

A Persistent Identifier Practice For A Research Data Repository

Cheng-Jen Lee (cjlee@iis.sinica.edu.tw)

Institute of Information Science

Academia Sinica

Taiwan

Tyng-Ruey Chuang (trc@iis.sinica.edu.tw)

Institute of Information Science | Research Center for Information Technology Innovation |
Research Center for Humanities and Social Sciences

Academia Sinica

Taiwan

Abstract

We present the design and the implementation of a persistent identifier (PID) system based on the Archival Resource Keys (ARK) scheme at the depositor, a research data repository built on top of CKAN. We will discuss practical issues when adopting ARK identifiers to a data repository: formatting rules of identifiers, the registry that mints and binds identifiers, the CRUD operations for identifiers, the metadata records of identifiers, and the name resolution service for identifiers.

Keywords

ARK, CKAN, PID

Proposal

A persistent identifier (PID) is a long-lasting reference to a document, file, web page, or other object [1]. In recent years, the Archival Resource Keys (ARK) [2] attracts attentions with its openness, unlimited identifiers, and non-paywalled model, especially for small research groups. As of the end of 2022, there are 8.2 billion ARK numbers assigned by over 1,000 organizations. In this presentation, we will first make a brief overview of the ARK feature [3] in the *depositor* [4], a research data repository built on top of CKAN [5]. On the depositor, all public datasets with basic descriptions (title of the dataset, temporal period that the dataset covers, and the creator of the dataset) will acquire an ARK identifier at the time of creation. The ARK identifier also comes with simple metadata, which is named Electronic Resource Citation (ERC), to describe itself. Take the ARK “ark:37281/k516v4d6w” for example, the URL: <https://n2t.net/ark:37281/k516v4d6w> will resolve to the dataset: https://data.depositor.io/dataset/se_rdm_guides. And, with the “?info” inflection appended: <https://n2t.net/ark:37281/k516v4d6w?info> will return an ERC record. In addition, the ARK identifier will be defunct while the dataset is being removed or private.

We will also discuss how to incorporate the ARK identifiers to the depositor. We have developed a `ckanext-ark` Python package [6] for CKAN, which is available on GitHub [7]. A tour of

the code repository is planned as follows. First, the noid-mint [8] Python package is used to mint up to 24,389,000 identifiers under a given formatting rule. Second, the CRUD (Create, Read, Update and Delete) operations for the ARK identifiers are designed to conform to the life cycle of datasets and metadata requirements for assigning ARKs. Third, a database table is created and acts as a registry to bind the internal id of the dataset and the ARK identifier with ERC records collected from the metadata of the dataset. Finally, a name resolution service consisting of Flask routing functions is defined to resolve an ARK URL and show its “?info” inflection in JSON format. Some future enhancements such as ARK for each version of a dataset and ARK for resources (i.e., data files) will be covered at the end of the presentation.

Technical Requirements

Wi-Fi connection

References

1. *Persistent identifier* - Wikipedia <https://en.wikipedia.org/wiki/Persistent_identifier>. ↩
2. *ARK Alliance — Home of the Archival Resource Key (ARK)* <<https://arks.org/>>. ↩
3. *ARK Persistent Identifier — User guide — depositar 6.5.7 documentation* <<https://docs.depositar.io/en/stable/user-guide.html#ark-persistent-identifier>>. ↩
4. *What is depositar?* <<https://data.depositar.io/about>>; documentation: <<https://docs.depositar.io/>>; code: <<https://github.com/depositar/>>. ↩
5. *CKAN – The open source data management system* <<https://ckan.org/>>; documentation: <<http://docs.ckan.org/>>; code: <<https://github.com/ckan/>>. ↩
6. *ckanext-ark · PyPI* <<https://pypi.org/project/ckanext-ark/>>. ↩
7. *depositar/ckanext-ark: A CKAN extension for assigning ARK identifiers to datasets* <<https://github.com/depositar/ckanext-ark/>>. ↩
8. *noid-mint · PyPI* <<https://pypi.org/project/noid-mint/>>. ↩