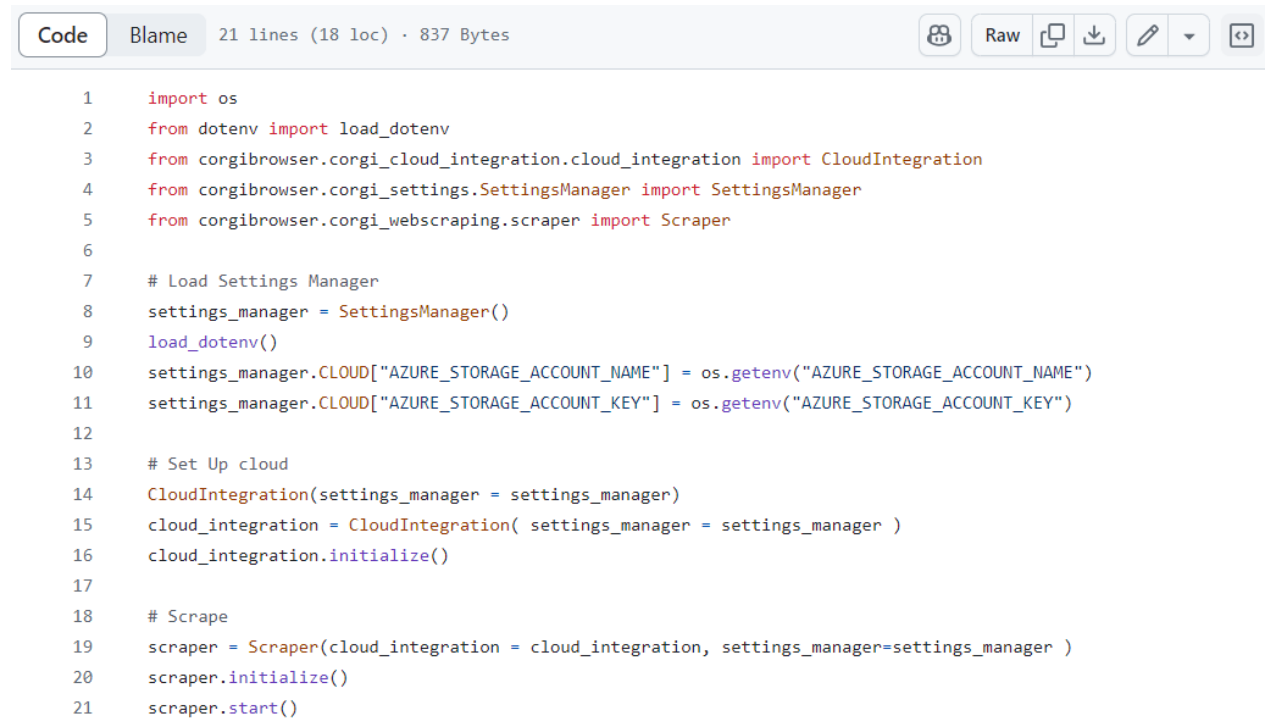
Figure 6: Code of example default crawler

C) Create a Web Scraper:

https://github.com/j-enriquez/corgibrowser/blob/main/user/default/demo_scraper.py

Lines 1-16 follow the similar steps from the crawler to import the packages, configure the settings manager, and connections with cloud provider.

From line 19-21 it's what user will run to start working on web scraping.



```
1  import os
2  from dotenv import load_dotenv
3  from corgibrowser.corgi_cloud_integration.cloud_integration import CloudIntegration
4  from corgibrowser.corgi_settings.SettingsManager import SettingsManager
5  from corgibrowser.corgi_web scraping.scraper import Scraper
6
7  # Load Settings Manager
8  settings_manager = SettingsManager()
9  load_dotenv()
10 settings_manager.CLOUD["AZURE_STORAGE_ACCOUNT_NAME"] = os.getenv("AZURE_STORAGE_ACCOUNT_NAME")
11 settings_manager.CLOUD["AZURE_STORAGE_ACCOUNT_KEY"] = os.getenv("AZURE_STORAGE_ACCOUNT_KEY")
12
13 # Set Up cloud
14 CloudIntegration(settings_manager = settings_manager)
15 cloud_integration = CloudIntegration( settings_manager = settings_manager )
16 cloud_integration.initialize()
17
18 # Scrape
19 scraper = Scraper(cloud_integration = cloud_integration, settings_manager=settings_manager )
20 scraper.initialize()
21 scraper.start()
```

Figure 7: Code of example default scraper

5.3 Compliance modules to assure Robots.txt file is respected.

Robots.txt is a file provided by owners of websites to allow or disallow a particular use of a user-agent.

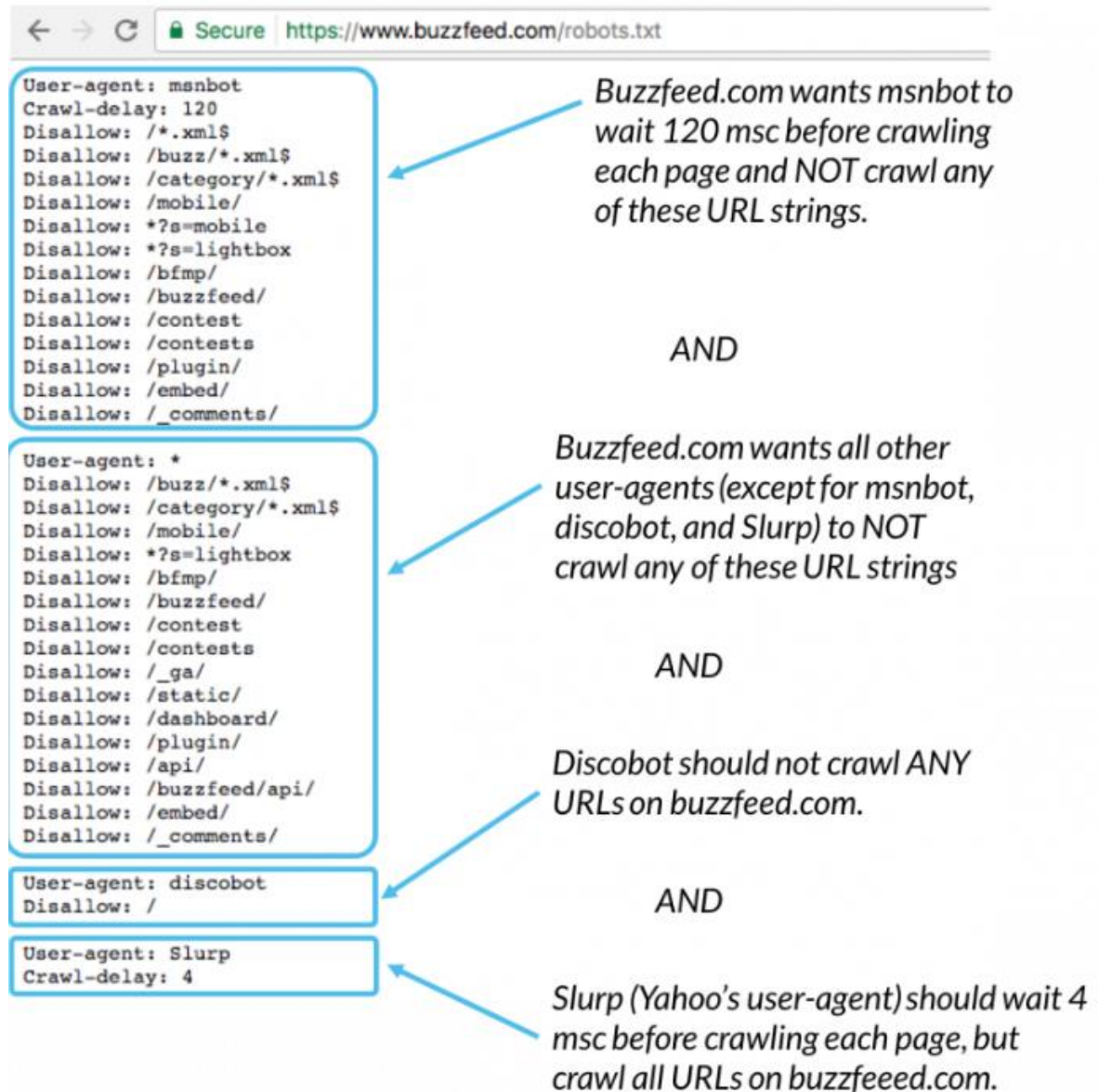


Figure 8: Example of robots.txt file by moz.com

On the previous example from moz.com, we can see how buzzfeed wants each of the user agents to interact with the website, allowing certain paths to be visited, and including the crawl-delay.


On Corgibrowser this is managed by the “[RobotsCache](#)”, this class uses the library [urllib.robotparser](#) to determine if crawler can fetch.

```
can_fetch(useragent, url) ¶  
Returns True if the useragent is allowed to fetch the url according to the rules contained in the parsed robots.txt file.
```

Figure 9: Python urllib.robotparser method

At the initialization of each crawler before making a visit to a website, the framework reviews the /robots.txt file and determines if the call can be made based on the can_fetch parameters, and if we are compliant with the website crawl-delay (if website does not contain a crawl-delay, default by the framework is 1).

To be compliant with the crawl-delay rules from the website, corgibrowser implemented a table called corgiwebtrotting, where based on the last timestamp we can determine the last time the website was visited. Access to this table is by PartitionKey and RowKey, which makes the access of the information faster by calling directly to the index we want to retrieve. Logic of can_process_domain is on the [Cloud Integration](#) module. There are future improvements in this area to allow the use of an in-memory cache like Redis instead of a table and thanks to the modular and domain area approach of the framework, the logic will stay the same for crawler, and only adaptations on the DataManagement domain will be needed.

 Add filter

Sorting all 1228 items

<input type="checkbox"/>	PartitionKey	RowKey	Timestamp	ThrottlingLimitSeconds ↓
<input type="checkbox"/>	RateLimit	www.wcgazette.com	2024-03-25T19:23:45.09...	10
<input type="checkbox"/>	RateLimit	www.westseattleherald.com	2024-03-25T09:26:31.25...	10
<input type="checkbox"/>	RateLimit	www.whyy.org	2024-03-25T19:22:19.24...	10
<input type="checkbox"/>	RateLimit	thechampionnewspaper.com	2024-03-25T19:22:00.68...	5
<input type="checkbox"/>	RateLimit	longislandweekly.com	2024-03-25T19:21:50.42...	3
<input type="checkbox"/>	RateLimit	www.dallasvoice.com	2024-03-25T19:23:26.28...	3
<input type="checkbox"/>	RateLimit	www.decorahnewspapers.com	2024-03-25T19:22:22.35...	3
<input type="checkbox"/>	RateLimit	www.georgiabulletin.org	2024-03-25T19:22:38.81...	3
<input type="checkbox"/>	RateLimit	www.kten.com	2024-03-25T19:23:32.98...	3
<input type="checkbox"/>	RateLimit	www.wgmd.com	2024-03-25T19:22:18.80...	3
<input type="checkbox"/>	RateLimit	www.whitewaterbanner.com	2024-03-25T19:23:45.89...	3

Figure 10: corgiwebthrottling table

For compliance and to be able to have proof of all requests made, on the table corgiwebrequestslog will be stored each http call made to an external service, if website specific url was not visited due to website being throttled or robots.txt not allowing a specific path, this data will also be stored on the logs table. Each log will consist of domain partition key for the specific website, the timestamp when this call was made as the row key to allow queries by time to be made easier, instance id of the unique process, original url from the website to visit, and the status code responded from website or the reason why website was not visited.

Tables > corgiwebrequestslog

Authentication method: Access key (Switch to Microsoft Entra user account)

Add filter

Showing the first 100 items

<input type="checkbox"/>	PartitionKey	RowKey	Timestamp	InstanceId	StatusCode	Url
<input type="checkbox"/>	abc30.com	20240325T084039Z	2024-03-25T08:40:39.98...	demo001-20240325084...	200	http://abc30.com/
<input type="checkbox"/>	abc30.com	20240325T085421Z	2024-03-25T08:54:21.23...	demo001-20240325085...	200	http://abc30.com/
<input type="checkbox"/>	abc30.com	20240325T090646Z	2024-03-25T09:06:46.70...	demo001-20240325090...	200	http://abc30.com/
<input type="checkbox"/>	abc7chicago.com	20240325T084040Z	2024-03-25T08:40:40.30...	demo001-20240325084...	200	http://abc7chicago.com/
<input type="checkbox"/>	abc7chicago.com	20240325T085421Z	2024-03-25T08:54:21.88...	demo001-20240325085...	200	http://abc7chicago.com/
<input type="checkbox"/>	abc7chicago.com	20240325T090647Z	2024-03-25T09:06:47.70...	demo001-20240325090...	200	http://abc7chicago.com/
<input type="checkbox"/>	abcnews.go.com	20240325T084039Z	2024-03-25T08:40:39.43...	demo001-20240325084...	200	https://abcnews.go.com/
<input type="checkbox"/>	abcnews.go.com	20240325T084040Z	2024-03-25T08:40:40.52...	demo001-20240325084...	Throttled	https://abcnews.go.com/
<input type="checkbox"/>	abcnews.go.com	20240325T085422Z	2024-03-25T08:54:22.27...	demo001-20240325085...	200	https://abcnews.go.com/
<input type="checkbox"/>	abcnews.go.com	20240325T090648Z	2024-03-25T09:06:48.14...	demo001-20240325090...	200	https://abcnews.go.com/
<input type="checkbox"/>	acuoptimist.com	20240325T084039Z	2024-03-25T08:40:39.83...	demo001-20240325084...	NotAllowed	http://acuoptimist.com/
<input type="checkbox"/>	acuoptimist.com	20240325T084040Z	2024-03-25T08:40:40.81...	demo001-20240325084...	NotAllowed	http://acuoptimist.com/
<input type="checkbox"/>	acuoptimist.com	20240325T085422Z	2024-03-25T08:54:22.64...	demo001-20240325085...	NotAllowed	http://acuoptimist.com/
<input type="checkbox"/>	advancetitan.com	20240325T084039Z	2024-03-25T08:40:39.93...	demo001-20240325084...	NotAllowed	https://advancetitan.com/

Figure 11: corgiwebrequestslog table

On the web scraping domain, when we are reviewing all the new urls to see if this can be added to the queue, the framework also validates if this URL's can be visited by Robots.txt by making a call to the **can_fetch** method before processing a new url found.

5.4 Partitioned Tables/Queuing system to allow customized search patterns to target specific websites

Users will be able to decide which website domains the crawler/scrapper instances will be running and getting values from its queues. For each website domain, the DataManagement area will create a Table, a Queue, and a Container.

Because each domain is partitioned the user can decide how to split the work between instances and prioritize the search of certain website domains based on the business needs.

If the setting “QUEUE_ONLY_DOMAINS” variable is empty, the default value will be to retrieve all domains available to visit from the table “[corgiwebqueuepreference](#)”.

Tables > corgiwebqueuepreference

Authentication method: Access key (Switch to Microsoft Entra user account)

🔍 Add filter

Showing the first 1100 items

<input type="checkbox"/>	PartitionKey	RowKey	Timestamp	ItemsToPopFromQueue	VisibilityTimeout
<input type="checkbox"/>	bufferdata	abc30com	2024-03-25T09:04:54.47...	5	18000
<input type="checkbox"/>	bufferdata	abc7chicagocom	2024-03-25T09:05:31.39...	5	18000
<input type="checkbox"/>	bufferdata	abcnewsgocom	2024-03-25T09:05:21.65...	5	18000
<input type="checkbox"/>	bufferdata	acuoptimistcom	2024-03-25T09:06:27.35...	5	18000
<input type="checkbox"/>	bufferdata	advancetitancom	2024-03-25T09:06:38.13...	5	18000
<input type="checkbox"/>	bufferdata	advocatejbuedu	2024-03-25T09:04:54.36...	5	18000

Figure 12: Example of corgiwebqueuepreference table

On the following example the user is creating a crawler that will only be visiting the websites from the specified domains and using the values on the table “[corgiwebqueuepreference](#)” for ItemsToPopFromQueue and VisibilityTimeout:

```
# Crawl
settings_manager.CRAWLER["QUEUE_ONLY_DOMAINS"] = [
    "fbrefcom",
    "wwweluniversalcommx",
    "abcnewsgocom",
    "wwwcnncom"]

crawler = WebCrawler(cloud_integration = cloud_integration, settings_manager=settings_manager )
crawler.initialize()
crawler.start()
```

Figure 13: Code example of Crawler with specific domains

The following image shows an example of the queues by domain:

004 | Queues ⚙️ ☆ ...

+ Queue Refresh Delete Give feedback

Search queues by prefix

Queue	Url
<input type="checkbox"/> abc30com	https://democorgibrowser004.queue.core.windows.net/abc30com
<input type="checkbox"/> abc7chicagocom	https://democorgibrowser004.queue.core.windows.net/abc7chicagocom
<input type="checkbox"/> abcnewsgocom	https://democorgibrowser004.queue.core.windows.net/abcnewsgocom
<input type="checkbox"/> acuoptimistcom	https://democorgibrowser004.queue.core.windows.net/acuoptimistcom
<input type="checkbox"/> advancettitancom	https://democorgibrowser004.queue.core.windows.net/advancettitancom
<input type="checkbox"/> advocatetjbuedu	https://democorgibrowser004.queue.core.windows.net/advocatetjbuedu

Figure 14: Domain Specific Queues

And each of the queues will store the messages of the new URL’s to visit following the schema of [corgi_web_queue_version_1](#)

Refresh + Add message Dequeue message Clear queue Give feedback

Authentication method: Access key (Switch to Microsoft Entra user account)

Search to filter items...

Id	Message text	Insertion time
9bee774b-25af-4921-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
ftc484f9-1469-497d-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
1fd48c6-692f-4212-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
3f2b1559-cc23-4379-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
6c35746a-8570-44a-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
bb50cce3-142c-4a41-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
3a877a9e-e8aa-421e-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
e6980b7f-55ae-4ce7-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
d494bf74-8bf9-45ae-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
ac6755c6-5afd-466c-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1
ce54c073-a1e0-4417-...	["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "...	3/25/2024, 1

Message properties

ID
9bee774b-25af-4921-8372-1aca95cd2a93

MESSAGE BODY
["toVisitUrl": "http://abc30.com/", "version": 1, "originalDomain": "http://abc30.com/", "originalUrl": "http://abc30.com/", "partitionKey": "http://abc30.com/", "rowKey": "98f", "metadata": {}, "timestamp": "2024-03-25T08:38:51.499442", "status": "pending"]

INSERTION TIME
Mon Mar 25 2024 01:38:51 GMT-0700 (Pacific Daylight Time)

EXPIRATION TIME
Mon Apr 01 2024 01:38:51 GMT-0700 (Pacific Daylight Time)

DEQUEUE COUNT
-

Figure 15: Message properties of queues

The tables are also partitioned by domain

r004 | Tables ✨ ☆ ...

+ Table Refresh Delete Give feedback

Authentication method: Access key (Switch to Microsoft Entra user account)

Search tables by prefix

Table	Url
<input type="checkbox"/> abc30com	https://democorgibrowser004.table.core.windows.net/abc30com
<input type="checkbox"/> abc7chicagocom	https://democorgibrowser004.table.core.windows.net/abc7chicagocom
<input type="checkbox"/> abcnewsgocom	https://democorgibrowser004.table.core.windows.net/abcnewsgocom
<input type="checkbox"/> acuoptimistcom	https://democorgibrowser004.table.core.windows.net/acuoptimistcom
<input type="checkbox"/> advancetitancom	https://democorgibrowser004.table.core.windows.net/advancetitancom

Figure 16: Domain Specific Tables

And each of the tables will follow the shema of [corgiweb_queuepreference](#), containing relevant information about the visited urls and the urls to visit. This table can help in case re-visit rules are needed to add visited URLs to the queue.

...

+ Add entity Refresh Delete

Tables > abc30com

Authentication method: Access key (Switch to Micro

Add filter

Showing all 1 items

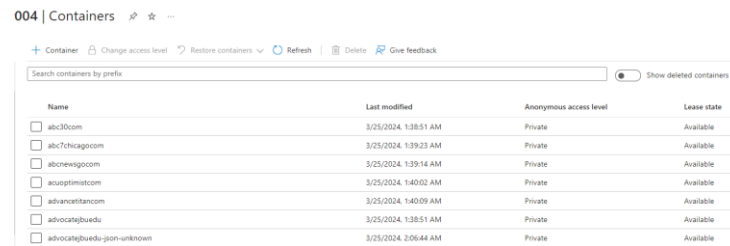
<input checked="" type="checkbox"/>	PartitionKey	RowKey
<input checked="" type="checkbox"/>	abc30com	http%3A%2F%2Fabc30.com%2F

Edit entity

Property Name	Type	Value
PartitionKey	String	abc30com
RowKey	String	http%3A%2F%2Fabc30.com%2F
Timestamp	DateTi...	2024-03-25T09:06:46.871638Z
BlobName	String	http%3A%2F%2Fabc30.com%2F
ContainerName	String	abc30com
FullUrl	String	http://abc30.com/
OriginalDomain	String	abc30.com
OriginalUrl	String	http://abc30.com/
Status	String	Processed
TableName	String	abc30com
visitedCount	Int32	4

Figure 17: corgiweb_queuepreference entity values

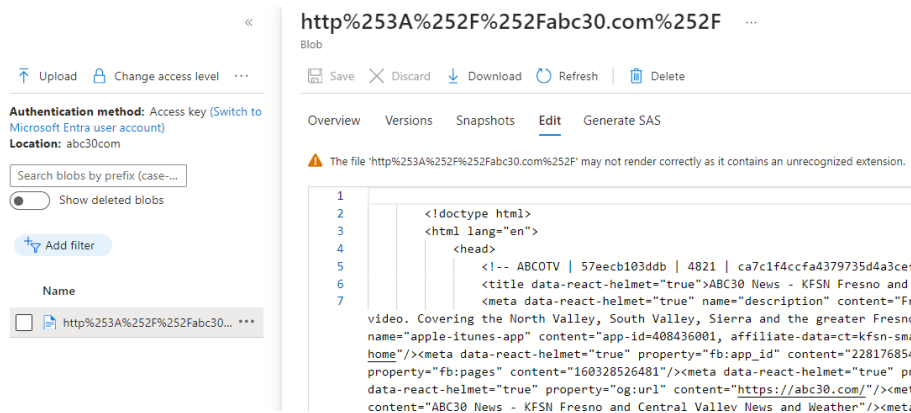
Containers are also partitioned by domain.



Name	Last modified	Anonymous access level	Lease state
<input type="checkbox"/> abc30com	3/25/2024 1:38:51 AM	Private	Available
<input type="checkbox"/> abc7chicagocom	3/25/2024 1:39:23 AM	Private	Available
<input type="checkbox"/> abcnewsgocom	3/25/2024 1:39:14 AM	Private	Available
<input type="checkbox"/> asumptimocom	3/25/2024 1:40:02 AM	Private	Available
<input type="checkbox"/> advancementcom	3/25/2024 1:40:09 AM	Private	Available
<input type="checkbox"/> advocatejbu.edu	3/25/2024 1:38:51 AM	Private	Available
<input type="checkbox"/> advocatejbu.edu-jon-unknown	3/25/2024 2:06:44 AM	Private	Available

Figure 18: Domain Specific Containers

And initially from the crawler the data is stored in the form of the original HTML



Authentication method: Access key (Switch to Microsoft Entra user account)
Location: abc30com

Search blobs by prefix (case-...)
Show deleted blobs

Add filter

Name

☐ http%253A%252F%252Fabc30.com%252F

http%253A%252F%252Fabc30.com%252F ...

Save Discard Download Refresh Delete

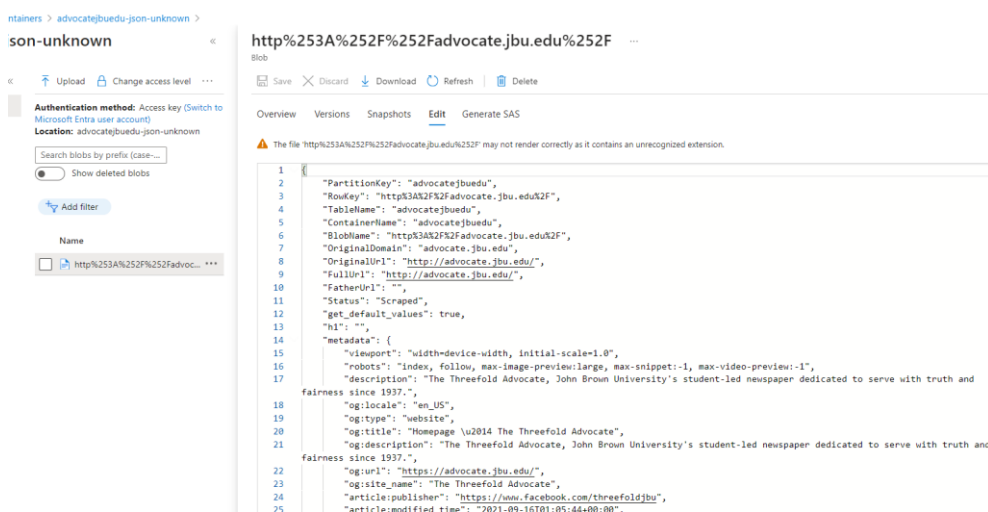
Overview Versions Snapshots Edit Generate SAS

The file 'http%253A%252F%252Fabc30.com%252F' may not render correctly as it contains an unrecognized extension.

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <!-- ABC0TV | 57eecb103ddb | 4821 | ca7c1f4ccfa4379735d4a3cef5
5     <title data-react-helmet="true">ABC30 News - KFSN Fresno and C
6     <meta data-react-helmet="true" name="description" content="Fre
7     video. Covering the North Valley, South Valley, Sierra and the greater Fresno
      name="apple-itunes-app" content="app-id=408436001, affiliate-data=ct=kfsn-smar
      home"/><meta data-react-helmet="true" property="fb:app_id" content="2281768540
      property="fb:pages" content="160328526481"/><meta data-react-helmet="true" pro
      data-react-helmet="true" property="og:url" content="https://abc30.com/"><meta
      content="ABC30 News - KFSN Fresno and Central Valley News and Weather"/><meta
```

Figure 19: Example of downloaded url as HTML

After if the url is already processed by the Web scraping module, the data will be stored in form of a JSON.



Authentication method: Access key (Switch to Microsoft Entra user account)
Location: advocatejbu.edu-jon-unknown

Search blobs by prefix (case-...)
Show deleted blobs

Add filter

Name

☐ http%253A%252F%252Fadvocate.jbu.edu%252F

http%253A%252F%252Fadvocate.jbu.edu%252F ...

Save Discard Download Refresh Delete

Overview Versions Snapshots Edit Generate SAS

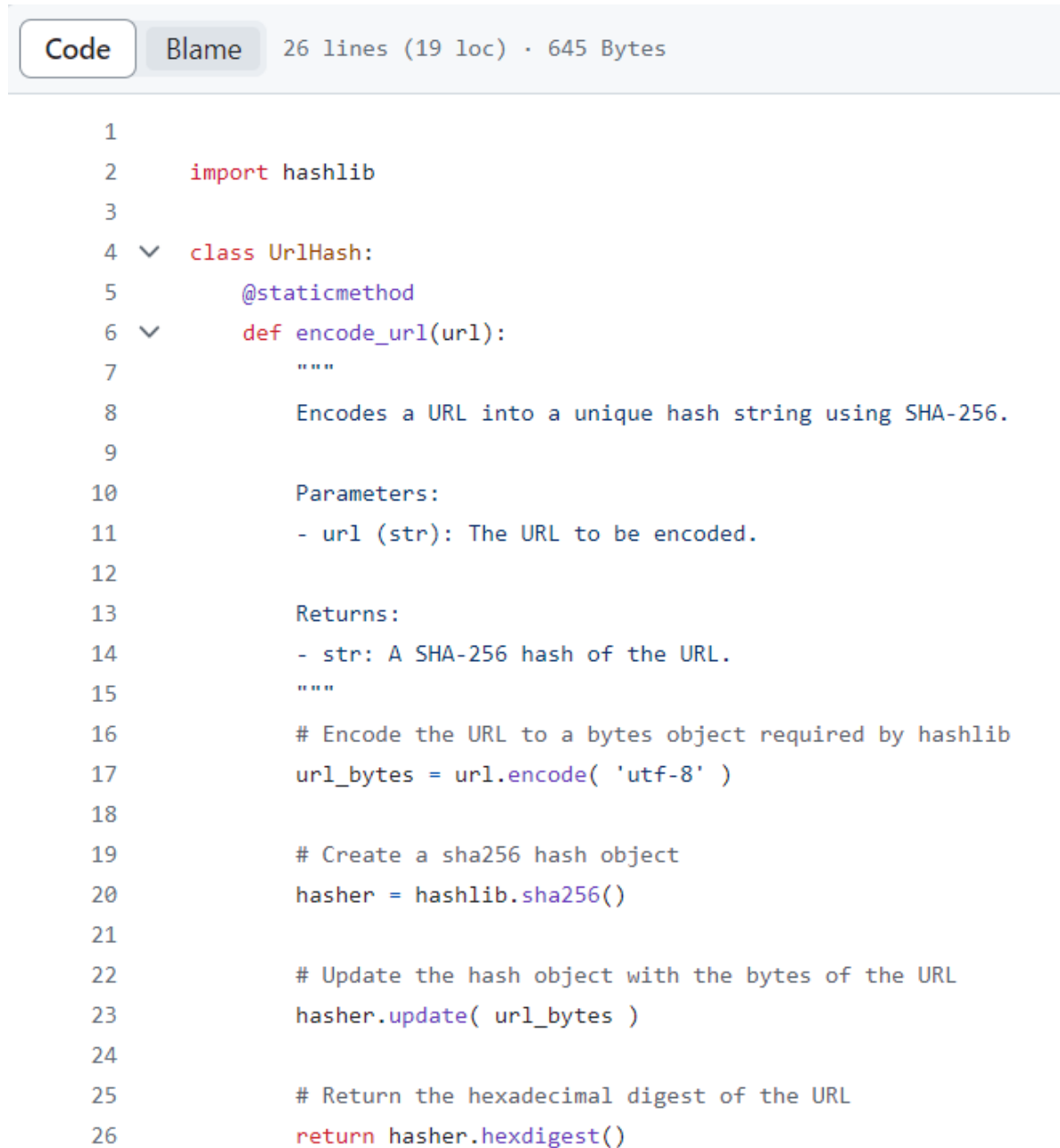
The file 'http%253A%252F%252Fadvocate.jbu.edu%252F' may not render correctly as it contains an unrecognized extension.

```
1 {
2   "PartitionKey": "advocatejbu.edu",
3   "RowKey": "http%253A%252F%252Fadvocate.jbu.edu%252F",
4   "TableName": "advocatejbu.edu",
5   "ContainerName": "advocatejbu.edu",
6   "BlobName": "http%253A%252F%252Fadvocate.jbu.edu%252F",
7   "OriginalDomain": "advocate.jbu.edu",
8   "OriginalUrl": "http://advocate.jbu.edu/",
9   "FullUrl": "http://advocate.jbu.edu/",
10  "FatherUrl": "",
11  "Status": "Scraped",
12  "get_default_values": true,
13  "hl": "",
14  "metadata": {
15    "viewport": "width=device-width, initial-scale=1.0",
16    "robots": "Index, follow, max-image-preview:large, max-snippet:-1, max-video-preview:-1",
17    "description": "The Threefold Advocate, John Brown University's student-led newspaper dedicated to serve with truth and
      fairness since 1937.",
18    "og:locale": "en_US",
19    "og:type": "website",
20    "og:title": "Homepage \u2014 The Threefold Advocate",
21    "og:description": "The Threefold Advocate, John Brown University's student-led newspaper dedicated to serve with truth and
      fairness since 1937.",
22    "og:url": "https://advocate.jbu.edu/",
23    "og:site_name": "The Threefold Advocate",
24    "article:publisher": "https://www.facebook.com/threefoldjbu",
25    "article:modified time": "2021-09-16T01:05:44-00:00".
26  }
```

Figure 20: Example of Downloaded and processed URL as a JSON

5.5 Visited websites hash table

For each url that is added to the webcrawling tables or queues, the framework encodes the url with SHA-256 using [UrlHash](#) class.



The image shows a code editor interface with a light blue header bar. On the left, there are two buttons: 'Code' (highlighted) and 'Blame'. To the right of these buttons, it says '26 lines (19 loc) · 645 Bytes'. Below the header, the code for the `UrlHash` class is displayed. The code is as follows:

```
1
2     import hashlib
3
4  ✓ class UrlHash:
5         @staticmethod
6  ✓     def encode_url(url):
7         """
8             Encodes a URL into a unique hash string using SHA-256.
9
10            Parameters:
11            - url (str): The URL to be encoded.
12
13            Returns:
14            - str: A SHA-256 hash of the URL.
15            """
16            # Encode the URL to a bytes object required by hashlib
17            url_bytes = url.encode( 'utf-8' )
18
19            # Create a sha256 hash object
20            hasher = hashlib.sha256()
21
22            # Update the hash object with the bytes of the URL
23            hasher.update( url_bytes )
24
25            # Return the hexadecimal digest of the URL
26            return hasher.hexdigest()
```

Figure 21: Code of UrlHash

Each Hash is added to the corgiwebhashtable, stored by partition and hash as the rowkey.

Tables > corgiwebhashtable

Authentication method: Access key (Switch to Microsoft Entra user account)

Add filter

Showing the first 100 items

<input type="checkbox"/>	PartitionKey	RowKey	Timestamp	FullUrl
<input type="checkbox"/>	advocatejbuedu	013bd46e1256879037c0332a4afe913bfcea1ab3095fee2e9962fc4274bf4707	2024-03-25T09:06:44.64...	https://advocate.jbu.edu...
<input type="checkbox"/>	advocatejbuedu	04b04c76087b612a84be5623b5d2f81d18d4a018b343e5e64174f39577b47de5	2024-03-25T09:06:47.41...	https://advocate.jbu.edu...
<input type="checkbox"/>	advocatejbuedu	055e9ff2da76a7d6cc9a5a97efb72b29c120ffa5a40602e7ac2400182f28b615	2024-03-25T09:06:47.04...	https://advocate.jbu.edu...
<input type="checkbox"/>	advocatejbuedu	0725bc8582188c01ee761b77a4af04d0643112fa8285d69280ab261c54c11f64	2024-03-25T09:06:47.19...	https://advocate.jbu.edu...
<input type="checkbox"/>	advocatejbuedu	0b021aacd4dc0bad3efea2142bf10f3be5463be5aa2e2e14c1a61b5edf2794fc	2024-03-25T09:06:45.33...	http://advocate.jbu.edu/...

Figure 22: corgiwebhashtable example

The user can decide on the initialization of each of the scraping instances which HASH_PARTITIONS wants to retrieve and manage in-memory to reduce the number of calls to the tables to know if an url was already visited. On the following example the user is specifying 4 partitions to initialize the Scraper with in memory storage of the urls hash list.

```

28 settings_manager.SCRAPER["HASH_PARTITIONS"] = [
29     "fbrefcom",
30     "wwweluniversalcommx",
31     "abcnewsgocom",
32     "wwwcnncom"]
33 > scraper_dict = scraper_dict = { ...
38 }
39 scraper = Scraper(cloud_integration = cloud_integration, settings_manager=settings_manager, scraper_dict=scraper_dict )
40 scraper.initialize()
41 scraper.start()

```

Figure 23: Code of simple Scraper using HASH_PARTITIONS

5.6 Availability to integrate customized scraping templates.

For Web crawling and Web scraping, users need the possibility to use their own customized templated for scraping, and in some cases use their preferred library to manipulate the html retrieved files.

On the following example [user/customTemplates/demo_scraper_templates.py](#) the custom user is making a scraper targeting 4 domains. User is providing values in ONLY_DOMAINS to determine from which containers wants to read html files. HASH_PARTITIONS to use an in-memory cache of visited urls and identify visited urls in memory, and scraper_dict which maps each domain with a customized class that will determine the different rules for scraping each of the websites.

```
# Scrape
settings_manager.SCRAPER["ONLY_DOMAINS"] = [
    "fbrefcom",
    "wwweluniversalcommx",
    "abcnewsgocom",
    "wwwcnncom"]
settings_manager.SCRAPER["HASH_PARTITIONS"] = [
    "fbrefcom",
    "wwweluniversalcommx",
    "abcnewsgocom",
    "wwwcnncom"]
scraper_dict = scraper_dict = {
    "fbrefcom": fbrefcom,
    "wwweluniversalcommx": wwweluniversalcommx,
    "abcnewsgocom": abcnewsgocom,
    "wwwcnncom": wwwcnncom
}
scraper = Scraper(cloud_integration = cloud_integration, settings_manager=settings_manager, scraper_dict=scraper_dict )
scraper.initialize()
scraper.start()
```

Figure 24: Sharding of database example

On the following file there is a customized example of one of the customized scrapings class:

[user/customTemplates/scraping_templates/abcnewsgocom.py](#)

Where the user uses the ScrapingTemplate as base, and then provides customized rules on how this website data is retrieved, all the data saved on .extra_keys will automatically be stored on the result container in the JSON file created from the scraper.

```
Code Blame 43 lines (30 loc) • 1.96 KB

1 from urllib.parse import unquote
2
3 from bs4 import BeautifulSoup
4 from parsel import Selector
5
6 from corgibrowser.corgi_web scraping.default_scrape_template import ScrapingTemplate
7
8
9 class abcnewsgocom(ScrapingTemplate):
10     def initialize(self, ):
11         self.soup = BeautifulSoup( self.html_text, "lxml" )
12
13     def extra_data(self, ):
14         self.sel = Selector( text = self.html_text )
15
16         if self.sel.xpath("//meta[@property='og:type' and @content='article']"):
17             self.handle_article()
18         else:
19             self.extra_keys[ "ContainersSuffix" ] = "unknown2"
20             self.extra_keys[ "html_text" ] = self.html_text
21
22
23     def handle_homepage(self, ):
24         self.extra_keys[ "ContainersSuffix" ] = "homepage"
25         self.extra_keys[ "images" ] = ""
26         self.extra_keys[ "paragraphs" ] = ""
27         self.extra_keys[ "html_text" ] = ""
28
29     def handle_article(self, ):
30         self.extra_keys[ "ContainersSuffix" ] = "article"
31
32         content_sel = self.sel.xpath("//div[contains(concat(' ', normalize-space(@class), ' '), ' FIIT_Article_main__body')]")
33         self.extra_keys[ "h1" ] = self.get_image_urls_by_xpath( self.sel, "//meta[@property='og:title']/@content" )
34         self.extra_keys[ "images" ] = self.get_image_urls_by_xpath( content_sel, "//img/@src" )
35         self.extra_keys[ "category" ] = self.extract_segment_in_path( unquote(unquote(self.row_key)), 0 )
36
37         self.extra_keys[ "author" ] = self.get_image_urls_by_xpath( self.sel, "//meta[@name='author']/@content" )
38         self.extra_keys[ "author_date" ] = self.get_image_urls_by_xpath( self.sel, "//meta[@property='lastPublishedDate']/@content" )
39
40         self.extra_keys[ "paragraphs" ] = self.extract_all_text( content_sel, "//div[contains(concat(' ', normalize-space(@data-testid), ' '), ' prism-article-body')]")
41
42         self.extra_keys[ "SourceDataField" ] = self.extra_keys[ "h1" ] + " " + self.extra_keys[ "paragraphs" ]
43         self.extra_keys[ "SourceDataField" ] = self.extra_keys[ "SourceDataField" ][ : 2000 ]
```

Figure 25: Code of abcnewsgocom scraping template

```
container_suffix = "-" + "json" + "-" + blob_row["ContainerSuffix"])
```

Home

Storage accounts

[> crawlingframeworkwkus1 Containers](#)
[> abcnews-gocom-json-article](#)

abcnews-gocom-json-article

Container

Search

Upload

Change access level

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Microsoft Entra user account)

Location: abcnews-gocom-json-article

Search blobs by prefix (cas...)

☐ Show deleted blobs

Add filter

	Name
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g
<input type="checkbox"/>	http%253A%252F%252Fabcnews.g

http%253A%252F%252Fabcnews.go.com%252FAManda%252FClinton-slams-fbis-manner-ann...

Blob

Save

Discard

Download

Refresh

Delete

Overview

Versions

Snapshots

Edit

Generate SAS

The file "http%253A%252F%252Fabcnews.go.com%252FAManda%252FClinton-slams-fbis-manner-announcing-email-review-unprecedented%252Fstory%253Fid%253D43162416" may not render correctly as it contains an unrecognized extension.

94

"PageType": "article",

95

"Images": "https://s.abcnews.com/assets/dctci/images/hulu.svg",

96

"Category": "Amanda",

97

"author": "ABC News",

98

"author_date": "2016-10-30T21:47:19Z",

99

"SourceDataField": "Clinton Slams FBI's Manner of Announcing Email Review as 'Unprecedented' and 'Deeply Troubling' Daytona Beach -- Hillary Clinton at a rally in Florida on Saturday afternoon slammed the manner of the FBI director's announcement of a review of new emails as 'unprecedented' u2019 and u2018deeply troubling. u2019 u2019 'How, I'm sure that some of you may have heard about a letter that the FBI director sent out yesterday," she told the crowd of about 900. "Well, if you're like me, you probably have a few questions about it." "It's pretty strange to put something like that out with such little information right before an election," she continued, referring to Friday's announcement coming 11 days before Election Day. "In fact, it's not just strange, it's unprecedented and it is deeply troubling because voters deserve to get full and complete facts." Clinton also doubled down on her campaign's call for FBI Director James Comey to release more details on what is in the emails that are now under review. The Democratic former secretary of state then brought up her Republican opponent, Donald Trump, who is attacking Clinton over the FBI review. "He is doing his best to confuse, mislead and discourage the American people," Clinton said of Trump. The newly-found emails that are under review by the FBI were discovered in a separate federal investigation of former Congressman Anthony Weiner, whose wife, Huma Abedin, is a longtime aide to Clinton and vice chair of her campaign. The FBI found emails on at least one device used by both Weiner and Abedin. Abedin, who was on the campaign trail with Clinton when the news broke Friday, did not travel with her Saturday. Aides have not said whether this is related to the investigation. Trump pondered publicly at a campaign event Saturday whether the Clinton campaign would "keep Huma," saying she is a "problem." The Clinton camp, though, said Abedin has their full support. "We of course stand behind her," Clinton campaign chairman John Podesta told reporters on a conference call earlier in the afternoon. The FBI review of the newly-found emails could feed into the concerns of some voters who express doubt about the former secretary of state's trustworthiness. Clinton, however, told reporters on Friday that she's not too concerned about it affecting the election. "You know I think people a long time ago made up their minds about the emails," she said. Some of her supporters at a canvassing kickoff event on Saturday seemed to agree. "I'm with Bernie Sanders," Jo Bouvier, 51, told ABC News. "Enough with the damn emails." This is a breaking news story. Please check back for updates."

Figure 26: Processed url data from abcnewsgocom

Example of [user/customTemplates/scraping_templates/wwwcnncom.py](#)

```
Code Blame 43 lines (31 loc) · 1.94 KB

1  from bs4 import BeautifulSoup
2  from parsel import Selector
3
4  from corgibrowser.corgi_webscraping.default_scrape_template import ScrapingTemplate
5
6
7  class wwwcnncom(ScrapingTemplate):
8      def initialize(self, ):
9          self.soup = BeautifulSoup( self.html_text, "lxml" )
10
11     def extra_data(self, ):
12         self.sel = Selector( text = self.html_text )
13
14         # if self.sel.xpath( "//meta[@property='og:type' and @content='article']" ):
15         #     self.handle_homepage()
16         if self.sel.xpath("//main[@class='article__main']" ):
17             self.handle_article()
18         else:
19             self.extra_keys[ "ContainerSuffix" ] = "unknown2"
20         self.extra_keys[ "html_text" ] = self.html_text
21
22
23     def handle_homepage(self, ):
24         self.extra_keys[ "ContainerSuffix" ] = "homepage"
25         self.extra_keys[ "images" ] = ""
26         self.extra_keys[ "paragraphs" ] = ""
27         self.extra_keys[ "html_text" ] = ""
28
29     def handle_article(self, ):
30         self.extra_keys[ "ContainerSuffix" ] = "article"
31
32         self.extra_keys[ "h1" ] = self.get_image_urls_by_xpath(self.sel, "//meta[@property='og:title']", "/@content")
33         self.extra_keys[ "images" ] = self.get_image_urls_by_xpath(self.sel, "//meta[@name='twitter:image']", "/@content")
34         self.extra_keys[ "category" ] = self.get_image_urls_by_xpath(self.sel, "//meta[@name='twitter:image']", "/@content")
35
36         self.extra_keys[ "author" ] = self.get_image_urls_by_xpath(self.sel, "//meta[@name='author']", "/@content")
37         self.extra_keys[ "author_date" ] = self.get_image_urls_by_xpath(self.sel, "//meta[@property='article:published_time']", "/@content")
38
39         article_sel = self.sel.xpath( "//div[@class='article__content']" )
40         self.extra_keys[ "paragraphs" ] = self.extract_all_text(article_sel, "//p")[: 2000 ]
41
42         self.extra_keys[ "SourceDataField" ] = self.extra_keys[ "h1" ] + " " + self.extra_keys[ "paragraphs" ]
43         self.extra_keys[ "SourceDataField" ] = self.extra_keys[ "SourceDataField" ][: 2000 ]
```

Figure 27: Code of wwwcnncom scraping template

Example of [user/customTemplates/scraping_templates/wwweluniversalcommx.py](#)

```
Code Blame 44 lines (33 loc) · 2.76 KB
1 from bs4 import BeautifulSoup
2 from parsel import Selector
3
4 from corgibrowser.corgi_web scraping.default_scrape_template import ScrapingTemplate
5
6
7 class wwweluniversalcommx(ScrapingTemplate):
8     def initialize(self, ):
9         self.soup = BeautifulSoup( self.html_text, "lxml" )
10
11     def extra_data(self, ):
12         self.sel = Selector( text = self.html_text )
13
14         if self.sel.xpath( "//meta[@name='mr:sections' and @content='homepage']" ):
15             self.handle_homepage()
16         if self.sel.xpath( "//body[contains(concat(' ', normalize-space(@class), ' '), ' homepage')]" ):
17             self.handle_homepage()
18         if self.sel.xpath( "//h1[contains(concat(' ', normalize-space(@class), ' '), ' home-custom-title')]" ):
19             self.handle_homepage()
20         if self.sel.xpath( "//meta[@property='og:type']/@content" ) and self.sel.xpath( "//div[contains(concat(' ', normalize-space(@class), ' '), ' encabezado ')]" ) and self.sel.xpath(
21             self.handle_article()
22
23     def handle_homepage(self, ):
24         self.extra_keys["ContainerSuffix"] = "homepage"
25         self.extra_keys["images"] = ""
26         self.extra_keys["paragraphs"] = ""
27         self.extra_keys["html_text"] = ""
28
29     def handle_article(self, ):
30         self.extra_keys["ContainerSuffix"] = "article"
31
32         headers_sel = self.sel.xpath( "//div[contains(concat(' ', normalize-space(@class), ' '), ' encabezado ')]" )
33         self.extra_keys["h1"] = self.get_text_by_xpath( headers_sel, "//h1[contains(@class, 'title') and contains(@class, 'font-bold')]" )
34         self.extra_keys["images"] = self.get_image_urls_by_xpath( headers_sel, "//picture[contains(@class, 'story_pic') and contains(@class, 'block') and contains(@class, 'w-full') and
35             self.extra_keys["category"] = self.get_text_by_xpath( headers_sel, "//a[contains(concat(' ', normalize-space(@class), ' '), 'sc__author--category')]" )
36
37         headers_sel = self.sel.xpath( "//div[contains(concat(' ', normalize-space(@class), ' '), ' sc__author ')]" )
38         self.extra_keys["author"] = self.extract_all_text( headers_sel, "//div[contains(concat(' ', normalize-space(@class), ' '), 'sc__author-nota')]" )
39         self.extra_keys["author_date"] = self.extract_all_text( headers_sel, "//span[contains(concat(' ', normalize-space(@class), ' '), 'sc__author--date')]" )
40
41         self.extra_keys["paragraphs"] = self.get_text_by_xpath( self.sel, "//div[contains(@class, 'sc') and contains(@class, 'pl-3')]/p[@itemprop='description']" )
42         self.extra_keys["html_text"] = ""
43
44         self.extra_keys["SourceDataField"] = self.extra_keys["h1"] + " " + self.extra_keys["paragraphs"]
```

Figure 28: Code of wwweluniversalcommx scraping template