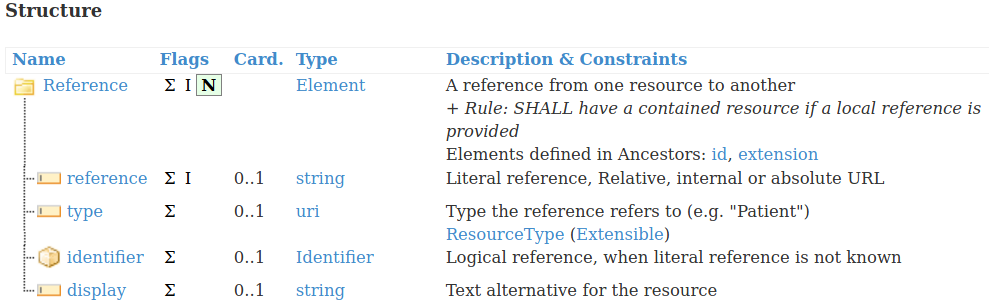
HL7 FHIR Documentation

## Resource References

I riferimenti (References) sono sempre definite tramite un unico verso (one direction) - da una risorsa (source) verso l’altra (target).

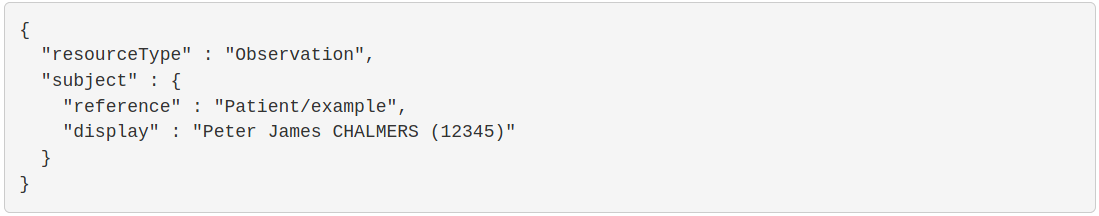
Le risorse contengono due tipi di riferimenti ad altre risorse:

* **Resource references** - riferimento generale tra risorse;
* **Canonical references** - riferimento tramite canonical URL (PREFERITO).



### > Rendering References in Resource Narratives

As an example, consider an Observation with a patient reference:



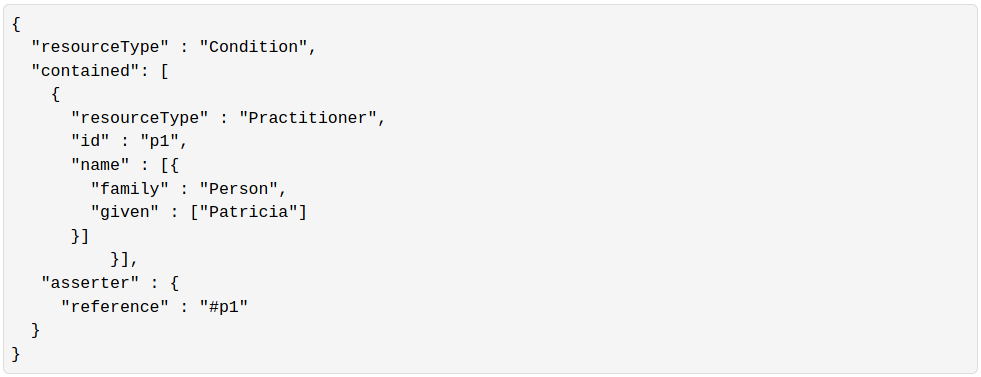
The canonical type does not have a display element because applications are generally expected to carry cached copies of the resources that are the target of the canonical references

### > Contained Resources

A **contained resource** is a resource that is embedded in another resource. A contained resource only exists in the context of the containing resource, and it cannot exist on its own. This means that a contained resource is not externally identifiable.

**Example**

It is quite common that a medication order contains medication names only, without any identifying information. In this case, you should send the Medication resources as contained resources in the MedicationRequest resource.



An internal contained reference always starts with **#** followed by the id of the contained resource. Note that this id does not exist outside the containing resource. So it is not possible to run a GET request on Practitioner/practitioner1 to achieve the contained resource from your server.

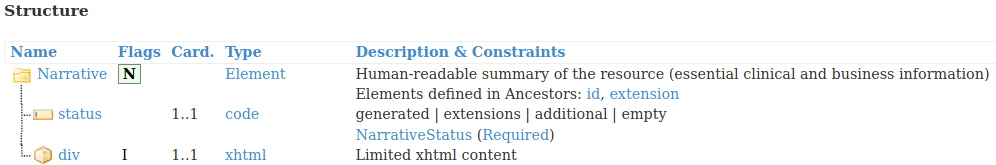
## Narrative

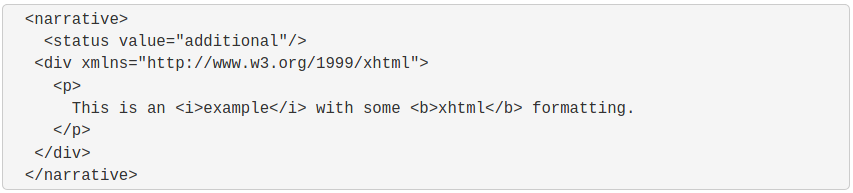
Any resource that is a [DomainResource](https://www.hl7.org/fhir/domainresource.html) (all resources except [Bundle](https://www.hl7.org/fhir/bundle.html), [Parameters](https://www.hl7.org/fhir/parameters.html) and [Binary](https://www.hl7.org/fhir/binary.html)) may include a human-readable narrative that contains a summary of the resource and may be used to represent the content of the resource to a human.

* Narrative for a resource SHOULD include summary information about any referenced resources.
* Structured data SHOULD NOT generally contain information of importance to human readers that is omitted from the narrative.
* The narrative SHALL have some non-whitespace content

**Example**

A MedicationRequest might include brief summary information about the referenced patient, prescriber and medication





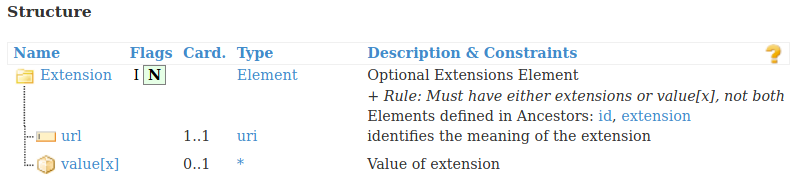
In some contexts, it is useful to link between the two representations of the same content: structured data, and human readable narrative. This can be used to assert that the text is a representation of the data



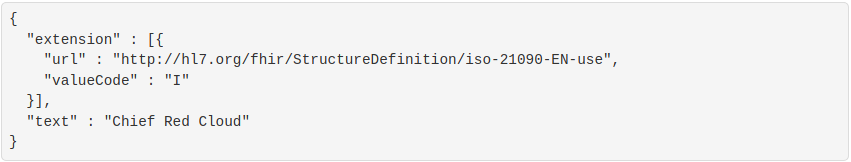
## 

## Extensibility

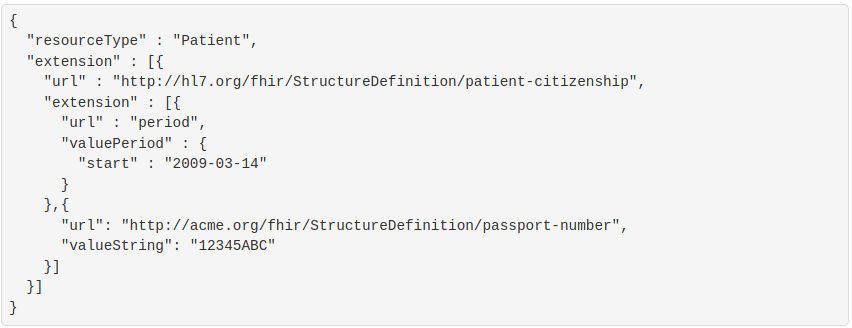
Every element in a resource can have extension child elements to represent additional information that is not part of the basic definition of the resource.



* The url is a mandatory attribute / property and identifies a retrievable [extension definition](https://www.hl7.org/fhir/defining-extensions.html).



As an example, consider extending a patient with information about citizenship. This extension can be extended again, by adding a "passport-number" extension.



## Terminologies

Many elements in the FHIR resources have a **coded value**: some fixed string assigned elsewhere that identifies some defined "concept”.

The general pattern for representing coded elements is using the following four elements:

|  |  |
| --- | --- |
| system | A URI identifies the system |
| version | Identifies the version of the system |
| code | A string pattern that identifies a concept as defined by the code system |
| display | A description of the concept as defined by the code system |

## 

## Restful API

* **base**: indirizzo del server
* **mime-type**:
  + XML: application/fhir+xml
  + JSON: application/fhir+json
  + RDF: application/fhir+turtle
* **type**: il nome del tipo di risorsa (e.g. "Patient")
* **id**: valore assegnato dal server
* **vid**: mantiene lo storico
* **compartment**: nome del compartimento (set di risorse che condividono caratteristiche)
* **parameters**: parametri inseriti all’interno del URL

### > General parameters

* **\_format**: permette di richiedere le informazioni in un particolare formato;
* **\_pretty**: richiede di stampare la risposta in un formato di facile lettura;
* **\_summary**: richiede al server di inviare solamente una porzione delle risorse. Può assumere valore:
  + true: ritorna una sottoparte marcata come “summary”;
  + text: ritorna solamente l’elemento ‘text’;
  + data: rimuove l’elemento ‘text’;
  + count: ritorna solamente il numero di risorse trovate;
  + false: ritorna tutte le parti delle risorse.
* **\_elements**: permette di richiedere uno specifico set di elementi come nel caso di \_summary ma ne consente l’inserimento manuale.



### > Read



* GET di una risorsa rimossa -> **410**
* GET di una risorsa non presente -> **404**

### > VRead



Effettua una Read su una specifica versione della risorsa interessata.

* GET di una risorsa rimossa -> **410**
* Se il server non supporta il versioning -> **404**

### > Update



L’operazione di Update ha risvolti differenti:

* Se la risorsa ([id]) esiste -> crea una nuova versione della risorsa **200** ;
* Se la risorsa ([id]) non esiste -> crea una nuova risorsa con l’id specificato **201**.

L’operazione di Update può essere rifiutata dal server:

* Fallisce la validazione di FHIR -> **400** - Bad request;
* Autorizzazione richiesta -> **401** - Not Authorized;
* Risorsa non presente -> **404** - Not found;
* Conflitto di versione -> **409**/**412;**
* La risorsa non rispetta le regole del server -> **422**.

### > Patch



Alternativa ad Update nel caso si voglia risparmiare banda. Possono essere specificati nella richiesta solo alcuni elementi e non tutta la risorsa.

### > Delete



Nel caso di server che mantengono lo storico delle versioni, in alcuni casi lo storico non viene effettivamente rimosso ma marcato come “delete”. Ciò significa che la risorsa cancellata e il suo storico possono essere ripristinate in futuro.

### > Create

Se il server supporta il versioning delle risorse, per la risorsa creata viene istanziato anche uno storico.

L’operazione di Create può essere rifiutata dal server:

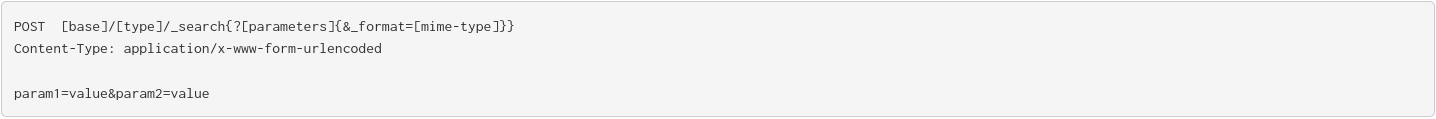
* Fallisce la validazione di FHIR -> **400** - Bad request;
* Il tipo della risorsa non esiste -> **404** - Not found;
* La risorsa non rispetta le regole del server -> **422.**

### > Search

Permette di ricercare e filtrare un set di risorse sulla base di parametri [parameters].



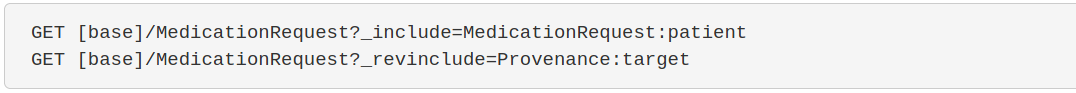
La ricerca può essere effettuata tramite POST passando i parametri all'interno del *body* della richiesta.

****

è possibile applicare dei parametri di ricerca validi per tutti i tipi di risorse:

* \_content, \_id, \_lastUpdated, \_profile, \_query, \_security, \_source, \_tag;
* [\_text](https://www.hl7.org/fhir/search.html#text) e [\_filter](https://www.hl7.org/fhir/search.html#filter);

#### >> Includere altre risorse nel risultato



* \_include: include nel risultato le risorse alle quali la risorsa specificata fa riferimento. Nell’esempio oltre alle *MedicationRequest* vengono richiesti anche i subject cioè i *Patient* ai quali le prime fanno riferimento.
* \_revinclude: in questo caso si ottengono le risorse che fanno riferimento a quella specificata. Nell’esempio oltre alle *MedicationRequest* vengono richieste anche i *Provenance* che fanno riferimento alla prime.

#### >> Contained Resources



Permette di effettuare ricerche filtrando per risorse contenute all’interno di altre.

Specificando il parametro \_contained con valore:

* false (default): non ritorna le risorse contenute;
* true: ritorna solamente le risorse contenute;
* both: ritorna entrambe le risorse;

### > History

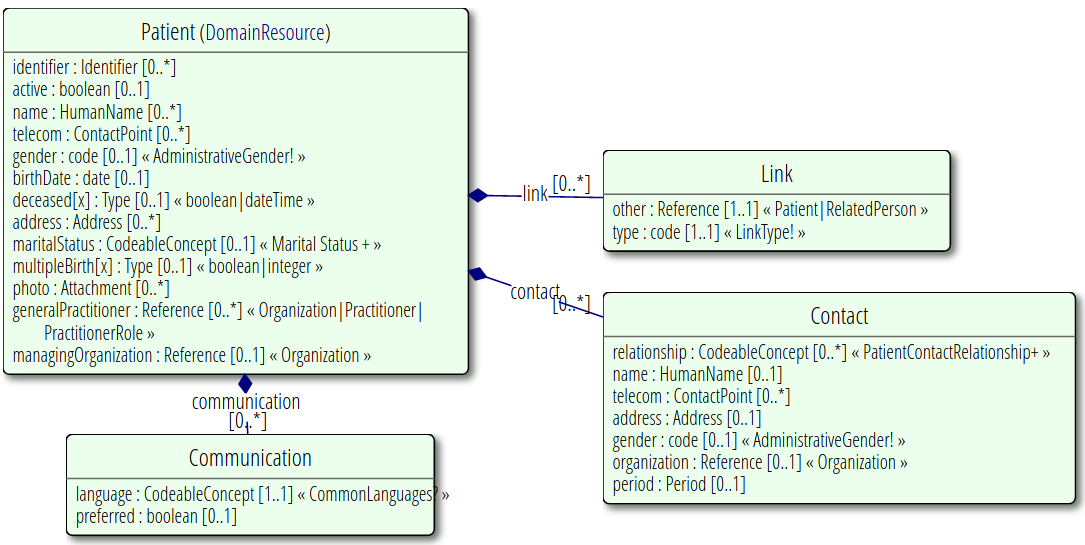


Tramite l’operazione History è possibile ricavare lo storico:

* di una particolare risorsa;
* di tutte le risorse di un certo tipo;
* di tutte le risorse supportate dal sistema.

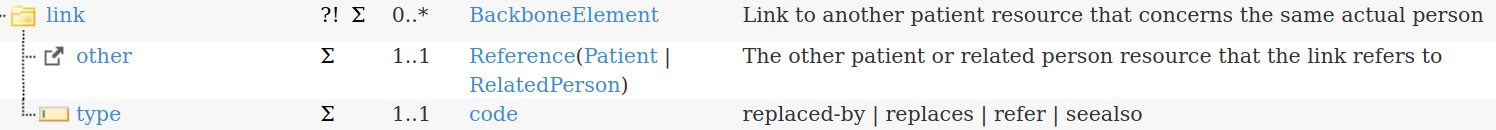
Per ogni versione ricevuta viene specificata anche l’operazione che ha portato alla creazione di questa: *create*, *update*, *delete*.

## Patient



### >> Linking Patients

L’elemento *link* viene usato per affermare che delle risorse *Patient* fanno riferimento allo stesso paziente. Questo fornisce una soluzione nel caso di errori di registrazione che si traducono in risorse duplicate e permette di mantenere i dati relativi ad un paziente all’interno di un sistema distribuito.



Set di valori per il parametro *type*:

* replaced-by: la risorsa *Patient* contenente il *link* non è più utilizzata;
* replaces: la risorsa *Patient* contenente il *link* è il record attualmente attivo;
* refer: la risorsa *Patient* contenente il *link* è un record attivo ma non il principale;
* seeealso: come *refer* ma non specifica il ruolo/importanza.

### >> Search parameters

