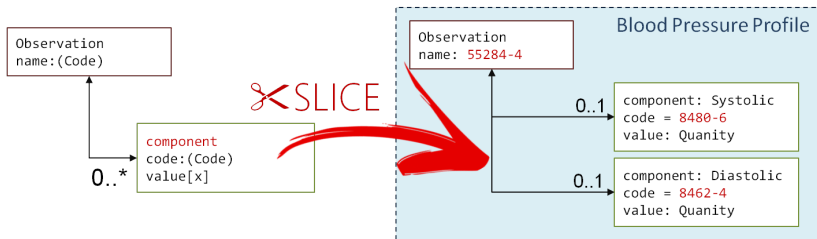


- Häufig enthalten Ressourcen Elemente, die mehrfach vorkommen können
 - Das Basis-Profil für die Ressource Patient erlaubt eine Liste des Elementes **Patient.identifier (0..*)**
- Slicing ermöglicht mehrfach vorkommende Elemente (Listen) in Sub-Listen (**Slices**) zu *zerteilen*
- Jeder Slice verfügt über unterschiedliche Restriktion bzgl.
 - erlaubter Elemente
 - sowie Einschränkungen bzgl. Datentyp, Kardinalität, erlaubte Codes, etc.
- Ein Slice erhält im Sinne eines konkreten Anwendungsfalls zusätzliche Semantik

- Beispiel: Blood Pressure Observation
 - Es wird je ein **Slice** für die Systole und die Diastole erstellt
 - Die Definition der Slices erfolgt im Profil



- Beispiel 11 **hard**
 - Zukünftig soll der AustrianPatient zusätzlich über eine Sozialversicherungsnummer verfügen
 - Weitere Patient.identifier Elemente sollen zulässig sein



- Beispiel 11 **hard**
 - Slicing-Verhalten wird über eigenes ValueSet **SlicingRules** festgelegt

Code	Display	Definition
closed	Closed	No additional content is allowed other than that described by the slices in this profile.
open	Open	Additional content is allowed anywhere in the list.
openAtEnd	Open at End	Additional content is allowed, but only at the end of the list. Note that using this requires that the slices be ordered, which makes it hard to share uses. This should only be done where absolutely required.

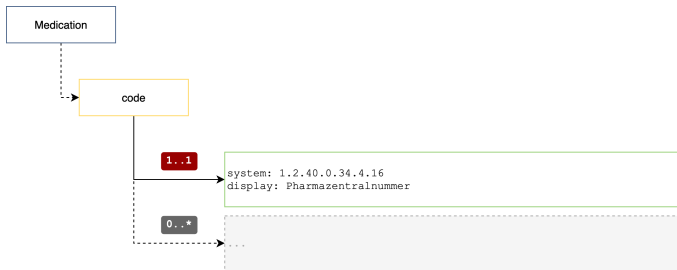
- Resultat Beispiel 11 **hard**

```
<element>
...
<path value="Patient.identifier"/>
<slicing>
  <rules value="openAtEnd"/>
</slicing>
<min value="1"/>
<max value="*/>
...
</element>
<element>
...
<path value="Patient.identifier"/>
<name value="svnr"/>
<min value="1"/>
<max value="1"/>
<base>
  <path value="Patient.identifier"/>
  <min value="0"/>
  <max value="*/>
</base>
...
</element>
```

Regel für das Slicing-Verhalten

Slice für Sozialversicherungsnummer

- Beispiel 12 **hard**
 - Das Profil für **eMedication** soll um den Code für die Pharmazentralnummer erweitert werden
 - Weitere **Medication.code** Elemente sollen zulässig sein



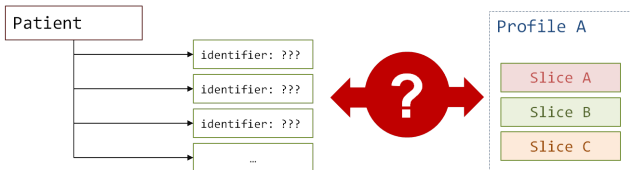
- Resultat Beispiel 12 hard

```
<element id="Medication.code.coding">
...
  <path value="Medication.code.coding" />
  <slicing>
    <rules value="openAtEnd" />
  </slicing>
</element>
<element id="Medication.code.coding:
  pharmazentral">
...
  <path value="Medication.code.coding" />
  <sliceName value="pharmazentral" />
  <max value="1" />
  <fixedCoding>
  <system value="1.2.40.0.34.4.16" />
  <display value="Pharmazentralnummer" />
  </fixedCoding>
...
</element>
```

Regel für das Slicing-Verhalten

Slice für die Pharmazentralnummer

- Wenn eine **StructureDefinition** Slices enthält
 - Prüfen jedes Listenelementes einer konkreten Ressource, gegen die im jeweiligen Slice definierten Regeln
 - Zusätzliches Konzept benötigt, um Elemente einer Liste effizient einem Slice zuzuordnen
 - Verwendung eines Discriminators (in FHIR)
 - Unterscheidung zwischen verschiedenen Slices
 - Discriminator enthält Pfadangabe des zu "beschreibenden" Elements



- Beispiel 13 **hard**
 - Für den im vorhergehenden Beispiel erstellten Slice wird ein Discriminator benötigt, der eine Differenzierung zwischen der Sozialversicherungsnummer und anderen **Patient.identifier**-Elementen ermöglicht

- Resultat Beispiel 13 hard

```
<element>
  ...
  <path value="Patient.identifier"/>
  <slicing>
    <discriminator value="assigner.reference"/>
    <rules value="openAtEnd"/>
  </slicing>
  <min value="1"/>
  <max value="*/>
  <base>
    <path value="Patient.identifier"/>
    <min value="0"/>
    <max value="*/>
  </base>
  <isSummary value="true"/>
  ...
</element>
```

Identifier

use : code [0..1] « IdentifierUse! »
type : CodeableConcept [0..1] « Identifier Type + »
system : uri [0..1]
value : string [0..1]
period : Period [0..1]
assigner : Reference [0..1] « Organization »

- Beispiel 14 **hard**
 - Für den erstellten Slice auf das Element **code** im Profil **eMedication** wird ein Discriminator benötigt, der es erlaubt die verfügbaren Codes zu unterscheiden und so die eindeutige Identifikation der Pharmazentralnummer erlaubt.

- Resultat Beispiel 14 hard

```
<element id="Medication.code.coding">
...
  <path value="Medication.code.coding" />
  <slicing>
    <discriminator>
      <type value="value" />
      <path value="system" />
    </discriminator>
    <rules value="open" />
  </slicing>
</element>
<element id="Medication.code.coding:pharmazentral
">
  <path value="Medication.code.coding" />
  <sliceName value="pharmazentral" />
  <max value="1" />
  <fixedCoding>
    <system value="1.2.40.0.34.4.16" />
    <display value="Pharmazentralnummer" />
  </fixedCoding>
...
</element>
```

Coding

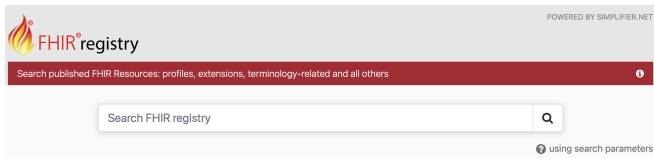
system : uri [0..1]
version : string [0..1]
code : code [0..1]
display : string [0..1]
userSelected : boolean [0..1]

- HL7 stellt IGPublisher Werkzeug für das Paktieren und Publizieren von Implementierungsleitfäden bereit
 - IGPublisher Dokumentation
- Konvertierung eines **ImplementationGuide** in 3 verschiedene Repräsentationen
 - Ressourcen für die direkte Verwendung (xml, json, ttl)
 - HTML-Webseite für Publikation
 - Zip-Dateien für die Verwendung in Implementierungen
- Infos, Hilfestellungen zur Verwendung bei **fhir-tc**

Offizielle FHIR Registries



- HL7 Profil Registry
 - <http://registry.fhir.org>
 - offizielle Profile publiziert von HL7 und zugehörigen Arbeitsgruppen
- FHIR Implementation Guide Registry
 - <http://fhir.org/guides/registry>
 - verwaltet offizielle Implementation Guides publiziert von HL7



Outline



Profiling

Conformance Module

ImplementationGuide

Medication Modul

StructureDefinition

Extensions in FHIR

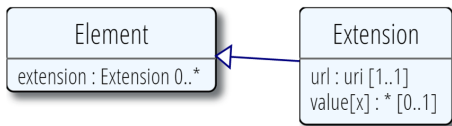
SearchParameter

FHIR Operations Framework

CapabilityStatement

Extensions I

- Extensions sind fundamentales Konzept in FHIR (vgl. 80/20 Regel)
- Erweiterung der Basisspezifikation um zusätzliche Elemente
- Alle Elemente in Ressourcen und Datentypen beinhalten zusätzliches, optionales Element **extension (0..*)**



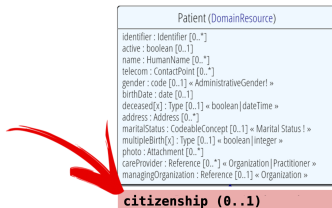
Information

Da es sich bei Extension ebenfalls um ein Element handelt, können Extensions beliebig geschachtelt werden.

- Extension besteht aus
 - **url** : verpflichtend, eindeutiger Bezeichner der Extension
 - **value[x]** : verschiedene value Types für die Extension stehen zur Auswahl (String, Integer, etc.)
- Extension verwendet entweder **value[x]** -Element oder fügt weiteres **extension** -Element hinzu
- Wenn korrekte Verarbeitung von Ressource ohne Extensions nicht sichergestellt → **modifierExtension**

Erstellen einer Extension

- Beispiel 15 **hard**
 - Der **AustrianPatient** soll um ein zusätzliches Element **citizenship** erweitert werden
 - Als Grundlage für die Extension soll die offizielle HL7 **patient-citizenship** verwendet werden
 - <http://hl7.org/fhir/StructureDefinition/patient-citizenship>



Erstellen einer Extension II



- Resultat Beispiel 15 **hard**
- Verwendung der offiziellen HL7 **citizenship** Extension

```
<element id="Patient.extension:citizenship">  
  <path value="Patient.extension" />  
  <sliceName value="citizenship" />  
  <type>  
    <code value="Extension" />  
    <profile value="http://hl7.org/fhir/StructureDefinition/patient-citizenship" />  
  </type>  
</element>
```

Outline



Profiling

Conformance Module

ImplementationGuide

Medication Modul

StructureDefinition

Extensions in FHIR

SearchParameter

FHIR Operations Framework

CapabilityStatement

- Vereinfachte Such- und Filteranfragen für Ressourcen
- **SearchParameter** definiert
 - die Verwendung der Parameter am Client
 - die serverseitige Interpretation der Parameter
 - wie der *name* des **SearchParameter** auf ein Element abgebildet wird.

birthdate	Patient.birthDate
death-date	Patient.deceased.as(DateTime)

- Es existiert eine Registry für definierte SearchParameter! [1]

Beispiel 16 moderate

- Es soll ein Suchparameter für den **AustrianPatient** definiert werden, der es nach Patienten mit bestimmter **Nationalität** zu suchen.

GET [base]/Patient?citizenship = urn:iso:std:iso:3166|AT

system|code



Information

Der Name eines Suchparameters muss sich zwangsläufig nicht mit dem Namen eines Elementes einer Ressource decken.

Resultat Beispiel 16 moderate

```
<resource>
  <type value="Patient"/>
  <profile>
    <reference value="http://hl7.org/fhir/Profile/Patient"/>
  </profile>
  <interaction>
    <code value="read"/>
  </interaction>
  <interaction>
    <code value="vread"/>
  </interaction>
  <interaction>
    <code value="create"/>
  </interaction>
  <interaction>
    <code value="search-type"/>
  </interaction>
  <searchParam>
    <name value="citizenship"/>
    <type value="token"/>
  </searchParam>
</resource>
```

Neu festgelegter Suchparameter

Resultat Beispiel 16 moderate

- Annotieren entsprechender Methoden mit `@Search`
- `@RequestParam` definiert Parameternamen
- Übersicht möglicher Parametertypen unter: <http://...>

```
@Search public List<AustrianPatient> findPatientsForNationality  
(@RequestParam (name="nationality") TokenParam code) {  
    String systemVal = code.getSystem();  
    String codeVal = code.getValue();  
    // Implement backend retrieval for AustrianPatient matching desired criteria  
}
```


Outline



Profiling

Conformance Module

ImplementationGuide

Medication Modul

StructureDefinition

Extensions in FHIR

SearchParameter

FHIR Operations Framework

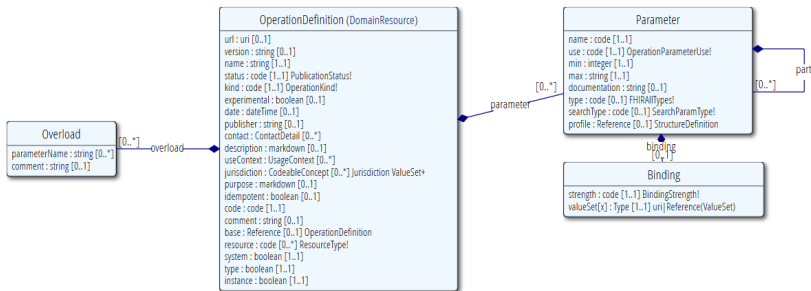
CapabilityStatement

Operationen STU3 (== R4)



- Alles mit \$ ist Operation
- POST [http://fhir.someserver.org/fhir/Patient/1/\\$everything](http://fhir.someserver.org/fhir/Patient/1/$everything)**
- Operationsaufruf entweder mit POST oder GET
 - POST – Operation kann Änderungen auf Ressource verursachen
 - GET – Idempotente Operationen (jeder Aufruf erzeugt exakt gleiches Ergebnis) oder Operationen welche Daten nicht ändern
- Operationen können auf verschiedene Levels definiert werden
 - Direkt auf den Endpoint (<http://example.com/fhir>)
 - Bsp.: \$extensions → Finden aller Extensions am Server
 - Auf einen Ressourcentyp (<http://example.com/fhir/Patient>)
 - Bsp.: \$count → Zählen aller Ressourcen
 - Auf eine spezifische Instanz (<http://example.com/fhir/Patient/1>)
 - Bsp.: \$patientSummary → Patient Summary für Patienten
 - Auf eine spezifische Version (http://example.com/fhir/Patient/1/_history/3)
 - Bsp.: \$difference → Unterschiede zur aktuellen Version

OperationDefinition



OperationDefinition (simplified) |



```
<OperationDefinition xmlns="http://hl7.org/fhir">
  <id value="Questionnaire-populate"/>
  <url value="http://hl7.org/fhir/OperationDefinition/Questionnaire-populate"/>
  <name value="Populate Questionnaire"/>
  ...
  <code value="populate"/>
  <system value="false"/>
  <type value="Questionnaire"/>
  <instance value="true"/>
  ...
</OperationDefinition>
```

Eindeutige URL

Code → Aufruf mit \$populate

System → Ob Operation auf Endpoint anwendbar ist

Operation auf Ressourcentyp (0..*)

Operation auf Instanzlevel

```
<OperationDefinition xmlns="http://hl7.org/fhir">
  ...
  <parameter>
    <name value="local"/>
    <use value="in"/>
    <min value="0"/>
    <max value="1"/>
    <documentation value="Human Readable Info on Parameter."/>
    <type value="boolean"/>
  </parameter>
  ...
</OperationDefinition>
```

Inparameter

Optional

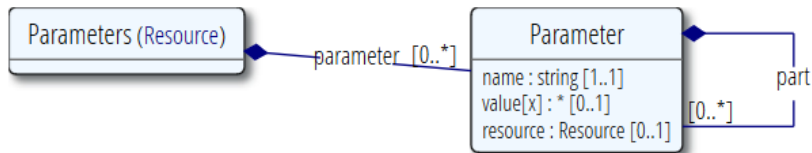
Type → Simple | Complex Datatype, Ressource

```
<OperationDefinition xmlns="http://hl7.org/fhir">
  ...
  <parameter>
    <name value="return"/>
    <use value="out"/>
    <min value="1"/>
    <max value="1"/>
    <documentation value="The partially (or fully)-populated set of answers for the specified
      Questionnaire"/>
    <type value="QuestionnaireResponse"/>
  </parameter>
  ...
</OperationDefinition>
```

Outparameter

Required

- Aufruf → Ohne Parameter GET oder POST
- Aufruf → Mit Ressource Parameters im Body
- Antwort → Ebenfalls Ressource Parameters
- Parameters:
 - Beinhaltet eine Liste von Parametern
 - Jeder Parameter ist Datatype ODER Ressource
 - Beschrieben in OperationDefinition



Parameters

```
<Parameters xmlns="http://hl7.org/fhir">
  <id value="example"/>
  <parameter>
    <name value="start"/>
    <valueDate value="2010-01-01"/>
  </parameter>
  <parameter>
    <name value="end"/>
    <resource>
      <Binary>
        <contentType value="text/plain"/>
        <content value="VGhpcyBpcyBhIHRlc3QgZXhhbXBsZQ==" />
      </Binary>
    </resource>
  </parameter>
</Parameters>
```

Start → Typparameter

End → Ressourcenparameter

- Details: HAPI FHIR **Extended-Operations**
- Auf Type Level (in Resource Provider!):

```
@Operation (name="$everything", idempotent=true)
public Bundle patientTypeOperation(
    @OperationParam (name="start") DateType theStart,
    @OperationParam (name="end") DateType theEnd) {

    Bundle retVal = new Bundle();
    // Populate bundle with matching resources
    return retVal;
}
```

Operation \$everything

GET + POST Erlaubt

ReturnType

Parameters

- Auf Instanz Level (in ResourceProvider):

```
@Operation (name="$everything", idempotent=true)
public Bundle patientInstanceOperation(
    @IdParam IdType thePatientId,
    @OperationParam (name="start") DateType theStart,
    @OperationParam (name="end") DateType theEnd) {

    Bundle retVal = new Bundle();
    // Populate bundle with matching resources
    return retVal;
}
```

Parameters + ID

- Auf Endpoint (nicht in ResourceProvider):

```
@Operation (name="$closure" )
public ConceptMap closureOperation(
    @OperationParam (name="name") StringType theStart,
    @OperationParam (name="concept") List theEnd,
    @OperationParam (name="version") IdType theVersion) {

    ConceptMap retVal = new ConceptMap();
    // Populate bundle with matching resources
    return retVal;
}
```

Operation \$closure

Nur POST (keine Idempotenz)

ReturnType

Parameter

\$merge

Beispiel 17 moderate

- Die Operation vereint 2 Patienten **mit gleicher Identifikation**
- Ein Patient merge wird vom User validiert (neuer Patient)

Beispiel Operation II



- Resultat Beispiel 17 moderate

```
<OperationDefinition xmlns="http://hl7.org/fhir">
  <name value="merge"/>
  <status value="active"/>
  <kind value="operation"/>
  <idempotent value="true"/>
  <code value="merge"/>
  <resource value="Patient"/>
  <system value="false"/>
  <type value="false"/>
  <instance value="true"/>
  <parameter>
    <name value="with"/>
    <use value="in"/>
    <min value="0"/>
    <max value="1"/>
    <type value="id"/>
  </parameter>
</OperationDefinition>
```

```
<operation>
  <name value="merge"/>
  <definition>
    <reference value="OperationDefinition/Patient-i-merge"/>
  </definition>
</operation>
</rest>
</CapabilityStatement>
```

Outline



Profiling

Conformance Module

ImplementationGuide

Medication Modul

StructureDefinition

Extensions in FHIR

SearchParameter

FHIR Operations Framework

CapabilityStatement

- Dient der Beschreibung des Verhaltens und der Funktionalität eines spezifischen FHIR-Servers:
 - **UC1:** Welche Funktionalität wird vom FHIR-Server angeboten
 - **UC2:** Zur Beschreibung einer Softwarelösung
 - **UC3:** Zur Festlegung was eine spezifische Implementierung leisten soll

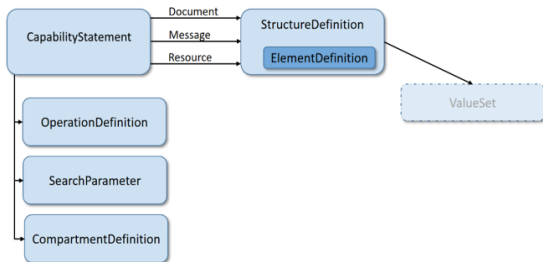


Information

CapabilityStatement ist keine gänzlich neue Ressource, vielmehr wurde die Ressource ConformanceStatement in CapabilityStatement umbenannt.

CapabilityStatement

- Ein CapabilityStatement besteht aus
 - **StructureDefinition** (Profile)
 - Documents
 - Messages
 - Resources
 - **OperationDefinition**
 - **SearchParameter**
 - **CompartmentDefinition**



- Beispiel: Abfrage des CapabilityStatements

GET [base]/metadata

```
{  
  "resourceType": "CapabilityStatement",  
  "status": "active",  
  "date": "2017-03-14T12:48:07-04:00",  
  "publisher": "Not provided",  
  "kind": "instance",  
  ...  
}
```

- Unterstützt 2 Arten von FHIR-Profiling
 - **CapabilityStatement.rest.resource.profile**
 - Definiert Funktionalität für spezifische Ressourcen

```
POST [base]/Patient
GET [base]/Patient
GET [base]/Patient/1
...
```

- Unabhängig von spezifischen Anwendungsfällen
- **CapabilityStatement.profile**
 - Definiert Profile für alle unterstützten Anwendungsfälle, unabhängig von einer spezifischen Ressource

Abgrenzung zu **ImplementationGuide**

- **CapabilityStatement** - definiert die Funktionalität einer FHIR-API
- **ImplementationGuide** - beschreibt einen Anwendungsfall unter Verwendung von Ressourcen und Profilen

HL7 International (2018a). *Defined Search Parameters*. URL:

<http://build.fhir.org/searchparameter-registry.html>.

Furore Health Informatics (2018). *Simplifier.NET*. URL:

<https://www.simplifier.net>.

HL7 International (2018b). *Resource CompartmentDefinition*. URL:

<http://build.fhir.org/compartmentdefinition.html>.

HL7 International (2018c). *FHIR*. URL:

<http://wiki.hl7.org/index.php?title=FHIR>.

HL7 International (2017). *Welcome to FHIR*. URL: <https://www.hl7.org/fhir/>.

Kramer, Ewout (2018). *SlideShare Ewout Kramer*. URL:

<http://de.slideshare.net/ewoutkramer>.