2020-09 Argonaut Granular Controls

Short Description

This track will focus on testing proposed updates to the SMART on FHIR Implementation Guide (IG). SMART v2 will add support for granular scopes (e. g., a the data category level) and will clarify SMART's capabilities model.

Long Description

The SMART on FHIR v1 IG has been widely adopted for clinician- and patient-facing app integration into EHRs and other FHIR data systems. Based on community feedback, the Argonaut Project has undertaken a 2020 effort to revise and improve the SMART App Launch IG. A key area of focus in adding support for "granular permissions," e.g. to provide access to resources at the category level in addition to the type level. This would allow apps to request narrower access, like "all vital signs" rather than "all observations." In this connectathon track, we'll focus on testing support for an emerging set of improvements to the SMART scope language, allowing access at the category level; allowing access based on tags/labels; and allowing access to FHIR operations beyond create/read/update/delete/search.

Type

- Test an Implementation Guide

Submitting Work Group/Project/Accelerator/Affiliate/Implementer Group

- FHIR-I / Argonaut Granular Data project

Proposed Track Lead

- Gino Canessa, Josh Mandel

Related tracks

FHIR Version

- Our focus will be on FHIR R4

Specification(s) this track uses

- We'll be focused on testing a small number of additions to the scope language.
- **See "v2 scopes" overview here**
- For further details and background, see
    - https://github.com/argonautproject/2020#projects
    - http://hl7.org/fhir/smart-app-launch

Expected participants

Sign Up Sheet

Zulip stream

#smart

Track Orientation

Recorded session is available on YouTube

Track details

System Roles

SMART Server: SMART server supporting granular scopes from the draft SMART IG v2.

SMART Client: SMART client requesting access using granular scopes from the draft SMART IG v2.

Connectathon Results

Firely - Christiaan Knaap

> Got about half way through implementation of the specifications. There are no remaining blockers to finishing, just need time to do so.

Microsoft - Josh Mandel:

> Got to do plenty of ad-hoc testing against all the different servers. Learned a lot and had great discussions.
>
> Happy to see that category-based scopes and our syntax are working. Still worried about things like chained search. Would like to provide a good balance between a rich set of capabilities and ability to implement.

T-System Inc - Chuck Feltner:

> Able to do more testing against Cerner and Epic - got data back and everything is looking good.

Epic - Jake Fisher:

> Goal was to let people test Scenarios 0 & 1, seems to have been working. Overall a success.

Cerner - Max Philips:

> Everything went well. More success today than yesterday. Got standalone patient launch working with a patient that includes all the necessary data.

Apple

> Working on client. Was able to request scopes using the new syntax and get expected results.

Microsoft - Gino Canessa:

> Worked on the test implementation at https://smart.argo.run . Was able to implement Scenarios 0, 1, and 2. Updated regularly based on feedback from testing. Proxy-server and client have basic capabilities.

Scenarios

Scenario 0: Share access to resources by interaction

SMART Client requests scopes and SMART Servers grant scopes at the resource level. Specifically we'll test support for

- `patient/Observation.rs`
- `patient/Observation.crs (optional)`

After being granted this scope, a client can query for all Observations via:

- `GET Observation?patient={}`

And the following queries should be rejected or results should be redacted (if no other scopes have been granted):

- `GET Observation`
- `GET Encounter?patient={}`

Scenario 1: Share access to data by category

SMART Client requests scopes and SMART Servers grant scopes at the category level. Specifically we'll test support for

- `patient/Observation.rs?category=vital-signs`
- `patient/Observation.crs?category=vital-signs`

After being granted this scope, a client can query for all vital signs via:

- `GET Observation?patient={}&category=vital-signs`
- `GET Observation?patient={}&category=http://terminology.hl7.org/CodeSystem/observation-category|vital-signs`

And the following queries should be rejected or results should be redacted (if no other scopes have been granted):

- `GET Observation?patient={}`
- `GET Observation?patient={}&category=laboratory`

Scenario 2: Share access to data by tag

SMART Client requests scopes and SMART Servers grant scopes at the tag level. Specifically we'll test support for

- `patient/Observation.rs?_security=L`
- `patient/Observation.crs?category=http://terminology.hl7.org/CodeSystem/v3-Confidentiality|L,http://terminology.hl7.org/CodeSystem/v3-Confidentiality|M,http://terminology.hl7.org/CodeSystem/v3-Confidentiality|N`

After being granted this scope, a client can query for all vital signs via:

- `GET Observation?patient=:id&_security=L`
- `GET Observation?patient=:id&_security=L&category=vital-signs`
- `GET Observation?patient=:id&category=http://terminology.hl7.org/CodeSystem/v3-Confidentiality|L`

And the following queries should be rejected or results should be redacted (if no other scopes have been granted):

- `GET Observation?patient=:id`
- `GET Observation?patient=:id&_security=V`
- `GET Observation?patient=:id&category=laboratory`

Scenario 3: Share access to operations

SMART Client requests scopes and SMART Servers grant scopes for FHIR $-operations. TODO: decide which to test on. Possibly Patient/:id/$everything or GET [base]/DocumentReference/$docref?{parameters} (see USCoreFetchDocumentReferences) ?

Scenario 4: Share access to custom web services

SMART Client requests scopes and SMART Servers grant scopes for RESTful or webservices outside of FHIR (e.g. NCPDP or IHE or v3 or v2 over HTTP) . TODO: decide which to test on. Possibly __

Security and Privacy Considerations

This track focused on a new set of core security capabilities for fine-grained authorization of SMART clients.