# Crawley: A Tool for Web Platform Discovery

Daniil Dobriy[1,*], Axel Polleres[1]

[1]Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

**Abstract**

Crawley, a Python-based command-line tool, provides an automated mechanism for web platform discovery. Incorporating capabilities such as Search Engine crawling, web platform validation and recursive hyperlink traversal, it facilitates the systematic identification and validation of a variety of web platforms. The tool's effectiveness and versatility are demonstrated via two successful use cases: the identification of Semantic MediaWikis instances, as well as the discovery of Open Data Portals including OpenDataSoft, Socrata, and CKAN. These empirical results underscore Crawley's capacity to support web-based research. We further outline potential enhancements of the tool, thereby positioning Crawley as a valuable tool in the field of web platform discovery.

**Keywords**

Web Crawling, Search Engine Automatisation, Web Platform Discovery, Open Data Portals, MediaWiki

## 1. Introduction

The field of web platform discovery, which involves the systematic identification of websites, is a research priority for discovering Linked Open Data (LOD) [1] and accessing the factual extent of the Semantic Web. This subject intersects with web crawling, an automated process concerned with the traversal and extraction of web content, and Search Engine scraping.

Investigations in the field [2] have presented scalable algorithms for pattern mining, significantly enhancing the efficiency of media-type focused crawling. Additionally, efforts like MultiCrawler have proposed pipeline architectures for more effective crawling and indexing of Semantic Web data [3]. Other notable tools, such as Apache Any23[1], offer extraction libraries and web services that transform structured data from HTML and other web documents to more useful formats. The relevance of the application of such tools is illustrated by services like Portalwatch [4] and WikiApiary[2], which monitor the deployment and usage of specific Open Data and Wiki platforms on the web. Finally, due to the inherent cost of the platform and dataset discovery, services like LOD Laundromat [5] and LOD Cloud[3] exist to provide an entry point and catalogue linked datasets.

In the case of WikiApiary, the service provides a comprehensive repository, which tracks and catalogues Wikis and their respective metadata on the web. Most notably, WikiApiary

*ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference, November 6–10, 2023, Athens, Greece*

*Corresponding author.

daniil.dobriy@wu.ac.at (D. Dobriy); axel.polleres@wu.ac.at (A. Polleres)

0000-0001-5242-302X (D. Dobriy); 0000-0001-5670-1146 (A. Polleres)



[1]https://any23.apache.org/

[2]https://wikiapiary.com

[3]http://lod-cloud.net

also collects Semantic Wikis: Semantic MediaWiki (SMW), Wikibase and Cargo instances, - presenting an ample and under-researched facet of LOD. Despite its extensive coverage and reliance on bots ("bees") to keep the metadata up-to-date, the catalogue is manually curated through community submissions, which could potentially introduce gaps in data collection. Another specific case, Portalwatch - an open-source project that aims to collect and monitor Open Data Portals, including portal metadata - also shares the same limitation. This constraint underscores the need for automated discovery tools to ensure a more exhaustive enumeration and characterisation of web platforms such as Semantic Wikis or Open Data Portals. The proposed tool aims to enhance and ease web platform discovery in this area.

The structure for the remainder of the paper is as follows: Section 2 introduces the architecture and features of the tool and Section 3 provides an overview of two successful use cases of Crawley. Finally, Section 4 draws conclusions and discusses potential directions for future work.

## 2. Architecture and Features

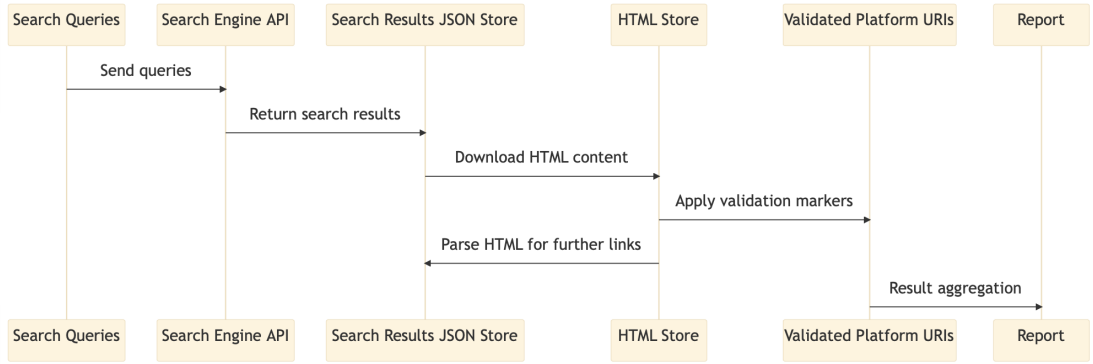


**Figure 1:** Crawley Architecture Diagram

Crawley is an open-source Python-engineered command-line tool designed to streamline the discovery and validation of specific technological platforms. It is currently available together with documentation on GitHub[4] under a CC-BY 4.0 license[5]. Figure 1 illustrates the high-level architecture of the tool. The tool extends various Search Engine APIs (SERP API, BING API) as a reliable solution to Search Engine querying. While the use of APIs subjects the tool to rate limits, the tool supports a multi-user approach.[6] Thus, the search is performed with Google, Bing, Yandex, Yahoo, DuckDuckGo, Baidu and Naver.

The user can initiate a search event, which is defined by a Search Engine (i.e., Google, Bing, Yandex, Yahoo, DuckDuckGo, Baidu, Naver) and the query itself.[7] The tool then queries the Search Engines, performing result pagination until all the query results are exhausted and prints

[4] http://purl.org/crawley
[5] http://creativecommons.org/licenses/by/4.0/
[6] cf. documentation for the tool on http://purl.org/crawley/readme
[7] cf. ibid

the actual number of unique sites, giving the user a heuristic estimation of how prodigious a certain query-Search Engine combination is, and aggregates the search results in the ./results folder. Although the queries can be formulated freely, we recommend using a subset of markers defined in the paragraph below that have a probability of being indexed by Search Engines (i.e., text snippets and image annotations, but not code excerpts). We observe a trade-off pattern whereby more general queries lead to more results, but fewer validation hits in the end, and more specific queries to fewer results, but a larger proportion of hits, which gives merit to formulating both general and specific queries.

The results/platform validation process with Crawley begins with the user identifying text/-code snippets commonly found on sites using a particular technology of interest: ”Powered by Semantic MediaWiki”, ”CKAN API”, ”Socrata API” as well as components of URL commonly used by a specific platform (e.g., *.../dataset*). We designate these as *markers*. Having identified possible markers and defined them in the configuration[8], the user can initiate a validation phase, whereby the tool requests HTML contents for the collected search results and then matches them against the markers, returning the total number of validation hits for each platform type and producing a validation report.

Finally, as a full-fledged crawler, the tool is able to recursively extract further links from validated sites. This is a useful feature which relies on the fact that similar platforms often contain hyperlinks to each other. The extracted links are then treated as search results in the pipeline and can be validated further, whereby previous HTML collection and validation events as well as results are cached for efficiency.

## 3. Use Cases

This section presents two successful use cases of Crawley, motivated by the need to discover and catalogue a broader range of Semantic Web data: Semantic Wikis and Open Data Portals.

### 3.1. Semantic Wiki Discovery

The first use case revolves around the discovery of Semantic Wikis, specifically Semantic MediaWikis, not captured by WikiApiary. To this end, a search (without recursive link collection) and validation have been performed with Crawley using the Bing Search Engine.

A set of custom markers has been identified in association with Semantic Wikis:

```
<meta name="generator" content="MediaWiki"
<link rel="ExportRDF"
Powered by MediaWiki
```

However, as noted before, only ”Powered by MediaWiki” was then used for the queries as other snippets are not indexed by Search Engines. Additional queries were therefore devised: ”MediaWiki”, ”Semantic MediaWiki”, and ”Semantic Wiki”.

Following this approach, 204 novel SMWs were discovered which were previously not catalogued by WikiApiary (which catalogues a total of 627 SMWs). The resulting catalogue has been

[8] cf. documentation for the tool on http://purl.org/crawley/readme

then used for the construction of a corpus of small and medium domain-specific Knowledge Graphs extracted from Semantic MediaWikis.

### 3.2. Open Data Portals Discovery

The second use case involves the identification of Open Data Portals. A search, a 2-step recursive link collection and validation have been performed with Crawley using all available Search Engines. Although the use case targeted multiple platforms (*CKAN*, *OpenDataSoft*, *Socrata*), we illustrate the defined markers specifically for *OpenDataSoft*:

```
BRAND_HOSTNAME: \"opendatasoft.com
ods.core.config
ods.minimal
ods.core.config
```

The queries used (as none of the markers is presumed to be indexed by SEs) were: "OpenDataSoft", "© OpenDataSoft" and ".opendatasoft.com". Following this approach, over 500 Open Data Portals, i.e., more than twice as many Open Data Portals could be identified as currently collected by the the original Portalwatch (256).

## 4. Conclusion and Future Work

In this work, we presented a novel tool for web platform discovery and illustrated its use in successful 2 real-life use cases. Thus, Crawley is both successful in standalone web platform discovery as it is for the extension of existing manually curated catalogues.

The promising areas for future work address the limitations of the current implementation of the tool and include 1) parallelizing requests to web resources instead of sequentially processing them, 2) implementing standalone Search Engine crawling and 3) enabling automatic marker discovery, which could greatly increase the efficiency of the discovery process, positioning Crawley as an increasingly valuable asset for comprehensive web platform discovery. We also plan to apply the tool to discover and monitor more Semantic Web resources, such as Wikibase instances and SPARQL endpoints.

## Acknowledgments

This work is part of a project funded by the *WU Anniversary Fund of the City of Vienna.*

## References

[1] C. Bizer, T. Heath, T. Berners-Lee, Linked data: The story so far, in: Semantic services, interoperability and web applications: emerging concepts, IGI global, 2011, pp. 205–227.

[2] J. Umbrich, M. Karnstedt, A. Harth, Fast and scalable pattern mining for media-type focused crawling, KDML (2009) 119.

[3] A. Harth, J. Umbrich, S. Decker, Multicrawler: A pipelined architecture for crawling and indexing semantic web data, in: The Semantic Web-ISWC 2006: 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006. Proceedings 5, Springer, 2006, pp. 258–271.

[4] J. Umbrich, S. Neumaier, A. Polleres, Towards assessing the quality evolution of open data portals, in: Proceedings of ODQ2015: Open Data Quality: from Theory to Practice Workshop, Munich, Germany, 2015.

[5] W. Beek, L. Rietveld, H. R. Bazoobandi, J. Wielemaker, S. Schlobach, Lod laundromat: a uniform way of publishing other people's dirty data, in: The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13, Springer, 2014, pp. 213–228.