

Introduction to APIs and JSON

<https://github.com/eightBitter/guestLecture-apis-2020>

Outline

- What's an API?
- REST APIs
- Characteristics of APIs
- Examples of APIs in libraries
- Thinking about ethics in APIs

What is an API?

What is an API?

- Application Programming Interface
- Building blocks for developing a computer program
- Web-based system, operating system, software library, etc
- Specifications for routines, data structures, variables, etc

Web APIs

When used in the context of web development, an API is typically a defined set of specifications, such as Hypertext Transfer Protocol (HTTP) request messages, along with a definition of the structure of response messages, which is usually in an Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format.

Wikipedia

REST APIs

- Representational State Transfer (REST)
- Based on the concept of exposing collections of elements (resources) to the user
- Responds with different representations of the same resource based on the user's request, such as HTML, JSON, and XML

The API

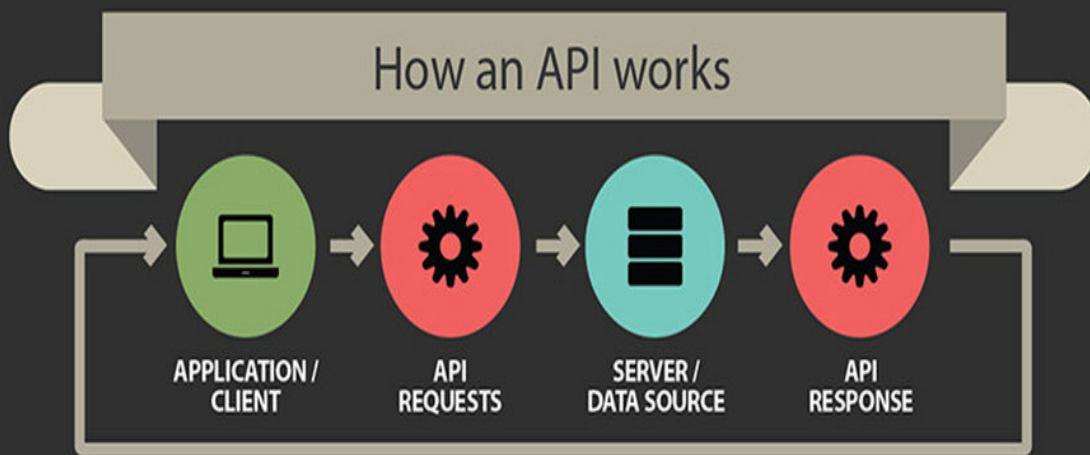
Application Programming Interface



API Definition

An application program interface that provides a developer with programmatic access to a proprietary software application. A software intermediary that makes it possible for application programs to interact with each other and share data.¹

How an API works



Retrieved from [What's an API and Why Do You Need One?](#)

Characteristics of APIs

Data formats

- Common formats are JSON and XML
- API data is representation of underlying system, but not a 1:1 representation
- Most APIs do not use metadata standards or schemas
- However, newer technologies like JSON-LD allow one to define interoperable JSON schemas

Data structures

- The structure of an API can come in many shapes and
- The data can be subsetting to multiple URLs based on
 - These usually follow a standard structure, such as
`https://example.org/resourceType/iter`
- Many APIs are searchable using a query and perimeter
 - `?filters=type mask, culture Haida`
 - `?
search_field=all_fields&q=programming`

Permissions

- An API can have read and/or write permissions
- Some APIs are open - accessible by anybody; while others require some kind of registration and authentication; some require a subscription

Examples of APIs in libraries

Examples of APIs in libraries

Reciprocal Research Network APIs

(<https://www.rrncommunity.org/api>)

- Purpose: "The RRN API is an interface that simplifies automated access to its publicly available digital collections records. This enables developers to make use of those records in new works and applications. While the API facilitates a new degree of public access to collections, it also represents an opportunity for originating communities to recontextualize their cultural heritage in new digital forms, reclaiming control over their representation of history and culture."
 - **Virtual Repatriation and the Application Programming Interface: From the Smithsonian Institution's MacFarlane Collection to "Inuvialuit Living History"**
- Permissions: open (some data requires an account?); read only
- Data format: XML and JSON

Examples of APIs in libraries

NCSU Libraries catalog API

([**https://catalog.lib.ncsu.edu/catalog**](https://catalog.lib.ncsu.edu/catalog))

- Purpose: exposes catalog resources from TRLN institutions for use in other applications
 - NCSU bento box search (QuickSearch ([**https://search.lib.ncsu.edu**](https://search.lib.ncsu.edu)))
- Permissions: open; read only
- Data format: JSON

Examples of APIs in libraries

Crossref API

(<https://www.crossref.org/services/metadata-delivery/rest-api/>)

- Purpose: "Our REST API exposes the metadata that members provide Crossref when they register their content with us. And it's not just the bibliographic metadata either. Funding data, license information, full-text links, ORCID iDs, abstracts, and Crossmark updates, are all available, if included in members' metadata."
 - Common tasks include text and data mining, auditing funder mandates, identifying author publications
- Permissions: open; read only
- Data format: JSON

Examples of APIs in libraries

ArchivesSpace API

(<https://archivesspace.github.io/archivesspace>)

- Purpose: used to edit metadata within an ArchivesSpace repository
- Permissions: authentication required; write
- Data format: JSON

Thinking about ethics in APIs

Thinking about ethics and APIs

APIs can provide democratic access to collections and data about those collections, and thus empower communities to self-represent and re-contextualize their cultural heritage. However, topics such as "virtual repatriation" continue to be thorny issues.

Thinking about ethics and APIs

APIs represent data within a system -> data within a system is created by humans or machines -> machines are created by humans -> humans are not neutral; therefore, APIs are not neutral.

Just like we need to lend an eye of scrutiny towards metadata in libraries, we need to do the same to those APIs that we use for application development and research.

Questions?

Thank you!

Jacob Shelby - jtshelby@ncsu.edu