# Product Name – Minerva
# Document Name – FHIR R4 Search Guide
# FHIR Version – FHIR V4.0.1: R4
# Product Version – V4
# Document Version – 1.0

# Revision History

| Date | Document Version | Description |
|---|---|---|
| 23rd Sept 2020 | 1.0 | New Document |

# Table of Contents

# Search Parameter Types

These parameters are used for filtering resources based on actual resource content. The syntax for search varies as per the data types. These parameters are majorly grouped into date, string, token, reference, quantity, and uri types. The parameters that can be used for searching are defined for individual resources in Query Parameter section.

## Date/DateTime

A date search parameter performs the search on date/time or period.
The date parameter format:

- yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm] (the standard XML format).
- Passed in UTF-8 URL encoded format.

Date parameter is used with the following data types:

- Date
- DateTime
- Instant
- Period

**Examples (for Date, DateTime & Instant):**

| | |
|---|---|
| [parameter]=eq2020-01-10 | Matches all the time from 00:00 on 10-Jan 2020 to 23:59 on 10-Jan 2020 |
| [parameter]=eq2020-01-10T05:30:00Z | Matches only 2020-01-10T05:30:00Z |
| [parameter]=lt2020-01-14T10:00 | Matches with all date & time before 2020-01-14 10:00 AM |
| [parameter]=gt2020-01-14 | Matches with all date & time after 2020-01-14 00:00 AM |
| [parameter]=ge2020-01-14T10:00 | Matches with all date & time on or after 2020-01-14 10:00 AM |
| [parameter]=le2020-01-14 | Matches with all date & time on or before 2020-01-14 00:00 AM |
| [parameter]=ne2020-01-14 | Matches with all dates before 2020-01-14 00:00 and after 2020-01-14 23:59 |

**Examples (for Period):**

Implicitly, a missing lower boundary is "less than" any actual date. A missing upper boundary is "greater than" any actual date.

| | |
|---|---|
| [parameter]=2020-01-10 | Matches all Period where start date is before 10-Jan 2020 00:00 AM and end date is after 10-Jan 2020 00:00 AM.<br>Matches all Period where start date is before 10-Jan 2020 00:00 AM and end date is missing.<br>Matches all Period where start date is missing, and end date is after 10-Jan 2020 00:00 AM. |
| [parameter]=sa2020-01-10 | Matches all Period where start date is after 10-Jan 2020 00:00 am |
| [parameter]=eb2020-01-10 | Matches all Period where end date is before 11-Jan 2020 00:00 am |

**Note:** Partial dates with only Year / Year-Month part are not supported.

To search all the procedures that occurred over a 2-year period, the request code will be as follows:

```
GET [base]/Patient/23/Procedure?date=ge2010-01-01&date=le2011-12-31
```

## String

A String search parameter performs the search on a simple string. By default, a field matches a string query if the value of the field equals or starts with the supplied parameter value. The '**:exact**' modifier returns results that matches the entire supplied parameter.

**Examples:**

| | |
|---|---|
| [base]/Patient?given=eve | Patients with a name containing a given part with "eve" at the start of the name. |
| [base]/Patient?given:exact=Eve | Patients with a name with a given part that is exactly "Eve". |

To search all the patients with a name containing a given part with "eve" at the start of the name, the request code will be as follows:

```
GET [base]/Patient?given=eve
```

## Token

A token type is a parameter that provides an exact match search on a string of characters. It is mostly used against a code or identifier data type where the search is performed against the pair from a Coding or an Identifier.
Tokens are also used against other fields where exact matches is required like uri, boolean, and ContactPoint.

**Examples (for Identifier):**

To search all the patients with an identifier value 2345, the request code will be as follows:

```
GET [base]/Patient?identifier=2345
```

To search all the patients with an identifier value 2345 in the system http://acme.org/patient, the request code will be as follows:

```
GET [base]/Patient?identifier=http://acme.org/patient|2345
```

To search all the patients with an identifier value "446053", type code "MR", and type system "http://terminology.hl7.org/CodeSystem/v2-0203", the request code will be as follows:

```
GET [base]/Patient?identifier:of-type=http://terminology.hl7.org/CodeSystem/v2-0203|MR|446053
```

**Note**: Type system is useful where the system ID for the MRN is not known.

**Examples (for Code):**

To search any patient with a gender that has the code "male", the request code will be as follows:

```
GET [base]/Patient?gender=male
```

To search any patient with a gender that does not have the code "male", the request code will be as follows:

```
GET [base]/Patient?gender:not=male
```

**Examples (for CodeableConcept):**

To search any condition with a code "ha125", the request code will be as follows:

```
GET [base]/Condition?code=ha125
```

To search any condition with a code "ha125" in the code system "http://acme.org/conditions/codes", the request code will be as follows:

```
GET [base]/Condition?code=http://acme.org/conditions/codes|ha125
```

To search any condition with a code that has a text "headache", the request code will be as follows:

```
GET [base]/Condition?code:text=headache
```

**Examples (for Boolean):**
To search for active patients, the request code will be as follows:

```
GET [base]/Patient?active=true
```

## Reference

A reference parameter refers to reference between resources.

**[parameter]=[type]/[id]** - the logical [id] of a resource of a specified type using a local reference (i.e. a relative reference).

**Examples:**
To search for any observations where the subject refers to the patient resource with the logical identifier "23", the request code will be as follows:

```
GET [base]/Observation?subject=Patient/23
```

Elements of type canonical have a different syntax. They may contain a version specific reference.

**[parameter]=[canonical_url]** – matches only against the canonical URL.
**[parameter]=[canonical_url]|[version]** – matches against canonical URL & specific version.

**Examples:**
To search for all the observations having canonical URL "http://acme.com/some-profile" regardless of the version, the request code will be as follows:

```
GET [base]/Observation?definition=http://acme.com/some-profile
```

To search for all the observations having canonical URL "http://acme.com/some-profile" and version 3.0, the request code will be as follows:

```
GET [base]/Observation?definition=http://acme.com/some-profile|3.0
```

To search for all the observations having canonical URL "http://acme.com/some-profile" and version <= 2.0, the request code will be as follows:

```
GET [base]/Observation?definition:below=http://acme.com/some-profile|2.0
```

## Quantity

A quantity parameter searches on the Quantity data type. The syntax for the value follows the form:

**[parameter]=[number]||[system]||[unit]** - matches a quantity with the given unit.

**Examples:**
To search for all the observations with a value of 5.4 irrespective of the system and unit, the request code will be as follows:

```
GET [base]/Observation?value-quantity=5.4
```

To search for all the observations with a value of 5.4 where the unit is “mg”, the request code will be as follows:

```
GET [base]/Observation?value-quantity=5.4||mg
```

To search for all the observations with a value of 5.4 where the unit is “mg” in the system “http://unitsofmeasure.org”, the request code will be as follows:

```
GET [base]/Observation?value-quantity=5.4|http://unitsofmeasure.org|mg
```

### URI

The uri parameter refers to an element that contains a URI data type.

**Examples:**
To search for all the procedures with instantiatesUri having exact URL “http://example.org/protocol-for-hypertension-during-pregnancy”, the request code will be as follows:

```
GET [base]/Procedure?instantiates-uri=http://example.org/protocol-for-hypertension-during-pregnancy
```

To search for all the procedures with instantiatesUri having exact oid urn:oid:1.2.3.4.5, the request code will be as follows:

```
GET [base]/Procedure?instantiates-uri=urn:oid:1.2.3.4.5
```

## Parameters for all Resources

These parameters are generic and applicable on all resources. These parameters provide capability of filtering resources based on Meta properties instead of actual resource content. These parameters can be clubbed with resource specific search parameters as well to perform a search operation.

### _id

The search parameter _id refers to the logical id of the resource and can be used when the search context specifies a resource type.
**Syntax**: https://<base-url>/minerva/fhir/r4/<FHIR_Resource>?_id=<Logical ID>

```
GET [base]/Patient?_id=25
```

Above request searches for patient with _id 25. This is equivalent to ‘Patient/25’.

### _lastUpdated

The search parameter _lastUpdated can be used to select resources based on the last time they were changed.
The date parameter format is yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm] (the standard XML format). All prefixes (eq, gt, ge, lt, le, ne) mentioned in date search parameter type are also applicable here.

**Syntax**: https://<base-url>/minerva/fhir/r4/<FHIR_Resource>?_lastUpdated=<Date>

```
GET [base]/Patient?_lastUpdated=gt2010-10-01
```

Above request searches for all patients changed after 2010-10-01 00:00 AM.

## _profile

The search parameter _profile can be used to perform search on the equivalent elements in the meta element. This parameter restricts the search to only resources that are tagged as conforming to a profile.

**Syntax**: https://<base-url>/minerva/fhir/r4/<FHIR_Resource>?_profile=<Profile URL>

```
GET [base]/DiagnosticReport?_profile=http://hl7.org/fhir/StructureDefinition/lipid
```

Above request searches for all DiagnosticReport conforming to profile http://hl7.org/fhir/StructureDefinition/lipid.

# Search Result Parameters

These parameters are used to manage the resources being returned in the response of the search request. Response can be ordered & limit using _sort & _count parameters. These parameters are used to display data in a meaningful order & load on server can be minimized by limiting data. Client may also choose to fetch only required field from resources instead of pulling all fields which might not be required.

## _sort

This parameter is used to indicate the order in which the results will be returned. An optional '-' prefix can be used with the parameter which indicates that decreasing order; in its absence, the parameter is applied in increasing order. Sorting can be applied on multiple fields using a comma-separated list of search parameter. Search parameters of only date & string type can be used as a sort parameter which are defined for individual resources in Query Parameter section.

**Syntax**: https://<base-url>/minerva/fhir/r4/<FHIR_Resource>?_sort=<Parameter>

```
GET [base]/Patient?_sort=name,-birthdate
```

For the above request, the response will have data in ascending order on name. If the name is same, data will be sorted in descending order in birth date.

## _count

The parameter _count is defined as an instruction to the server regarding how many resources should be returned in a single page. The search result set contains the URLs that the client uses to request additional pages from the search set in bundle response as links.

```
<link>
   <relation value="next"/>
   <url value="http://example.org/Patient?name=peter&_count=10&_skip=10"/>
</link>
```

The server repeats the original _count parameter in its returned page links. If _count has the value 0, the server returns a bundle that reports the total number of resources that match in Bundle.total, but with no entries. The _count parameter has no impact on the value of Bundle.total.

**Syntax**: https://<base-url>/minerva/fhir/r4/<FHIR_Resource>?_count=<Number of records to return>

```
GET [base]/Patient?_sort=name&_count=10
```

For the above request, the response will have only 10 results matching with the given criteria.

## _include

_include parameter is used to indicate that a related resource be included in the search results. This helps to reduce the overall network delay of repeated retrievals of related resources. This is useful when the client is searching on a clinical resource, but for every such resource returned, the client will also need the subject (patient) resource that the clinical resource refers to.

Each _include parameter specifies a search parameter to join on. These search parameters are defined for individual resources in Query Parameter section.

Parameter values for _include have two parts, separated by ':' character:

1. The name of the source resource from which the join comes
2. The name of the search parameter which must be of type *reference*

**Syntax**: https://<base-url>/minerva/fhir/r4/<FHIR_Resource>?_include=<Resource>:<Reference>

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

For the above request, the response will have patient and encounter resource included in the bundle being referred in condition resource.

## _elements

A client can request a specific set of elements be returned as part of a resource in the search results using the _elements parameter. The _elements parameter consists of a comma-separated list of base element names such as, elements defined at the root level in the resource. Only elements that are listed are to be returned. The list of elements does not apply to included resources.

**Syntax**: https://<base-url>/minerva/fhir/r4/<FHIR_Resource>?_elements=<Element 1>,<Element 2>

```
GET [base]/Patient?_elements=identifier,active,link
```

For the above request, the response will have only identifier, active and link field only in the Bundle's entry.