

IndyFHIR

**Capability Statement
ClinFHIR
GraphQL
(quick overview)**

Capability Statement

- Provides a declaration of what an implementation supports:
 - **rest (RestfulAPI)**
 - mode (client|server)
 - resource
 - type (Resource Type code)
 - interaction (code=read,update, etc.)
 - searchInclude
 - searchRevInclude
 - searchParam (name, type, definition)
 - interaction (operations supported such as: transaction, batch, search-system, history-system)
 - operation
 - Non-resourceType specific / custom queries
 - **messaging (Messages Supported)**
 - endpoint
 - event or supportedMessage description
 - Repeated
 - **document (Documents supported)**
 - mode (Producer | Consumer)
 - documentation (Description of support)
 - Profile (StructureDefinition)
 - Repeated

Kinds of Capability Statement

- Implementation
 - Capabilities of a specific installation of a service
 - Specific URLs such as for oauth2
- Capability
 - Capabilities independent of an installation
 - Won't specify actual URLs
 - Service side of the contract
- Requirements
 - Specification of what a client application requires
 - Could be used as part of an implementation guide or proposal for a service that supports a desired workflow

Use Cases

- API - Server
 - Document Resources and Services you support
 - Document Search Parameters
 - Document CRUD Operations
 - Document OAuth2 and other security parameters
- API - Client
 - Examine
 - Security Mechanisms
 - Supported Resources
 - Supported Operations
 - Supported Search Params

Messaging - Service

- Document Endpoints / Protocols
- Specify Messages or Events
- Specify Sender vs. Receiver

Messaging - Client

- Discover Supported Messages / Events

Document

- Document What FHIR Documents you can Consume or Produce

Constraints

- cpb-1: A Capability Statement SHALL have **at least one of REST, messaging or document element**.
(expression : rest.exists() or messaging.exists() or document.exists())
- cpb-12: On CapabilityStatement.rest.resource: **Search parameter names must be unique in the context of a resource**.
(expression on CapabilityStatement.rest.resource: searchParam.select(name).isDistinct())
- cpb-14: Capability Statements of **kind 'requirements'** do not have **software or implementation** elements. (expression :
(software.empty() and implementation.empty()) or kind != 'requirements')
- cpb-15: Capability Statements of **kind 'instance'** do not have **implementation elements**.
(expression : implementation.empty() or kind != 'capability')
- cpb-16: On CapabilityStatement.messaging: A Capability Statement **messaging element SHALL have either supportedMessage or event element, but not both**.
(expression on CapabilityStatement.messaging: supportedMessage.empty() != event.empty())
- cpb-2: A Capability Statement **SHALL have at least one of description, software, or implementation element**.
(expression : (description.count() + software.count() + implementation.count()) > 0)
- cpb-3: **Messaging end-point is required (and is only permitted) when a statement is for an implementation**.
(expression : messaging.endpoint.empty() or kind = 'instance')
- cpb-7: The **set of documents must be unique by the combination of profile and mode**.
(expression : document.select(profile.reference&mode).isDistinct())
- cpb-8: There **can only be one REST declaration per mode**.
(expression : rest.select(mode).isDistinct())
- cpb-9: On CapabilityStatement.rest: A **given resource can only be described once per RESTful mode**.
(expression on CapabilityStatement.rest: resource.select(type).isDistinct())

Examples

- Hapi FHIR Generated from Test Server STU3
 - <http://hapi.fhir.org/baseDstu3/metadata>
- Hapi FHIR Generated from Test Server DSTU2 (Conformance)
 - <http://hapi.fhir.org/baseDstu2/metadata>
- Hapi FHIR Generated from Test Server STU4
 - <http://hapi.fhir.org/baseR4/metadata>
- Cerner Sandbox Conformance (DSTU2)
 - <https://fhir.cerner.com/millennium/dstu2/conformance/>
- Epic
 - <https://open-ic.epic.com/FHIR/api/FHIR/DSTU2/metadata>

Consumption

Hapi FHIR Client (Java):

```
CapabilityStatement capabilityStatement =  
    client.capabilities().ofType(Cap  
        abilityStatement.class).execute();  
  
// Get type of first resource  
capabilityStatement.getRestFirstRep().getResou  
rc().get(0).getType()  
  
// Get first interaction code for first resource  
capabilityStatement.getRestFirstRep().getResou  
rc().get(0).getInteraction().get(0).getCode()  
  
// Get first search param for first resource  
capabilityStatement.getRestFirstRep().getResou  
rc().get(0).getSearchParam().get(0)
```

Javascript:

```
function capabilitySupported(caps, res, func) {  
    var found = false;  
    caps.rest[0].resource.some( function(curRes)  
    {  
        if ( curRes.type == res ) {  
            curRes.interaction.some(function( inter ) {  
                if ( inter.code == func ) {  
                    found=true;  
                    return true;  
                }  
            });  
            return true;  
        }  
    });  
    return found;  
}
```

Client Side Example / Demo

Cerner

Supports MedicationAdministration (but still errors out)

Epic

Does not support MedicationAdministration

Misc Links

Cerner - Documentation

<https://fhir.cerner.com/millennium/dstu2/>

Cerner - Sandbox Apps

<https://code.cerner.com/developer/smart-on-fhir/apps>

Epic - Documentation

<https://open.epic.com/>

Epic - Sandbox Launchpad

<https://open.epic.com/Launchpad/OAuth2Sso>

Shrimp Snomed Browser

<http://ontoserver.csiro.au/shrimp>

clin.fhir.com - Use FireFox not Chrome

- FHIR related Builders
- Query Interface
- Patient Viewer
- GraphQL Query Builder
- Many more tools

Appears to be broken today so no active demo

I'll message the group when it is back up

Facts:

Developed by David Hay

Links:

- Project:
<https://github.com/davidhay25/clinFHIR>
- Site:
<http://clin.fhir.com/>
- Wiki:
<https://clin.fhir.atlassian.net/wiki/spaces/C/LIN/pages/491526/Introduction>
- Training Links on Clin.fhir.com site - lower right

Clinfhir.com Training Videos

Scenario Builder

<https://vimeo.com/199927777>

Adding Structured Builder

<https://vimeo.com/203252984>

Logical Model

<https://vimeo.com/203277373>

FHIR Logical Model and Scenario Builder

<https://vimeo.com/200105501>

Rest Query

<https://vimeo.com/203259751>

GraphQL – Possible Future Presentation?

- Facebook API for Graph based queries
<https://graphql.org/>
- Sits on top of API / DAO Layers
- Define Schema:
 - Queries
 - Entities and Relationships (1st class)
 - Request results in detail - get back only what you want.
 - Server-Side Destructuring effectively
- Provides Slice through nested related FHIR Resources
- Possible to query reduced denormalized view based on complex criteria to simplify view model building
- Better performance from Server-side caching and destructuring optimizations
- Implementation Options
- Apollo:
 - Client side Java Script API
 - GraphQL Server
 - Engine that sits in front of a GraphQL Server but provides performance metrics, caching, and other benefits
- [GraphQL-Java](#)
<https://github.com/graphql-java>
- [David Hay blog on GraphQL for FHIR](#)

Questions?