Health Gorilla Patient360 APIs

Overview:

This is a walkthrough of the entire workflow for Patient360 APIs, including patient lookup, patient creation, and document retrieval.

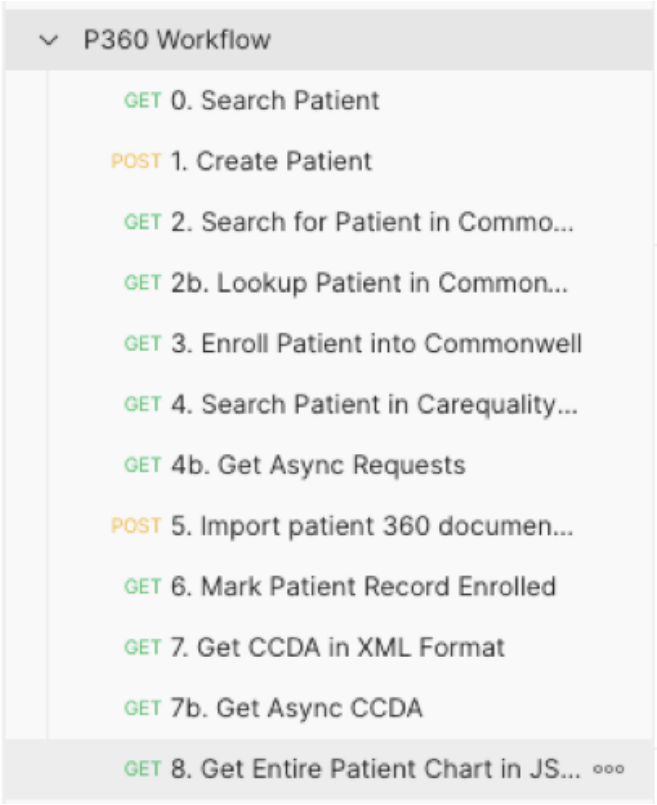
Requirements:

Before starting, you need the following to use the P360 Workflow Collection:

- Postman (or other HTTP Request tool)
- Health Gorilla Client ID
- Health Gorilla “secret” (used for creating OAuth Token)
- Health Gorilla assigned OAuth Token (Bearer Token)
- Health Gorilla Sandbox account to work in.

Getting Started:

Open the P360 Workflow Collection in Postman. If it is not already there, you’ll need to import it. It contains several Requests that we will use to call the Health Gorilla APIs.



Patient360 Workflow Collection in Postman

Health Gorilla API Requests

Search Patient

This Request can be used to search for the entire patient population, or search for a single patient if the ID is known.

Query Params:
None

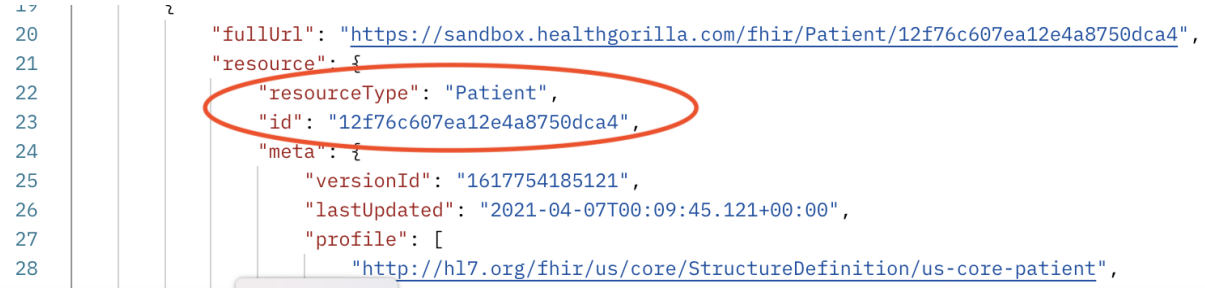
Headers:
Authorization = Bearer (your OAuth Token)
Content-Type = application/json

Body:
Blank - May use extended JSON logic to search.

Instructions:
In Headers, add in your bearer token and the content type. To search a specific patient, add the ID at the end of the URL. For the entire population, do not use an ID in the URL. Hit Send.

Results:

If the patient is found, the results should return the patient data. The Health Gorilla ID will be found in the bundle.



Health Gorilla ID is found in the bundle

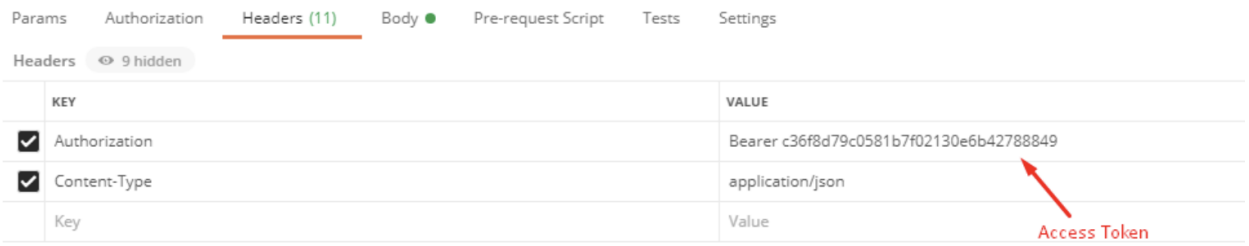
Create a Patient

Query Params:
None

Headers:
Authorization = Bearer (your OAuth Token)
Content-Type = application/json

Body:
Use the default body, but modify for your patient data parameters.

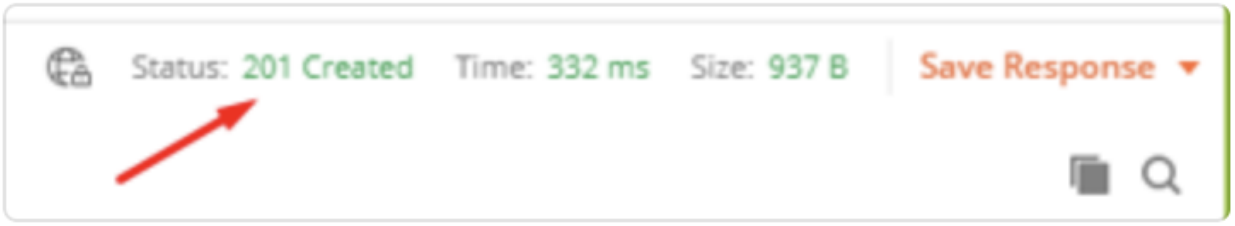
Instructions:
In Headers section, add in your bearer token and the Content-Type of application/json.
In Body, edit to include your patient information. Be sure to choose a unique MRN. Hit Send to send the request to Health Gorilla to create a patient. Note that if you attempt to create a patient with an MRN that already exists, you will get an error.
If you are successful, your resulting payload will contain the HG ID at the end of the Location_ID URL.



OAuth code in Header

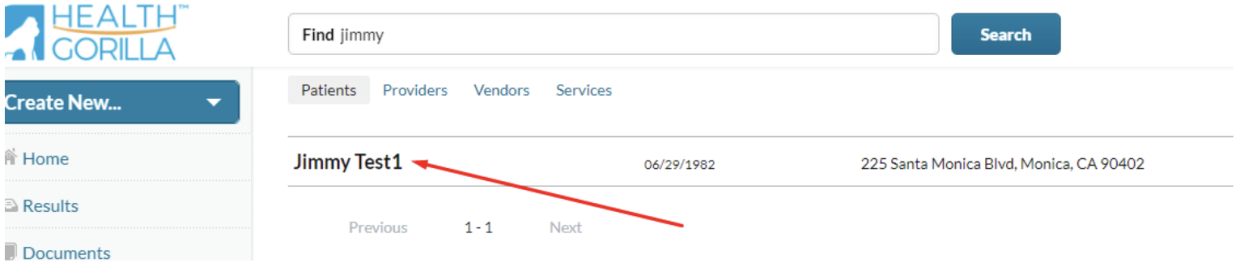
Unique MRN in Body

If successful, you will also see the Status of 201.



Status is found in the lower section.

To verify that the patient was created successfully, go to the Patient360 UI for your tenant and search for the patient.



In the UI, click the button to add the patient to Patient360.

Search for Patient in CommonWell

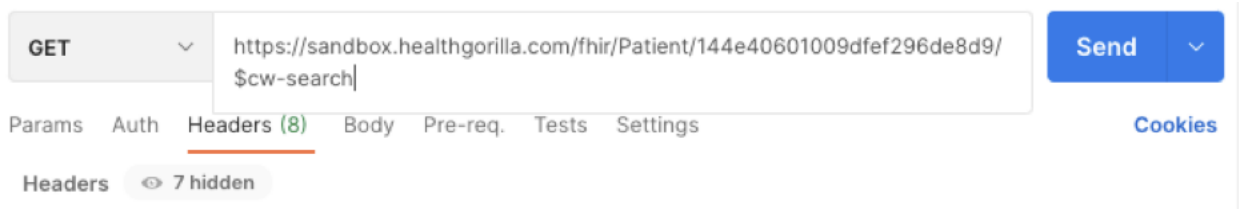
Query Params:
None

Headers:
Authorization = Bearer (your OAuth Token)

Body:
None

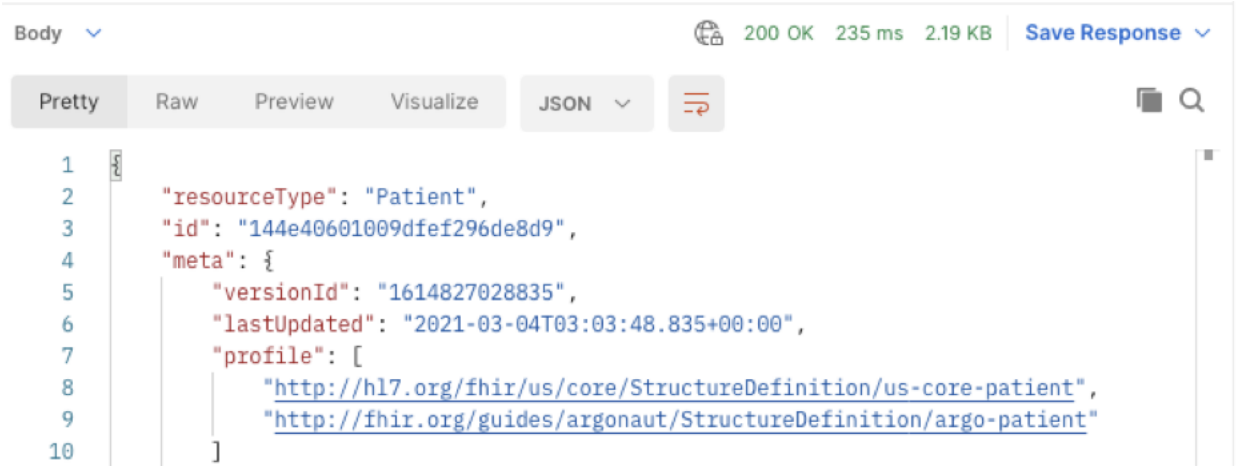
Instructions:

Add in your bearer token and the content type in the Headers section. Leave the Body blank. In the URL, put the Health Gorilla ID after “Patient”. Be sure to retain the last section that says “$cw-search”. Hit Send.



Results:

The body of the response will contain the patient data. If the patient does not exist, you will see “found=0” in the body.



Lookup Patient into CommonWell

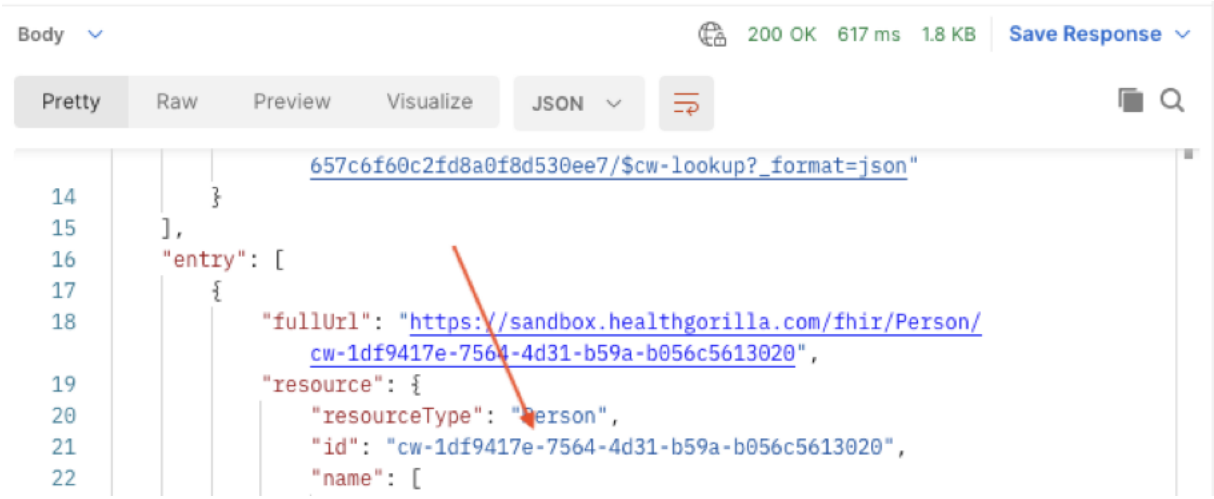
Query Params:
None

Headers:
Authorization = Bearer (your OAuth Token)

Body: None

Instructions:
Insert the HG ID into the URL after /Patient/ and hit Send.

Results:
JSON formatted text which includes the patient and the CW ID, which begins with “cw-“. Use this for enrolling the patient into CommonWell.



Enroll Patient into CommonWell

Query Params:
Person = CommonWell ID from previous step

Headers:
Authorization = Bearer (your OAuth Token)
Content-Type = application/json

Body:
None

Instructions:
Set the query value to the CommonWell ID from the previous step and use the HG_ID in the URL as shown below. Hit Send. This will enroll the patient in CommonWell.

Results:
If your enrollment was successful, results will show “true”.

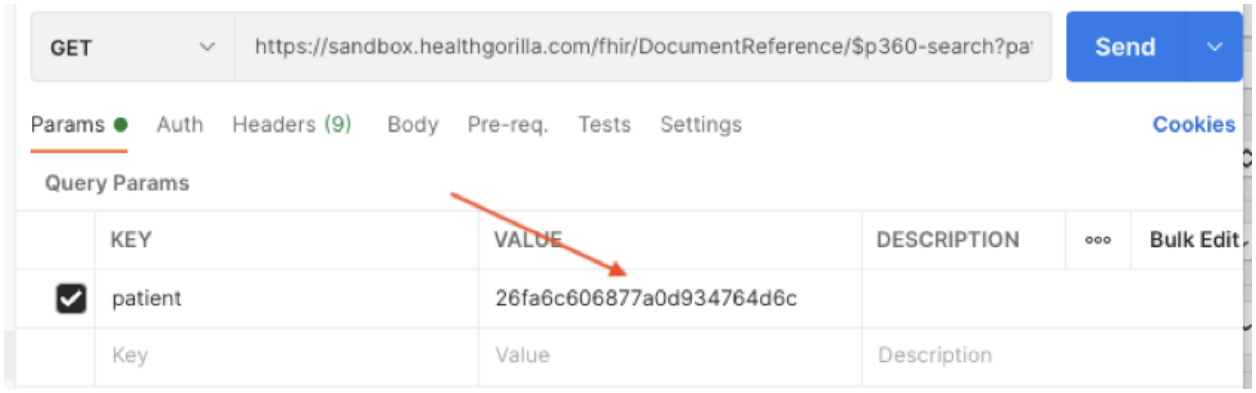
Search Patient in Carequality and CommonWell

Query Params:
Patient = HG ID

Headers:
Authorization = Bearer (your OAuth Token)
Prefer = respond-async

Body:
None

Instructions:
Enter the ID in the patient parameter for Query Params. Hit Send.



Results:
By running this with the “Prefer respond-async” header, you are asking for an asynchronous response. To find that async response, go to the lower section for the response and choose Headers. Get the Location, and that will be used in 4b. Get Async Requests.

Get Async Requests

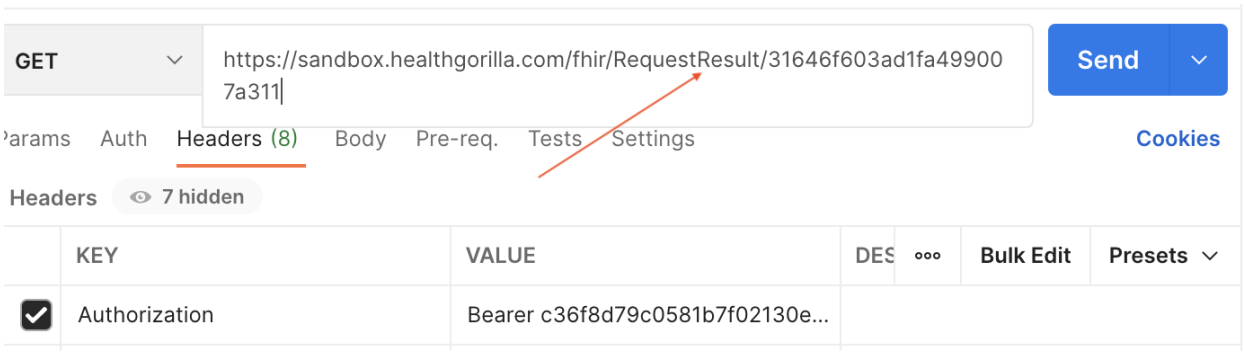
Query Params:
None

Headers:
Authorization = Bearer (your OAuth Token)

Body:
None

Instructions:

Take the URL from the previous Get request and paste it into the end of the URL (after www.healthgorilla.com) at the top of this to get the actual Async results.



Results:
The results of this Request will be used for import in the next Request. Grab the results from this Request and paste to a text editor. Then, grab everything from the line starting with “Entry”, all the way to the bottom. Use this in Step 5, Import Patient360 Documents.

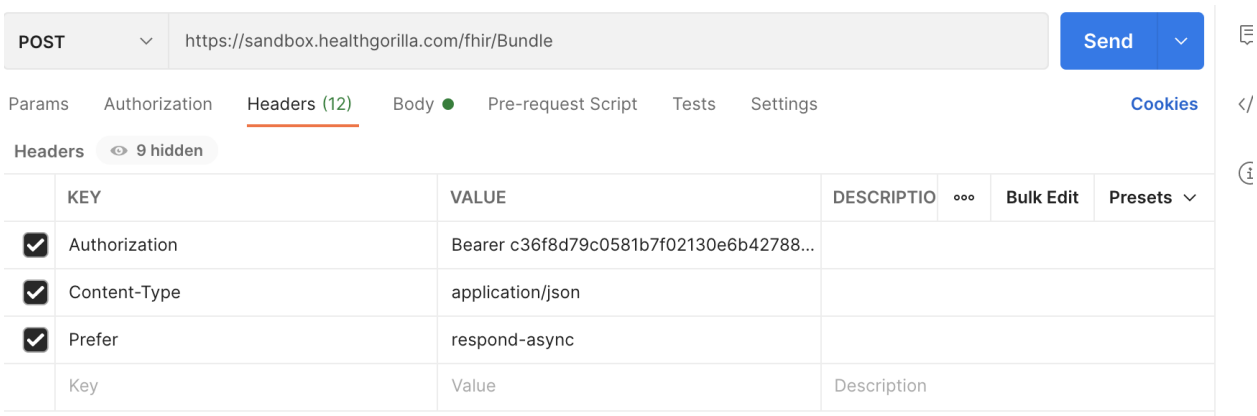
Import Patient 360 Documents

Query Params:
None

Headers:
Authorization = Bearer (your OAuth Token)
Content-Type = application/json
Prefer = respond-async

Body:
The Body will contain JSON data for the patient. This can come from the previous step if you are importing from Commonwell and Carequality, or can be data that you are importing and filing from your system. This data will be used to import the patient into Patient360. Be sure that the data is cut/pasted correctly. Malformed JSON will fail to import.

Instructions:
Update headers. Paste the import JSON data into the Body, replacing the lines that begin with “Entry” all the way to the bottom. Hit Send. This will file the data into Patient360.



Results:
Patients results in this JSON will be filed into Patient360, and can be verified in the Patient360 UI. Any results imported will have goldenrod highlight.

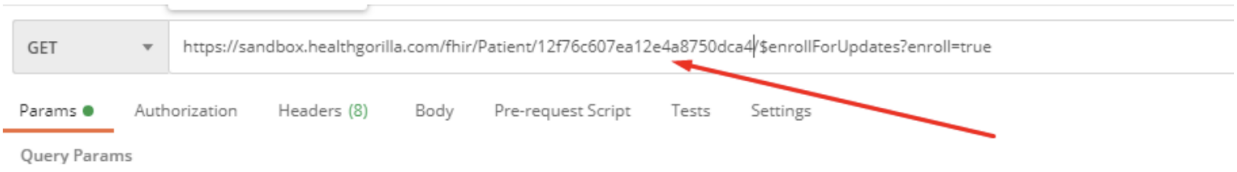
Mark Patient Record Enrolled

This step is used as
Query Params:
Enroll = true

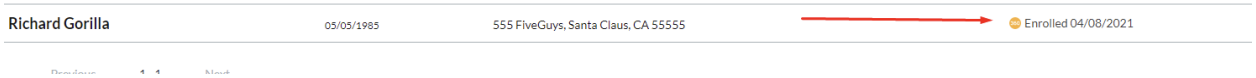
Headers:
Authorization = Bearer (your OAuth Token)

Body:
None

Instructions:
Enter the Query Param enroll=true and enter your OAuth Token. Replace the HG ID at the end of the URL with your patient’s HG ID. Hit Send to mark the patient as enrolled.



Results:
Validate that this patient is marked as Enrolled in the Patient360 UI.



Get CCDA in XML Format

Query Params:
None

Headers:
Authorization = Bearer (your OAuth Token)
Content-Type = application/xml
Prefer = respond-async

Auth:
Type = NoAuth

Body:
None

Instructions:
Headers, and change Type to NoAuth in Auth section. Enter the ID in the URL. Hit Send to get the CCDA.

Results:
By running this with the “Prefer respond-async” header, you are asking for an asynchronous response. To find that async response, go to the lower section for the response and choose Headers. Get the Location, and that will be used in 7b. Get Async CCDA.

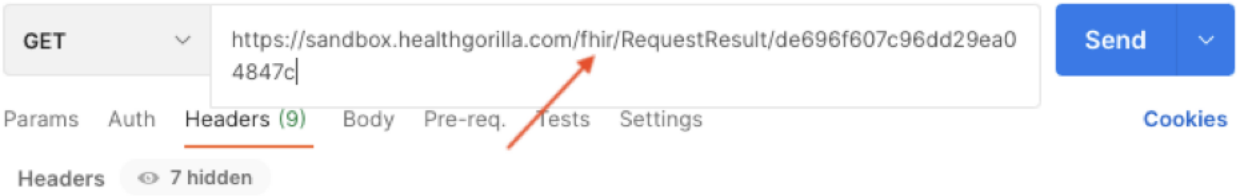
Get Async CCDA

Query Params:
None

Headers:
Authorization = Bearer (your OAuth Token)
Content-Type = application/xml

Body:
None

Instructions:
Enter Bearer Token in Headers. Take the URL from the previous Get request and paste it into the end of the URL at the top of this to get the actual Async results.



Results:

The results will appear as a CCDA in XML format. Note that this can be changed to JSON or other formats if necessary.

Get Entire Patient Chart in JSON

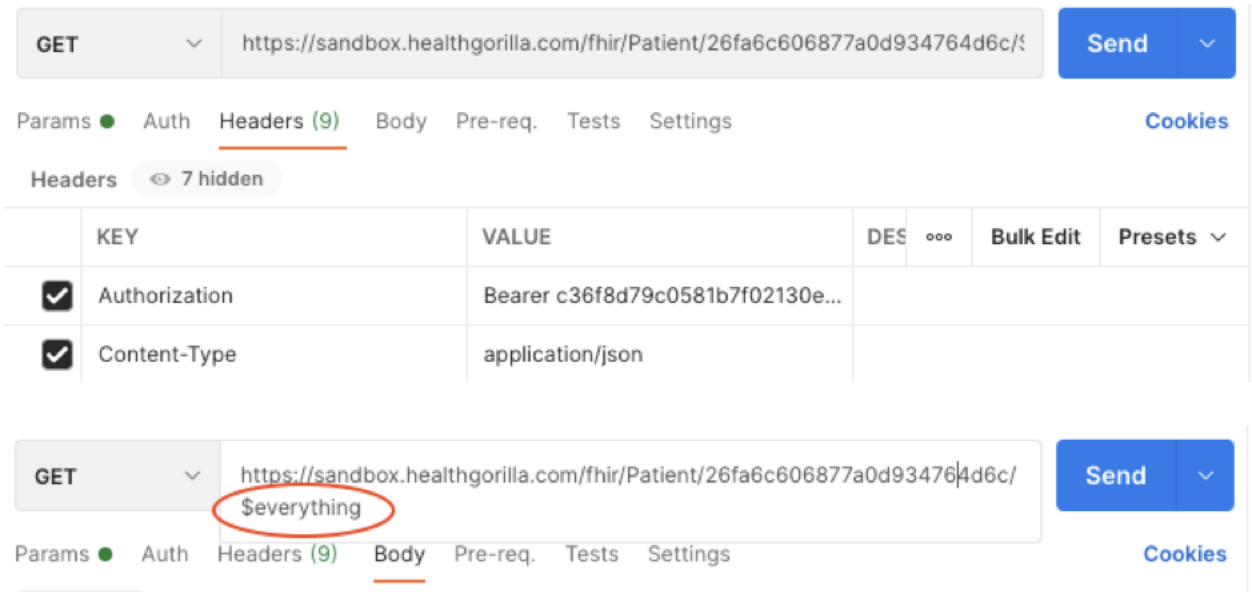
If the client prefers to have the entire patient record in a raw format instead of using the formatted version in the Patient360 UI, they can choose to export as JSON.

Query Params:
None

Headers:
Authorization = Bearer (your OAuth Token)
Content-Type = application/json

Body:
None

Instructions:
Fill in the Headers section, and enter the HG ID into the URL. At the end of the URL, add on /$everything to get all patient information.



Results:
The results will return a JSON formatted file containing all patient data. The format can also be changed to XML or other formats if necessary.


Body
200 OK
1579 ms
4.46 MB
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"resourceType": "Bundle",
"meta": {
"profile": [
"https://healthgorilla.com/fhir/StructureDefinition/hg-stateless-bundle"
]
},
"type": "searchset",
"total": 769,
"link": [
{
"relation": "self",