Information Regarding APIs

# MediaWiki

MediaWiki has a web-based API that I will use to retrieve information from the IMSLP site. Parameters are passed to the API as part of the request URL. For IMSLP, the URL is <http://imslp.org/api.php> .There are several possible values for the “action” parameter that represent broad categories of API functionality. I am primarily interested in searching for and retrieving data, so I will mostly be using the “query,” ”opensearch,” and “parse” actions.

* Query: Used for retrieving data from the wiki. The documentation for this particular action comprises a sizeable proportion of the auto-generated documentation. Any information I wish to retrieve from the wiki will be accessed through a query. This includes text, images, and metadata like which categories a page belongs to, page edit history, and many more.
* Opensearch: Returns pages with names matching the search string. The search is “smart” in the sense that it is not a raw text-based search – punctuation, diacritics, and “stopwords” or common words that do not provide useful results (I checked <https://en.wikipedia.org/wiki/Stop_words> to confirm the meaning)
* Parse: Parse “Wikitext” and return information in another format (HTML, XML, JSON). Wikitext is name of the markup language used for MediWiki sites (<https://www.mediawiki.org/wiki/Wikitext>)

The results of these actions are available in a variety of formats including XML and JSON. Many modern high-level programming languages have standard libraries to (de)serialize objects in these formats. There is a Python module called “mwclient” that serves as a Python wrapper for the API. I have not searched for libraries for other languages but there is a good chance they are freely available.  
Documentation: <http://imslp.org/api.php> , <https://www.mediawiki.org/wiki/API:Main_page>

# MusicBrainz API

MusicBrainz also has a web-based API that accepts HTTP GET requests and returns data in XML or JSON formats. For each of the 12 different “core entities” (artist, work, event, release, et cetera) in the MusicBrainz database, one can “search,” “lookup,” or “browse.”

* Search: Return information about the core entities except for URL. The search engine is built using Apache Lucene (<https://lucene.apache.org/>). Searches are performed on different indexes. These indexes correspond to core entities in the MusicBrainz database and contain different search fields.
* Lookup: Return the entity associated with a particular MBID (MusicBrianz ID, a unique identifier assigned to all of the core entities in the MusicBrainz database). One must first get the MBID of an entity by searching.
* Browse: Return lookups of all entities directly to the given entity, where “directly related” means there exists a relation in the database.

The output from this API is available in XML or JSON, but the JSON version is considered to be in beta.   
Documentation: <https://musicbrainz.org/doc/Development/XML_Web_Service/Version_2>