

## Advanced Search

Raman Dhanoa, Firely/Dogwood



HL7 FHIR DevDays 2023 | Hybrid Edition, Amsterdam | June 6–9, 2023 | @HL7 | @FirelyTeam | #fhirdevdays | [www.devdays.com](http://www.devdays.com)

ORGANIZED BY

**firely**

**HL7<sup>®</sup>**  
International

# Learning Objectives

- Define and differentiate between forward and reverse chaining.
- Understand the concept of bulk data access.
- Optimize Search Results
- Conduct Wildcard Searches
- Utilize More Complex Search Options
  - Example: `_filter` and `_query`
- Prerequisite knowledge: Knowledge of, and experience with, basic FHIR search.

# Basic Search (quick recap)

- Basic Search Syntax:
  - GET [base]/Patient?name=peter
- Prefixes:
  - eq (equal), ne (not equal), gt (greater than), lt (less than).
  - GET [base]/Observation?value-quantity=gt200
- Modifiers:
  - missing, exact, contains, text, in, not-in.
  - GET [base]/Patient?name:exact=Peter

# Agenda

- Chaining
  - `_include`, `_revinclude`
- Bulk Search
- More Complex Options
  - Wildcard searches (\*), `_filter`, `_query`



**Chaining, \_include, \_reinclude**



## Forward Chaining

- Resource type A (“WHAT”) has a query parameter of type ‘reference’ to resource type B
- All query parameters for resource type B can be accessed via ‘chaining’

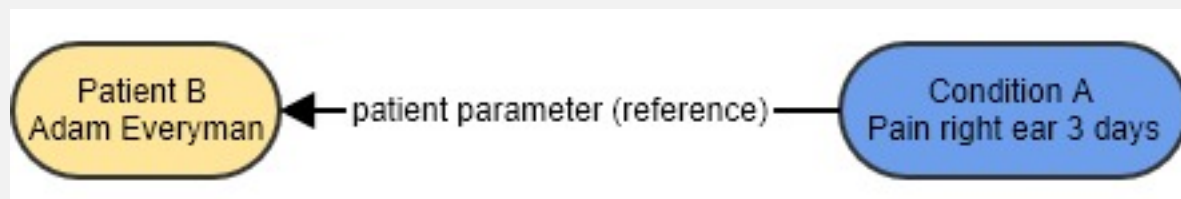


Query Condition resources (related to Ear Pain) AND (which reference patient Adam Everyman)

```
GET [base]/Condition?code=16001004&patient.identifier=1020304050
```

## Reverse Chaining

- Resource type A has a query parameter of type 'reference' to resource type B ("WHAT")
- All query parameters for resource type B can be accessed via 'reverse chaining'



Query for Patient resources who have had Ear Pain

```
GET [base]/Patient?_has:Condition:patient:code=16001004
```

## (Reverse) Includes

- Include *referenced* resources in the response bundle

Query **Condition resources (related to Ear Pain)** AND (**include referenced Patient**)

```
GET [base]/Condition?code=16001004&_include=Condition:patient
```

Query for **Patient resources (with name Adam)** AND (reverse include Conditions that refer to the Patient)

```
GET [base]/Patient?name=Adam&_revinclude=Condition:patient
```



# Demo: Forward Chain (with Include)

The screenshot displays the Insomnia API client interface. At the top, there is a menu bar with 'Application', 'Edit', 'View', 'Window', 'Tools', and 'Help'. Below the menu bar, the application title is 'Insomnia / New Document', and there are tabs for 'DESIGN', 'DEBUG', and 'TEST'. A 'Setup Git Sync' button is visible on the right. In the top right corner, there are 'Login' and 'Sign Up' buttons. The main interface is divided into a left sidebar and a main workspace. The sidebar contains a 'No Environment' dropdown, a 'Cookies' section, and a 'Filter' input field. The main workspace is currently empty, showing a context menu with the following options and keyboard shortcuts: 'New Request' (Ctrl + N), 'Switch Requests' (Ctrl + P), 'Edit Environments' (Ctrl + E), and 'New HTTP Request'. At the bottom left, there is a 'Preferences' button, and at the bottom right, it says 'Made with ❤️ by Kong'.

# Demo: Reverse Chain (w/ RevInclude)

Application Edit View Window Tools Help

Star 28,332

Insomnia / New Document ▾ DESIGN DEBUG TEST Setup Git Sync

Login Sign Up

No Environment ▾ Cookies

GET ▾ http://hapi.fhir.org/baseR4/Condition?code=16001004&patient.identifier=1 Send ▾ 200 OK 325 ms 29 KB 2 Minutes Ago ▾

Body ▾ Auth ▾ Query Headers Docs

Preview ▾ Headers 8 Cookies Timeline

```
1 [
2   "Condition",
3   "Condition",
4   "Condition",
5   "Condition",
6   "Condition",
7   "Condition",
8   "Condition",
9   "Patient",
10  "Patient",
11  "Patient",
12  "Patient",
13  "Patient",
14  "Patient",
15  "Patient"
16 ]
```

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia ↗

\$.entry[\*].resource.resourceType

Preferences

Made with ❤ by Kong



# Bulk Search



## Bulk Data Access

- \$everything (on Patient)
- \_type
  - GET [base]?\_type=Observation,Condition&patient=Patient/123
- \_since
  - GET [base]?\_since=2023-01-01T00:00:00Z
- \_typeFilter
  - GET [base]?\_typeFilter=Observation?status=final
- \$export
  - [base]/Patient/\$export?\_since=2022-01-01
  - [base]/\$export?\_type=Observation&\_since=2022-01-01

<https://build.fhir.org/ig/HL7/bulk-data/index.html>

# Bulk Access through Bundle

The FHIR Bundle resource can also be used to process a large number of requests at once. This is accomplished by making a single request to the server that contains a bundle of several individual requests.

```
{
  "resourceType": "Bundle",
  "id": "bundle-search",
  "type": "batch",
  "entry": [
    {
      "request": {
        "method": "GET",
        "url": "Patient?name=Smith"
      }
    },
    {
      "request": {
        "method": "GET",
        "url": "Observation?code=loinc:1234-5"
      }
    },
    {
      "request": {
        "method": "GET",
        "url": "Condition?subject:Patient=1"
      }
    }
  ]
}
```

# Demo: Bulk Access

The screenshot shows the Insomnia API client interface. At the top, there is a navigation bar with the Insomnia logo, a star count of 28,332, the current document name 'Insomnia / New Document', and tabs for 'DESIGN', 'DEBUG', and 'TEST'. There are also buttons for 'Login' and 'Sign Up'. On the left side, there is a sidebar with a home icon, a plus sign, and a search bar labeled 'Filt'. Below the search bar, there are two items: 'GET Reverse Chain' and 'GET Forward Chain'. The main area of the interface is dark and contains a menu with the following items:

- New Request (Ctrl + N)
- Switch Requests (Ctrl + P)
- Edit Environments (Ctrl + E)
- New HTTP Request



**More Complex Options...**



## Wildcard Searches

Allow clients to search for resources without knowing the exact values. The wildcard character in FHIR is `*`.

- String Parameters: For string parameters, a wildcard at the end of the string can be used to find resources that start with the given characters.
  - Example: `Patient?name=Sm*` will find Patients whose names start with "Sm".
- Token Parameters: For token parameters, a wildcard can be used to find resources that have any value for a given element.
  - Example: `Observation?code=http://loinc.org|*` will find Observations that have a LOINC code of any value.



# Controlling and Optimizing Search Results

- Parameters like `_summary`, `_elements`, `_count`, and `_page` can be used to modify the nature of the search results returned.
  - `_summary` - Patient?\_summary=true
  - `_elements` - Patient?\_elements=name,birthdate
  - `_count` - Observation?\_count=5
  - `_page` - Observation?\_count=5&\_page=2
  - `_sort` - Observation?\_sort=-date

## More Complex Options

- `_text`
  - GET /Patient?\_text=John
- `_content`
  - GET /Patient?\_content=John
- `_filter`
  - Patient: given eq "peter" and birthdate ge 2014-10-10
- `_query`
  - GET /Observation?\_query=recentObservations

# Q&A

ORGANIZED BY

