### Profili: tipologie

Il disegno di un profilo FHIR dipende fortemente dal suo scopo, che può essere molto specifico o totalmente generico.

Senza considerare per il momento la stratificazione dei profili per giurisdizione, per le finalità di HL7 Italia si può in prima istanza pensare ad una classificazione come quella indicata nella figura 1, in cui si individuano due distinte dimensioni:

1. il contesto d’uso (in verticale nella figura)
2. ed il tipo di target (in orizzontale)

|  |
| --- |
| Figura 1 – Tipologie di profili considerati  |

In pratica, un profilo può essere disegnato per essere usato per un particolare ***contesto*** d’uso o use case (e.g. Gestione delle vaccinazioni); o definire delle caratteristiche comuni a tutti i possibili contesti (profili ***base***).

Indipendentemente dal contesto, inoltre, un profilo può descrivere come usare una certa risorsa FHIR per un particolare attore od entità (e.g. come descrivere una azienda sanitaria) (***specifico***); o più genericamente definire le caratteristiche comuni a tutti diversi attori od entità che possono essere rappresentate con quella risorsa FHIR (e.g. profilo Practictioner) (***generico***).

**Questa guida definisce profili di tipo base.** Future guide potranno specificare come queste risorse base possano essere utilizzate per specifici casi d’uso.

### MustSupport

#### Descrizione generale

**mustSupport** è una proprietà di un profilo che indica che un certo elemento deve “essere supportato” dai sistemi che dichiarano di essere conformi a quel profilo.

Il significato di “essere supportato” non è definito dallo standard stesso, ma deve essere descritto da chi specifica il profilo.

Il lettore non deve confondere inoltre *mustSupport* e *cardinalità* di un elemento: è infatti del tutto lecito avere elementi opzionali che sono mustSupport.

Maggiori dettagli su questa proprietà sono forniti in <https://www.hl7.org/fhir/profiling.html#mustsupport>.

I seguenti paragrafi descrivono come interpretare questa proprietà per questa guida:

#### Uso del mustSupport nei modelli logici

In questa guida la proprietà mustSupport viene utilizzata nei modelli logici per flaggare gli elementi che costituiscono il set minimo di informazioni associate ad una specifica classe.

Gli elementi *mustSupport* POSSONO essere obbligatori od opzionali.

Modelli logici derivati e specifiche implementative “compliant” con questo modello DEVONO:

1. esplicitare come tali elementi siano realizzati nel modello o nelle specifiche derivate
2. Conservare il mustSupport = true se sono tali modelli/specifiche sono rappresentati attraverso risorse FHIR (modelli logici o profili)

#### Uso nei Profili

Nel contesto di questa guida la proprietà *mustSupport* DEVE essere interpretata come segue:

* Gli Implementatori conformi a questa Guida durante la creazione di contenuto:
	+ DEVONO essere capaci di includere gli elementi mustSupport.
* Gli Implementatori conformi a questa Guida quando ricevono del contenuto:
	+ DEVONO essere capaci di processare le istanze contenenti elementi mustSupport data elements senza generare errori o causare malfunzionamenti dell’applicazione.
	+ DOVREBBERO essere capaci di visualizzare tali contenuti per uso umano; o di processarli (e.g. memorizzare) per altri scopi; coerentemente con il tipo di elemento in oggetto ed il contesto di uso.
	+ DEVONO essere capaci di processare istanze di queste risorse che contengono elementi mustSupport dichiarando le informazioni mancanti.

#### Dati mancanti

##### Elementi mustSupport opzionali

Se un “creatore” (i.e. un sistema che genera del contenuto conforme a questa guida) non ha dati da includere in un elemento opzionale mustSupport, allora quell’elemento viene omesso.

##### Elementi mustSupport obbligatori

Se un “creatore” (i.e. un sistema che genera del contenuto conforme a questa guida) non ha dati da includere in un elemento obbligatorio mustSupport, la ragione di questa assenza deve essere specificata come segue:

1. Per elementi “non-codificati”, usare l’estensione DataAbsentReason nel data type.

Esempio:

{

 "resourceType" : "Patient",

 ...

 "birthDate":[

 "extension" : [

 "url" : "http://hl7.org/fhir/StructureDefinition/data-absent-reason",

 "valueCode" : "unknown"

 ]

 ]

 }

1. Per elementi *codificati*:
* In caso di binding strength example, preferred, or extensible  (CodeableConcept datatypes):
	+ se i sistemi di origine hanno testo ma nessun dato codificato, allora si usa il solo elemento *text*.
	+ Se invece non è disponibile né testo né dati codificati che rappresentano concetti reali (cioè non codici di eccezione):
		- usare il codice di eccezione appropriato incluso nel value set se disponibile
		- usare il concetto codificato appropriato dal Code System DataAbsentReason, se il value set non ha codici di eccezione.
* required binding strength (CodeableConcept or code datatypes):
	+ usare il codice di eccezione appropriato incluso nel value set

### Interpretazione dello slicing

#### Open slicing

Una delle funzionalità importanti e utili della profilazione FHIR è lo [slicing](http://hl7.org/fhir/profiling.html#slicing), attraverso cui è possibile definire più insiemi di vincoli per un caso d'uso specifico per un elemento od un gruppo di elementi complessi di una risorsa. Lo slicing può essere utilizzato con elementi ripetitivi, type choice od elementi non ripetitivi. La maggior parte delle slices specificate in questa guida è di tipo ***open*** (ovvero, lo slicing.rules  non è closed), il che significa che è possibile che istanze di risorse con elementi che non corrispondono a nessuna delle slice definite siano ancora conformi al profilo, fintanto che soddisfano i restanti vincoli del profilo.

Avere questo chiaro è importante per comprendere correttamente i profili pubblicati, in particolare quelli che usano il value set binding for discriminare le slices.

In questi casi, infatti, è consentito utilizzare value set o code systems alternativi che non sono quelli indicati nelle slice come "*required*".

Prendiamo come esempio lo slicing del Condition.code nel profilo [Condition-uv-ips](http://hl7.org/fhir/uv/ips/StructureDefinition-Condition-uv-ips.html). Questo profilo specifica due slices per questo elemento:

1. Uno per indicare un problema a partire dal SNOMED CT Global Patient Set (GPS) ( [CORE Problem List Finding/Situation/Event (GPS) - IPS](http://hl7.org/fhir/uv/ips/ValueSet-core-problem-finding-situation-event-gps-uv-ips.html) )
2. Uno per rappresentare l’assenza o la non conoscenza di problemi rilevanti ( [Absent or Unknown Problems - IPS](http://hl7.org/fhir/uv/ips/ValueSet-absent-or-unknown-problems-uv-ips.html) )

Poiché lo slicing è open, la presenza di questi due required value sets non impedisce agli implementatori o agli specificatori di rappresentare un problema utilizzando un codice di un code system alternativo (ad esempio ICD-11) come codice primario.

Un altro esempio è dato in questa guida dallo slicing per l’elemento [Patient.identifier](StructureDefinition-Patient-it-base.html). In questo caso sono rappresentati più possibili identificatori (codice fiscale ,id regionale, stp,…) ma una implementazione può decidere di usare uno o più di questi identificatori, od nessuno di questi; possono essere usati anche altri identificatori non citati nelle slice.

#### Slice e cardinalità

Le combinazioni di cardinalità ammesse nello slicing sono descritte in [Slice cardinality](https://www.hl7.org/fhir/profiling.html#slice-cardinality).

Una circostanza particolare da interpretare correttamente è il caso in cui l’elemento profilato ha cardinalità 0.. con slice 1.. In questo caso significa che l’elemento è opzionale, ma se presente deve essere presente almeno un elemento conforme allo slice 1..

### Translitterazione dei nomi

Esistono casi in cui risulta necessario catturare informazioni testuali usando rappresentazioni diverse (e.g. “Pellé” vs “Pelle’” oppure “Ægir” vs “Aegir”).

Per gestire questi casi si suggerisce l’utilizzo dell’estensione [translation](https://www.hl7.org/fhir/extension-translation.html) distinguendo attraverso il tag *lang* le diverse rappresentazioni, valorizzato con un codice conforme al BCP 47 [Tags for Identifying Languages](https://tools.ietf.org/html/bcp47). La differenziazione fra le diverse rappresentazioni può essere ottenuta se necessario utilizzando i subtag *script* e *variant* od in casi particolari *private use*.