The FamilySearch API in any Language

About the Speaker

Jimmy Zimmerman has been working in the field of genealogy software for over eight years. He currently manages the FamilySearch Partner Enablement Team. He has the privilege of working with the finest family history organizations around the world. This gives him a great view of things that are happening across the family history technology industry.

Contact Information

Email: jimmy@familysearch.org
Twitter: @jimmyzimmerman
Github: jimmyz
Blog: jimmyzimmerman.com

FamilySearch API in Any Language

The FamilySearch API is a RESTful API that can be accessed by virtually any programming language that has the ability to make network requests. To make integration easier, FamilySearch has made some software development kits (SDKs) available in a few programming languages. You may be using a language that is unsupported by the SDKs or you may choose not to use the SDKs.

This presentation will cover concepts you should understand, programming tools to use, best practices, and things to avoid when embarking on your journey to integrate an application with the FamilySearch API

Before You Begin: Things to Understand Really Well

Before you get started with code, it will be helpful to understand the following concepts.

FamilySearch Family Tree

The FamilySearch Family Tree is different than most other trees because it is a shared, global, collaborative tree. Make sure you spend plenty of time using the tree's features. You can play around with the family tree on the developer sandbox. You receive a sandbox user account when you register as a developer.

HTTP

The Hypertext Transfer Protocol (HTTP) is the foundation for all interaction with the FamilySearch API. It is essential that you understand how HTTP works. The FamilySearch API makes full use of HTTP response codes, headers, and methods.

- Introduction to HTTP
- HTTP Specification (RFC7231). Read Section 4, Section 5, Section 6.

OAuth2

OAuth2 is the standard used for providing access to the FamilySearch API. Before you can successfully make calls to the API, you will need an OAuth2 access token.

- FamilySearch Authentication Guide
- OAuth2 Spec (RFC6749). Read Section 4.1 (for web apps), Section 4.3 (mobile and desktop apps).

You can get an access token for quick development or request testing by visiting the /platform URL on any of the FamilySearch environments:

- https://sandbox.familysearch.org/platform
- https://beta.familysearch.org/platform
- https://familysearch.org/platform

FamilySearch Hypermedia Model

The FamilySearch API conforms to the HATEOAS constraint of REST, which basically means that resources, like persons, have links to other resources. Think of this like web pages linking to other web pages. This allows you to build robust clients that won't break if API URL structures change.

- Using Collections Resources to Navigate the FamilySearch API
- GEDCOM X RS is the REST service specification of GEDCOM X. It describes how resources link to each other.

Media Types

The FamilySearch API supports several media types. Think of this as a browser requesting text/html, image/gif, image/jpeg, etc. Each resource documents the type of content it will return, but you will need to specify the type by setting an "Accept" header to the appropriate type, or by appending .xml or .json to the end of the URL. The Accept header is the recommended approach.

For robustness, you can chain the media types that you request in a priority order:

XML:

```
Accept: application/x-fs-v1+xml, application/atom+xml, application/xml
```

JSON:

```
Accept: application/x-fs-v1+json, application/x-gedcomx-atom+json, application/json
```

GEDCOM X

The FamilySearch API uses GEDCOM X, a standard for representing genealogical data, as its media type. It has an XML and JSON representation. GEDCOM X is extensible, and the FamilySearch API takes advantage of this extensibility by adding new elements to the

documents. Therefore, the FamilySearch API uses the "`application/x-fs-v1+json`" and "`application/x-fs-v1+xml`" media types.

- GEDCOM X Specification
- FamilySearch Data Model: XML, JSON

Namespaces: if using XML

Warning: If you decide to use XML, you should understand the use of XML namespaces. Elements such as <gx:person> can change to <person> or <gedcomx:person> depending upon the definition of the namespaces at the root element of the document. These are all considered backwards compatible changes. Most XML parsers allow you to define namespaces by the URI that defines the namespace:

```
<gedcomx xmlns="http://gedcomx.org/v1/"
xmlns:fs="http://familysearch.org/v1/"
xmlns:atom="http://www.w3.org/2005/Atom" >
```

Essential Programming Tools

As you embark on your journey to integrate with the FamilySearch API, it is important to equip your toolbox with the following power tools.

HTTP Client Application

Several really great apps have been created to help you create sophisticated HTTP requests. This is really helpful when learning/exploring the API. Use these in conjunction with the /platform URLs listed above to get an access token.

- Dev HTTP Client (DHC): Chrome extension—Jimmy's preferred tool
- Advanced REST Client: Another great chrome extension
- Postman: A feature rich client

A Good HTTP Library

Most programming languages have really good HTTP libraries beyond the standard libraries offered by the language itself. Some more modern languages have really good HTTP libraries built in. Seek out a good library that offers the following features:

- Ability to log requests and responses
- Good URL handling that accepts absolute URLs, query string builders
- Setting of HTTP headers
- Configurable handling of redirects
- Good error handling to examine response codes

Programming Considerations

Here are a few things to consider as you code up your app.

Configuration Management

You may want to be able to configure from a central point the following:

- FamilySearch environment (production, sandbox, beta)
- Logging level (turn on and off API logging)
- Pending Modifications: allows you to test upcoming API changes before they become active

Error Handling

FamilySearch does its best to stay up, available, and stable. That said, on rare occasions, failures happen. Make sure your app has a way to gracefully degrade so that your users know what to do or what might be happening.

Best Practices

Here are some best practices to consider in your integration:

Use of HTTP Headers

Many things aspects of your API calls are controlled through HTTP headers. Make sure you are using Accept, Authorization, User-Agent, and X-FamilySearch-Feature-Tag headers.

FamilySearch Persistent Identifiers

Store and use Persistent Identifier URIs to persons. This provides a bookmark to these resources that you can drop in a browser or to make future API requests.

Things to Avoid

To avoid future pain, you may want to avoid doing the following things.

Hardcoding URLs

API URLs can change. Avoid potential breakage by using the Collections Resources for URL discovery.

Assumptions

Do not make the following assumptions:

- Data being present in all nodes of GEDCOM X documents. Always check for the null cases.
- 200 HTTP responses. Not all calls will result in a 20x level (success) response. Make sure you handle alternate http status codes.
- Clean data. Family tree data continues to improve over time, but you can run into strange situations such as looping pedigrees (I am my own grandpa), and persons with 1,000 parents, spouses, or children. Test your app with multiple users (on the Beta system) before releasing it to the masses. This can help you find strange data cases.

On to Success!

The FamilySearch Partner Enablement team is happy to help if you get stuck or run into issues with the API. Please send support requests to devsupport@familysearch.org.