

SESSION 4

Using FHIR in Custom Web Apps

INSTRUCTIONS

- READ THIS DOCUMENT
- STOP AND EXPLORE CONTENT AT THE EMBEDDED LINKS
- ANSWER THE QUESTIONS

TABLE OF CONTENTS

A. BACKGROUND	3
B. WHAT DO LOW-CODE OR NO-CODE WEB APP DEVELOPMENT PLATFORMS DO	4
C. WHAT IS THE DIFFERENCE BETWEEN FRONT END & BACK END OF A WEB APPLICATION	4
D. HOW DO YOU CREATE A BUBBLE.IO ACCOUNT	4
E. HOW DO YOU LOAD THE BUBBLE API CONNECTOR	5
F. HOW DO YOU CONFIGURE BUBBLE API CONNECTOR TO CONNECT TO A FHIR SERVER IN IOL	5
G. HOW DO YOU DISPLAY A PATIENT'S NAME IN A BROWSER	7
H. QUESTIONS	10

A. BACKGROUND

What is a vision of application-level interoperability?

Consider the role of the Google Play Store for Android smart-phone users or the role of the Apple App Store for Apple iOS smart-phone users. In both cases, an individual using either an Android or iOS device, typically a smart-phone, can access, download, deploy, and begin using new apps very quickly. On both platforms - Android and iOS - thanks to these app stores, consumers can utilize tens of thousands of apps of all kinds.



Figure 1. How can we do for EHRs and HIEs what App Stores already do for Smart-Phones?

How could this level of app interoperability be achieved for apps that connect to EHRs?

It turns out that large EHR vendors (e.g., Epic and Cerner) already provide app stores for their own EHRs. Other health informaticians have worked toward shareable apps that could interoperable with ANY EHR! One major project in this area is described next.

What are SMART-on-FHIR web apps? (This subsection is adapted from Braunstein, 2018*)

As we explained in Session 1 when FHIR was introduced, HL7 FHIR provides a set of “core” data models (called resources). It is true that many FHIR fields and value sets may NOT be sufficiently constrained to support specific app user requirements. To enable substitutable health apps as well as third-party application services, **SMART-on-FHIR** applies a set of “profiles” that provide developers with standard ways to use FHIR and predefined terminologies to represent medications, problems, labs, and other clinical data.

In the United States, SMART-on-FHIR has adopted a series of profiles. The five largest EHR vendors either have or will build SMART-on-FHIR into current or future releases of their EHR products. The vendors, the SMART-on-FHIR team, and HL7 are working together to standardize the SMART-on-FHIR API that supports SMART-on-FHIR apps in HL7 specifications.

Clearly, third party apps must not access protected health information without establishing trust in who is using those apps and respecting patient privacy choices. To facilitate this, SMART-on-FHIR provides login and data access authorization models based on the OpenID Connect and OAuth standards that are already widely used on the Internet and elsewhere.

To summarize, SMART-on-FHIR apps are apps intended to work seamlessly with any EHR platform after the platform has been upgraded to support the SMART-on-FHIR API and related standards. More information, with examples of actual SMART-on-FHIR apps can be found at these sites:

<https://smarthealthit.org/>

<https://apps.smarthealthit.org/>

* Braunstein ML. Health Informatics on FHIR: How HL7's New API is Transforming Healthcare. Springer International Publishing; 2018 Jul 26.

How are SMART-on-FHIR apps intended to be used to improve interoperability?

The SMART Health IT project for SMART-on-FHIR apps was launched with a New England Journal of Medicine article proposing a universal API (application programming interface) to transform EHRs into platforms for substitutable smart-phone-like apps. That article is here:

<https://www.nejm.org/doi/full/10.1056/NEJMp0900411>

(You may need to login to the University of Michigan libraries to access the paper.)

With federal investment, the SMART-on-FHIR API and examples of SMART-on-FHIR apps were developed as an open, free, and standards-based solution to improve app interoperability with EHRs.

App developers use SMART-on-FHIR to write an app once and have it run anywhere in the healthcare system. EHR vendors build support for SMART-on-FHIR into their products and platforms. Other technology companies do the same. For example, Apple relies on SMART-on-FHIR to give individuals access to their health records via iPhones. Hundreds of other app developers, including Google and Verily use SMART-on-FHIR technology as well. Microsoft and other cloud vendors launch health apps in their cloud services using SMART-on-FHIR. After successful lobbying, the 21st Century Cures Act and Final Rule from the Office of the National Coordinator for Health IT (ONC) made SMART-on-FHIR an EHR certification requirement.

With this background in mind, SMART-on-FHIR apps are intended to be used by anyone authorized to connect an app to an EHR in any organization. In this manner, the promise of SMART-on-FHIR is a high degree of app interoperability with EHRs and other Health IT systems.

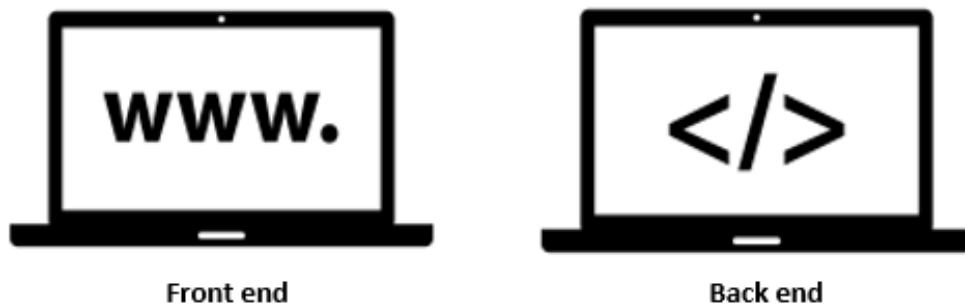
B. WHAT DO "LOW-CODE or "NO-CODE" WEB APP DEVELOPMENT PLATFORMS DO, e.g., Bubble.io?

Low-code (or no-code) development is one way to rapidly design enterprise-ready apps, build UIs, and integrate people, technologies, data, and systems.

Low code is a way for developers of all skill levels to design applications quickly by dragging, dropping, and configuring visual building blocks to create user views and whole applications.

Depending on user needs and low code platform capabilities, some custom software code may still need to be added by hand to complete a "low-code" web application.

C. WHAT IS THE DIFFERENCE BETWEEN THE "FRONT END" AND "BACK END" OF A WEB APPLICATION?



The front end of a web app is the user interface where the user navigates to access information whereas the back end of a web app is the server where the developers organize the data for the user.

D. HOW DO YOU CREATE A Bubble.io ACCOUNT?

To create a FREE Bubble.io account, follow these steps:

Step 1: Go to <https://bubble.io/>

Step 2: Enter your preferred email to "Get started for free"

Step 3: Enter your password and get started

Step 4: Accept the terms and conditions

Step 5: Complete the surveys

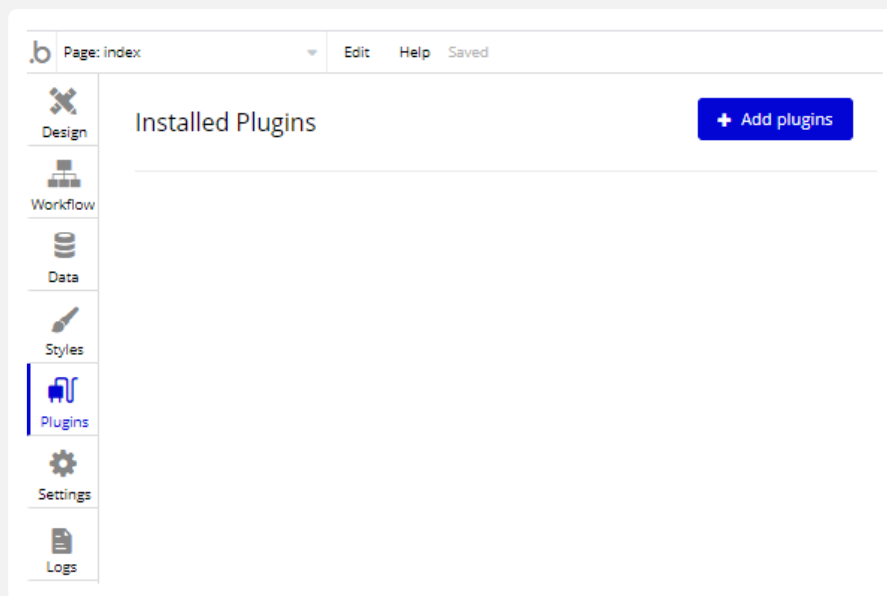
Step 6: Start a new web app

E. HOW DO YOU LOAD THE “Bubble API CONNECTOR” ?

To connect an API to the Bubble web app, follow these steps:

Step 1: Go to the taskbar and click on “plugins”

Step 2: Click on the +Add plugins button to view a list of plugins”



Step 3: Find and install the “Bubble API Connector” and click Done

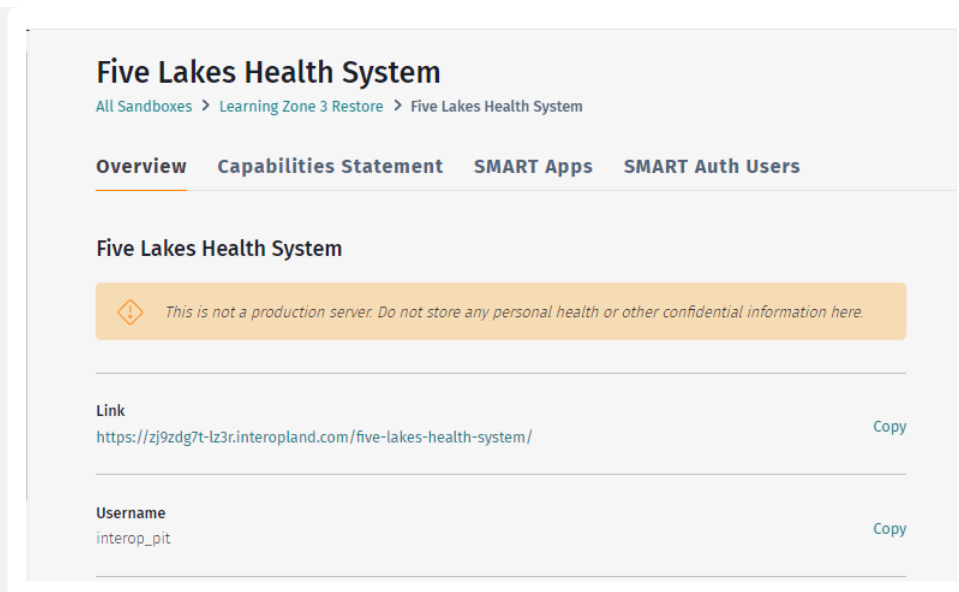
F. HOW DO YOU CONFIGURE THE “Bubble API Connector” TO CONNECT TO A FHIR SERVER IN IOL

To connect the Bubble API connector to a FHIR server in IOL, follow these steps:

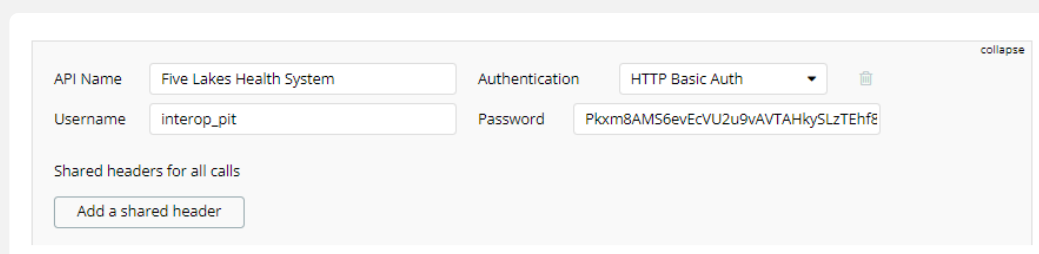
Step 1: To add an API click on **Add another API** under the Bubble API Connector

Step 2: Input an API name and select HTTP Basic Auth from the Authentication dropdown

Step 3: Login to IOL and open the Five Lakes Health System in Learning Zone 3 Restore to copy the Username and Password



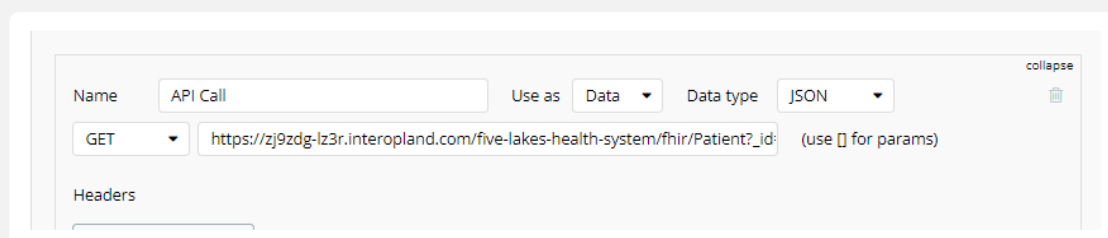
Step 4: Copy the username and password from IOL and paste on the API Connector as shown below



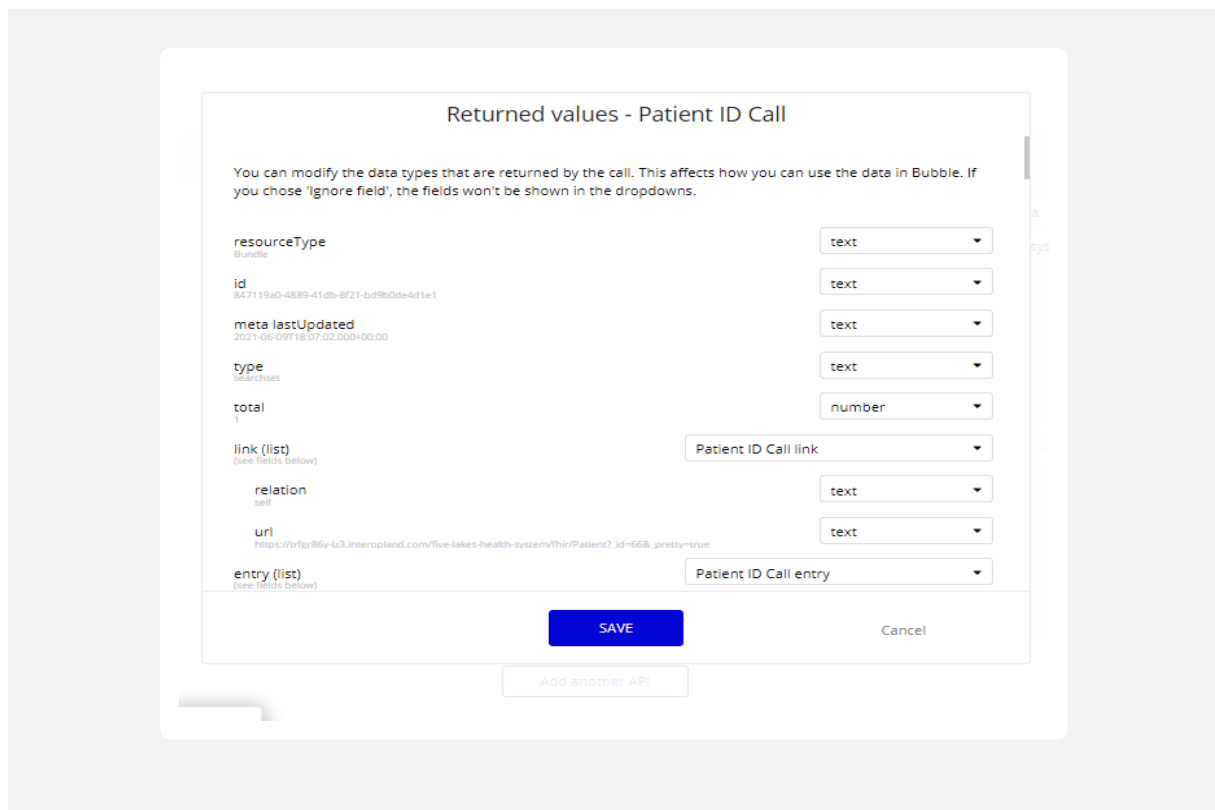
Step 5: To add an API Call expand the API Call section (Rename the call to a name that is easy to identify what the call does for example PatientID)
 For this example, we are going to use the Patient Resource to identify patient 66

Step 6: Paste the GET request below:

<https://zj9zdg-lz3r.interopland.com/five-lakes-health-system/fhir/Patient? id=66& pretty=true>



Step 7: Initialize the call to connect the API and Save

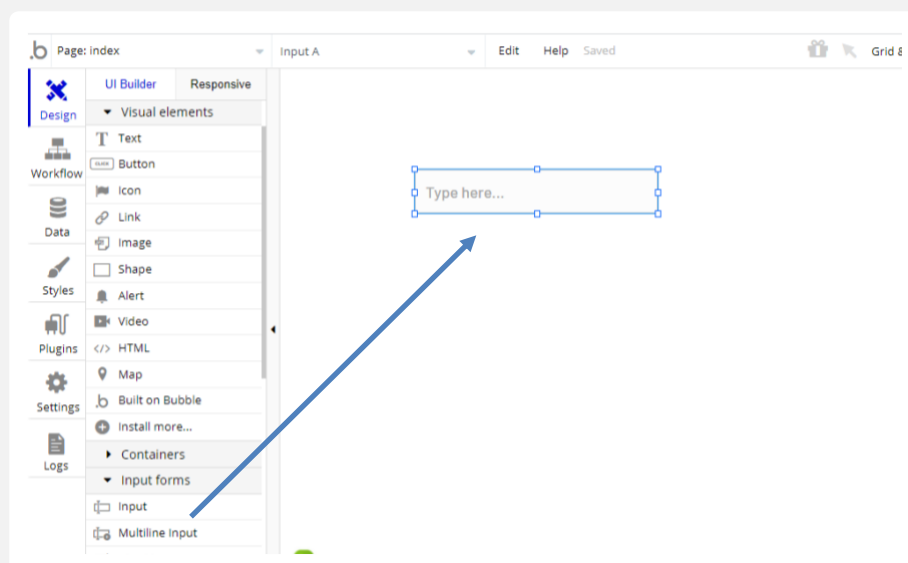


G. HOW DO YOU DISPLAY A PATIENT'S NAME IN A BROWSER?

To display a patient's name in a browser view , follow these steps:

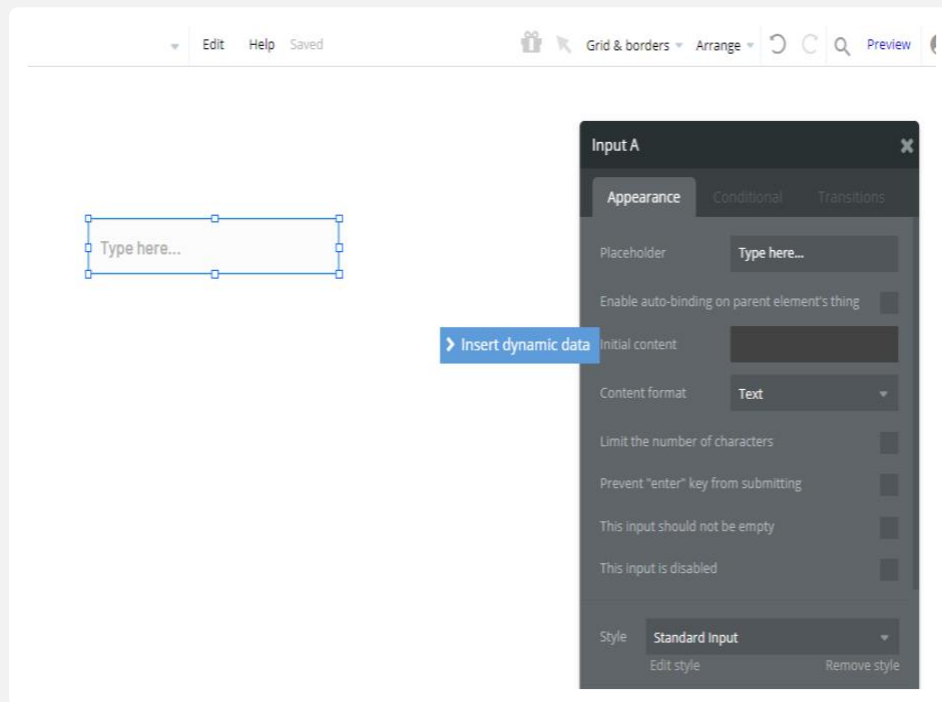
Step 1: Open the design tab

Step 2: Drag the "input form" and drop on the white space



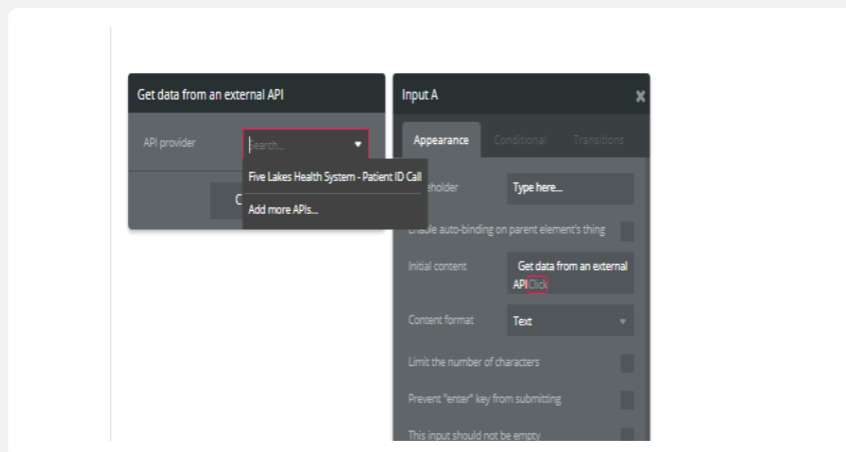
Step 3: To call data from the IOL FHIR server - Right click on the input form and select edit

Step 4: Go to the “Initial content” space and click on the “insert dynamic data” blue pop-up option

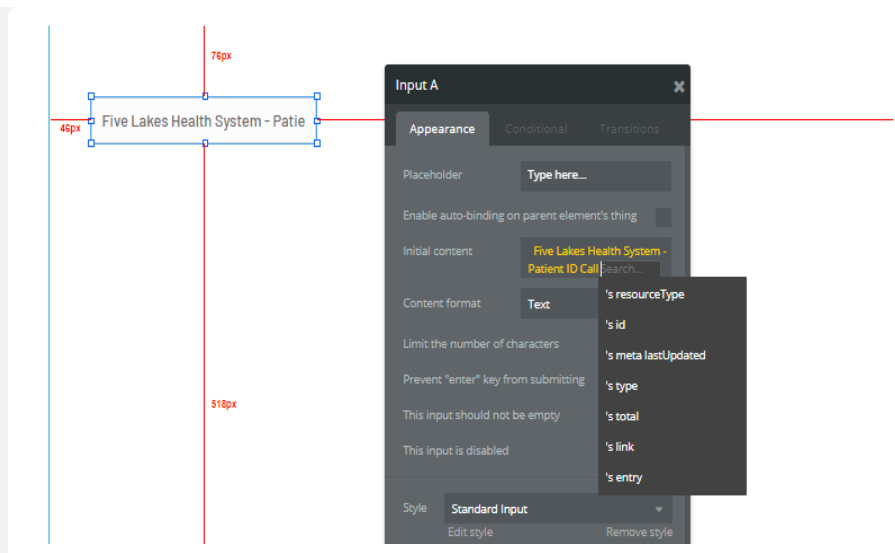


Step 5: From the drop-down list select “Get data from an external API”

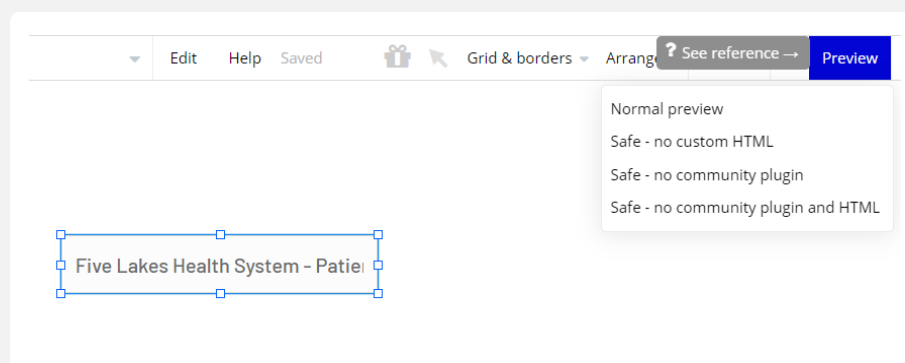
Step 6: On the “Get data from an external API” pop-up screen, select the “Five Lakes Health System-Patient ID Call” and close.



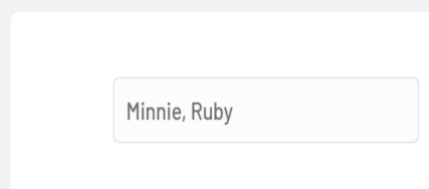
Step 7: From the dropdown lists select- ‘s entry, ‘s resource name, then ‘s given.



Step 8: Close form Input A and click on the Preview button to preview the Patient's name in a new tab.



Step 9: In the new tab you will see patient 66's name "Minnie, Ruby"



H. QUESTIONS TO TEST YOUR KNOWLEDGE

1. What is Patient 66's last name?

(Hint- perform steps in part G. above and select 's resources lastname)

Your answer here

2.

Your answer here