Enhancements for Wikidata extension

OpenRefine

proposal for GSoC 2020

Lu Liu
2w6f8c@gmail.com
https://github.com/afkbrb

Abstract

There are a few tasks to be finished for the Wikidata integration project of OpenRefine. Considering the duration of GSoC project, I would like to work on two of them:

1. OAuth support (#1612). Currently, the Wikidata extension uses password-based authentication to upload edits to Wikidata. Supporting OAuth will make it easier to host OpenRefine instances online for multiple users to share.

2. Generalization (#1640). The Wikidata extension was designed to work against Wikidata, which is an instance of Wikibase. It's possible to make it work against other Wikibase instances, too. Since there are now more and more Wikibase instances as listed at Wikibase Registry, it's meaningful to generalize the Wikidata extension.

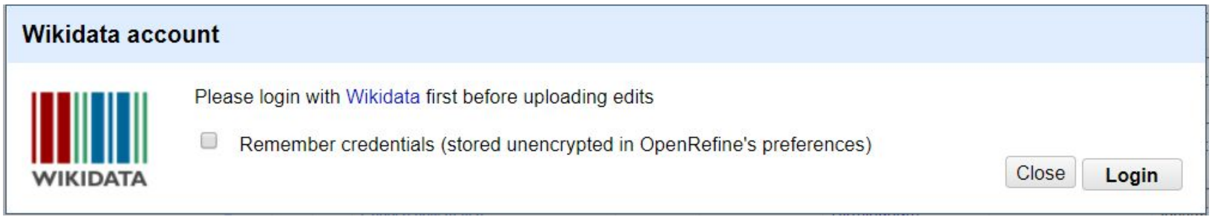
OAuth Support

Goals

The user will be able to configure consumer key / secret in *refine.ini* (I'll write a wiki to help the user to retrieve and configure the credentials) and use OAuth to authorize OpenRefine to upload edits to Wikidata.

Frontend

The original login dialog won't be changed. I'll create a new OAuth login dialog for logging in with Wikidata. The new dialog will show up only when the user has configured OAuth credentials in *refine.ini*. So the user can still use username / password to login by default.



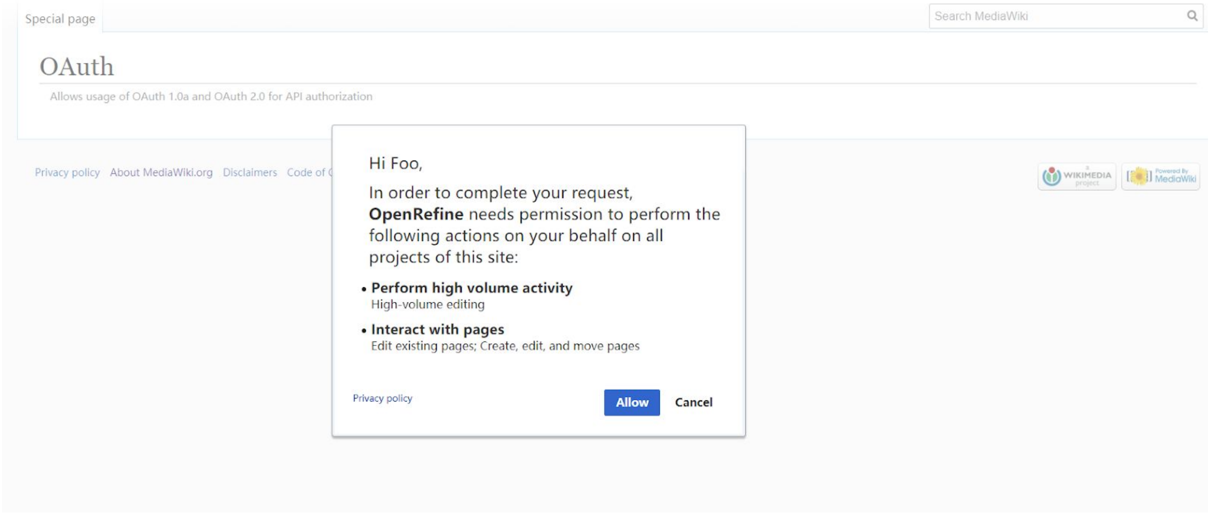
Wikidata account

Please login with Wikidata first before uploading edits

□ Remember credentials (stored unencrypted in OpenRefine's preferences)

pic 1: OAuth login dialog

Assume the user clicks on the "Login" button, he will be redirected to the following page to authorize OpenRefine:



OAuth

Allows usage of OAuth 1.0a and OAuth 2.0 for API authorization

Hi Foo,

In order to complete your request, **OpenRefine** needs permission to perform the following actions on your behalf on all projects of this site:

- **Perform high volume activity**
  High-volume editing
- **Interact with pages**
  Edit existing pages; Create, edit, and move pages

pic 2: authorization page

Suppose the user chooses "Allow", the backend will receive a verifier and use it to trade access token / secret. If the user checks "Remember credentials" in the OAuth login dialog, the credentials will be remembered by OpenRefine so that the user won't need to authorize OpenRefine again the next time.

Backend

As a prerequisite, I'll need to get Wikidata-Toolkit#411 merged first, so that we can use *OAuthApiConnection* from this library then.

To support configuring Wikidata OAuth credentials in *refine.ini*, I'll enable *refine* and *refine.bat* to read the credentials in *refine.ini* and pass them as JVM options. In #2392, I've done the same work for the gdata extension already, so it should be easy for me to do it here again.

I'll create an *AuthorizeCommand*, an *AuthorizedCommand* and update the original *ConnectionManager*. Assume the OAuth credentials have been configured, the OAuth workflow will be:

1. The user clicks on the "Login" button of the OAuth login dialog.
2. The browser sends a */command/wikidata/authorize* request to the backend.

3. *AuthorizeCommand* receives the request.
4. *ConnectionManager* fetches a request token and generates the authorization URL.
5. *AuthorizeCommand* responds to redirect the browser to the authorization URL.
6. The user grants OpenRefine with the permissions required.
7. The browser is redirected back to OpenRefine with a verifier.
8. *AuthorizedCommand* receives the verifier.
9. *ConnectionManager* uses the verifier to retrieve the access token / secret. The access token / secret will be used to sign requests to upload edits to Wikidata later. If the user chooses to remember the credentials, the access token / secret will be serialized to OpenRefine's preferences.

Generalization

Goals

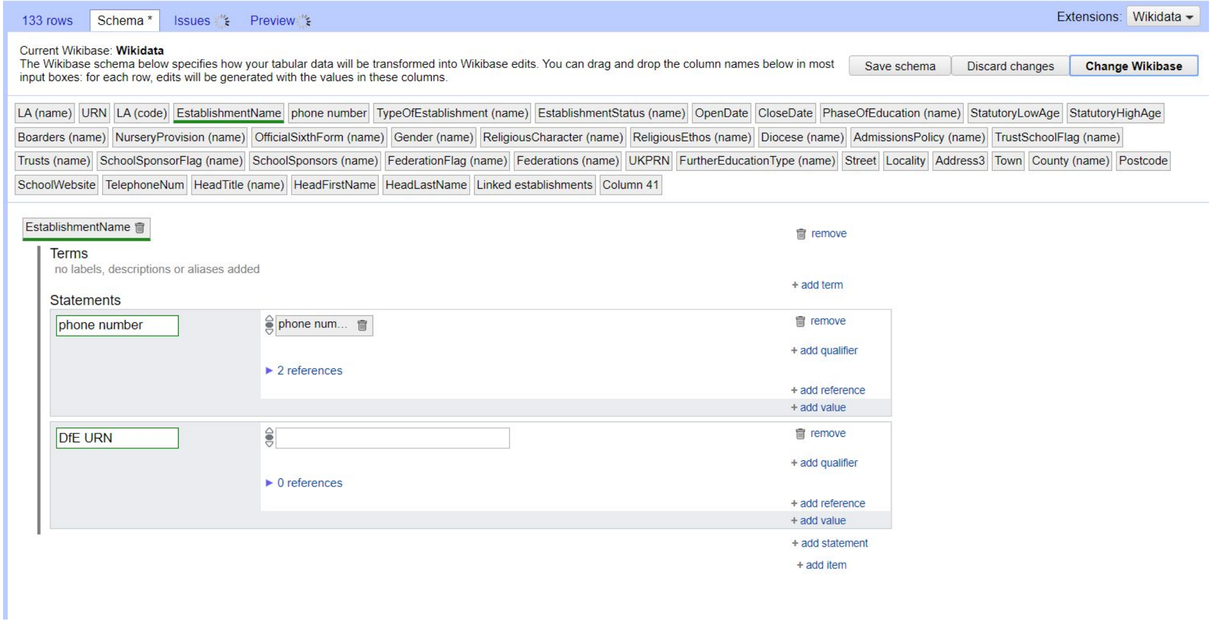
Transform the Wikidata extension to "Wikibase extension", enable it to work against arbitrary Wikibase instances. The user can add new Wikibase instance manifest URLs and specify which one to push the edits to, much like the addition and selection of the reconciliation services.

Roadmap

@wetneb has already provided a roadmap for this enhancement at #1640. The roadmap here will base on it and expand it.

This enhancement requires work on both the MediaWiki / Wikibase side and the Wikidata extension side. The work on the MediaWiki / Wikibase side has already begun but still needs a long time to be finished. One great outcome is the sample manifest for Wikibase by @despens. The work on the Wikidata extension side hasn't begun yet, so I'll begin and finish it for the GSoC project. The roadmap for me will be:

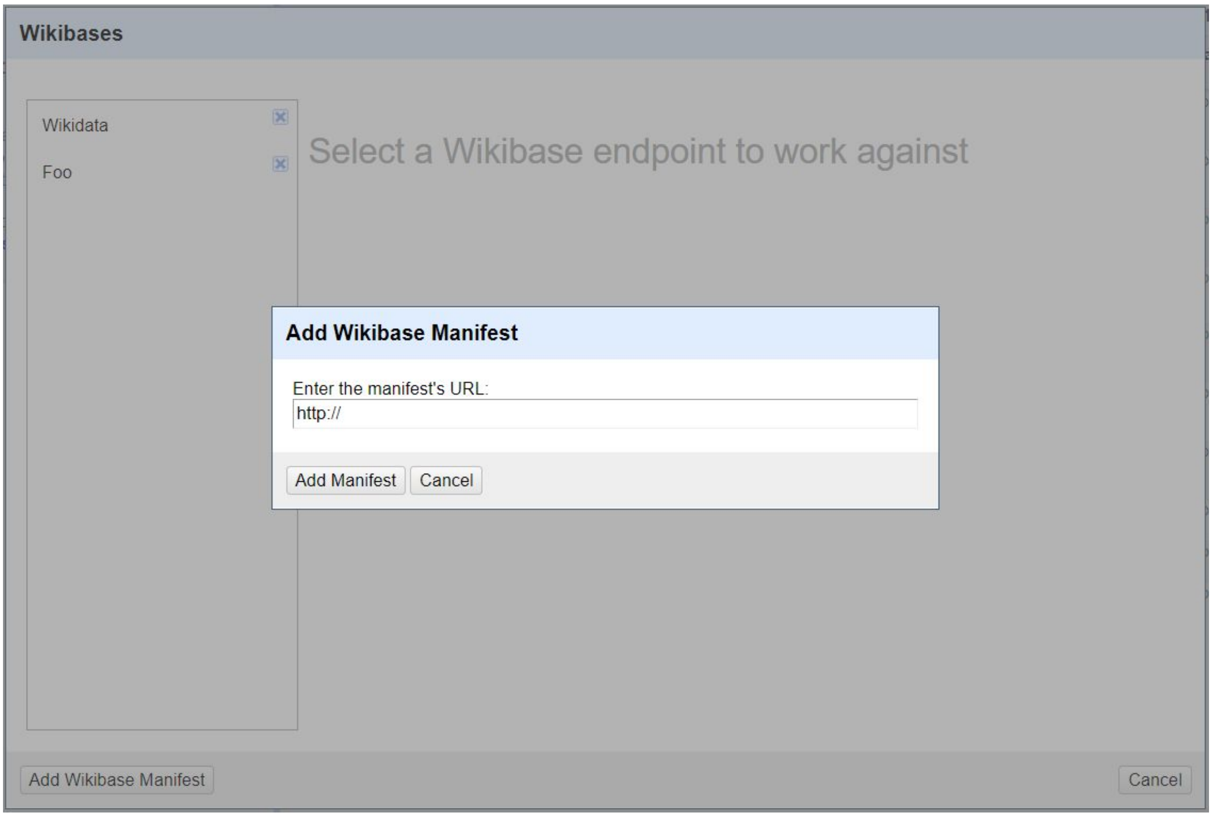
- Write a *WikibaseManifest* class. Given a Wikibase manifest URL, *WikibaseManifest* can retrieve the content of the manifest, parse it and set corresponding fields to represent it. The *WikibaseManifest* will contain necessary information to work against the Wikibase instance, such as MediaWiki API endpoint, reconciliation service API endpoint, etc. Just like the reconciliation services, the manifest will be serialized and stored at OpenRefine's preferences, so OpenRefine won't need to retrieve the manifest on the internet and parse it again the next time.
- Change the UI of the schema editor to enable the user to select which Wikibase instance to work against. Before uploading edits to a Wikibase instance, the user must create a schema first. So the schema editor is a proper place to offer the selection and addition functionality of Wikibase instances. The schema will use Wikidata as the default Wikibase instance.



Current Wikibase: **Wikidata**
The Wikibase schema below specifies how your tabular data will be transformed into Wikibase edits. You can drag and drop the column names below in most input boxes: for each row, edits will be generated with the values in these columns.

pic 3: schema editor

- Design the UI to list known Wikibase instances, the UI will provide an "Add" button to enable the user to add new Wikibase instances. The UI will look like:



pic 4: Add Wikibase Manifest

- Change schema serialization to include a key to indicate which Wikibase instance the schema is bound to. When deserializing, for backward compatibility, it's assumed that the Wikibase instance is Wikidata when the Wikibase information is not included in the json file.

- Update the code for schema evaluation, quality assurance, editing and especially OAuth. The main goal is to adjust the corresponding code to work for any Wikibase instance instead of being bound to Wikidata. Testing and debugging of this part could cost an amount of time.
- Change mentions of “Wikidata” to “Wikibase” in both source code and translations. This part will be relatively simple.

Deliverables

- OAuth 1.0a support for the Wikidata extension
- A wiki to help users to retrieve Wikidata consumer key / secret.
- Ability to configure the Wikidata extension to work against other Wikibase instances.
- Documents for the new changes of the Wikidata extension.
- Unit tests (I’ll follow TDD during development).

Benefits to Community

- Users will be able to use OAuth to login to Wikidata, so they don't need to worry about the security issues of their passwords then.
- The Wikidata extension can work on other Wikibase instances. This will hopefully enlarge OpenRefine’s user group.
- We'll make more progress on the Wikidata integration project.

Related Work

- @lucaswerkmeister added support for MediaWiki OAuth to the *scribejava* OAuth library with scribejava#852 . With his work, I won’t need to handle too much OAuth details myself for this project.
- @wetneb made a good start to add OAuth support to *Wikidata-Toolkit* with Wikidata-Toolkit#411. I’ll continue on his work to get it finished, as the OAuth functionality of *Wikidata-Toolkit* is needed for this project.
- I made OAuth credentials configurable in *refine.ini* for the gdata extension of OpenRefine itself with #2392. I can do the same for the Wikidata extension in this project.
- @wetneb provided a roadmap for the generalization part.
- @despens wrote a sample manifest for Wikibase. The goal of the generalization part is to enable the Wikidata extension to work against other Wikibases instances according to that manifest (I may change the manifest if needed).

Schedule

Date	Work
Prior - May 31	• Use OpenRefine more, try to discover the capabilities and limitations of OpenRefine from a user's view. • Get familiar with the community, learn about the evolution and future of OpenRefine. • Keep diving into OpenRefine's code by fixing issues. • Discuss with the mentor on more details of the plan. • Get Wikidata-Toolkit#411 merged, as it is the prerequisite of the OAuth support.
June 1 - 15	• Design new UI for OAuth login. • Change *refine* and *refine.bat* to support reading Wikidata OAuth credentials. • Create *AuthorizeCommand*, *AuthorizedCommand* and update *ConnectionManager* to handle OAuth workflow. • Add the functionality to serialize / deserialize OAuth credentials to / from OpenRefine’s preferences. • Integrate the frontend and backend to achieve the OAuth support. • Write a wiki to help users to retrieve OAuth consumer credentials.
June 16 - 30	• Study more about Reconciliation Service API, SPARQL, WDQS, Wikibase, QuickStatements, Quality Constraints, etc. These are important preparations to get the Generalization part done. • Figure out what’s redundant or missing in the sample manifest of Wikibase according to the study above. Update the manifest if needed.
July 1 - 31	• Create *WikibaseManifest* class to represent the manifest for Wikibase. • Design the new UI for both the schema editor and the Wikibase instances listing dialog. • Change the serialization and deserialization functionality of schema to include a mention to the Wikibase instance. • Update code for schema evaluation, quality assurance, edits uploading, etc. • Change mentions of “Wikidata” to “Wikibase”.
August 1 - 25	• Write user documentation and developer documentation. • Fix bugs and optimize the code. Try to make the code good enough to be merged. • Wrap up. Write summaries and blogs for all work done.

About Me

I'm a second year Computer Science and Technology undergraduate from University of Electronic Science and Technology of China. So the timezone for me is GMT+8. I'm a full stack developer, familiar with Java, Git, Maven, Vue.js, Linux, etc. I'm always admiring those who contribute to open source, and I would be happy to be one of them. So working on this project can be really helpful, and I'm hoping to become a regular contributor of OpenRefine.

So far (March 29), I've fixed the following issues of OpenRefine:

- #2320, Only first reference is exported for QuickStatements.
- #2103, New wikidata validator: description lengths and endings.
- #2380, Infinite preparing if missing slash after Google Spreadsheet URL.
- #2383, Configurable OAuth credentials for gdata extension.
- #2396, Misleading *JAVA_OPTIONS* configuration samples in *refine.ini*.
- #2398, Mock reconciliation calls in Wikitext importer tests.
- #2411, Mock Wikidata service for *PreviewWikibaseSchemaCommandTest*.
- #2461, GREL cross function should support lookup by numbers.
- #1950, Cross() function not working if applied to a column different from the column used for matching.
- More to come... View the merged PRs on GitHub

With these issues fixed, I'm quite familiar with OpenRefine now.

I'm looking forward to working on this project to contribute to OpenRefine more!