# Test results - SMART App Launch 2.0.0 Connectathon29 January 11, 2022 Ricardo Quintano, Stefan Pfundtner, Timon Grob - Philips

**Client:** Java app (Spring boot)

**Server:** <https://smart.argo.run/>

## Scenario 0: Asymmetric Authorization of client

*Client provides either a static JWKS key set, or a URL endpoint that serves one.*

*Server authorizes client using provided JWKS upon token request.*

**RESULTS**

1. [SUCCESS] Expected behavior worked with static JWKS.
2. [~~FAILED~~] Our test failed with URL endpoint - the endpoint was not available to the server (localhost:8080). We believe it would work if the endpoint is available.
3. [SUCCESS] [TEST NOT PLANNED IN CONNECTATHON] Signing with a different key set - produced error (500 - Error: no key found).
4. [SUCCESS] [TEST NOT PLANNED IN CONNECTATHON] Not providing the key set - produced error (500 - Error: no key found).

## Scenario 1: Discovery of SMARTv2 Support

*SMART Client requests ".well-known/smart-configuration" JSON file and Server responds with a payload that includes:*

* *capabilities[] with "permission-v2", "authorize-post"*
* *code\_challenge\_methods\_supported[] with "S256"*
* *introspection\_endpoint*

**RESULTS**

1. [SUCCESS] capabilities[] with "permission-v2", "authorize-post"
2. [SUCCESS] code\_challenge\_methods\_supported[] with "S256"
3. [SUCCESS] introspection\_endpoint

Tested in the Java client app and Postman

## Scenario 2: Share access to Observations with SMARTv2 resource-level scopes

*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={}  
  or GET Observation?patient={}&category=core-characteristics,social-history,vital-signs,laboratory (if category is required)*

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

* *GET Observation*
* *GET Encounter?patient={}*

**RESULTS**

1. [SUCCESS] SMART Client requests scopes and SMART Servers grant scopes at the resource level – “Specifically we'll test support for patient/Observation.rs”
2. [SUCCESS] GET Observation?patient={}
3. [SUCCESS] GET Observation?patient={}&category=vital-signs
4. [SUCCESS] GET Observation?patient={}&category=vital-signs,laboratory
5. [FAILED] GET Observation – we got different observations. Seems that the observations are from different patients (Patient/8c8bea6b-3352-4331-b5e8-544c0fef254a and Patient/fa064acf-b7f1-4279-83d3-7a94686da7ba)
6. [SUCCESS] With GET <https://smart.argo.run/v/r4/fhir/Encounter> and we got HTTP 403 Forbidden. Also tested GET https://smart.argo.run/v/r4/fhir/Encounter?subject=Patient%2F84451841-21bf-4754-9eb5-99d508c4282f

## Scenario 3: Like Scenario 2 but with Vital Signs Observations (SMARTv2 category-level scopes)

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

1. *patient/Observation.rs?category=http://terminology.hl7.org/CodeSystem/observation-category|vital-signs*
2. *patient/Observation.crs?category=http://terminology.hl7.org/CodeSystem/observation-category|vital-signs*

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

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

*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*

**Results**

1. [SUCCESS**?**] Requesting scope patient/Observation.rs?category=http://terminology.hl7.org/CodeSystem/observation-category|vital-signs. See bullet 4. We are not sure if the server granted the requested scope.
2. [~~FAILED~~] getting observations <https://smart.argo.run/v/r4/fhir/Observation?subject=Patient%2F476d3544-6370-479f-9ac9-c43434aae12f&category=vital-signs>. Got “HTTP 403 Forbidden”. **Should it work if not informing category=http://terminology.hl7.org/CodeSystem/observation-category?**
3. [FAILED] <https://smart.argo.run/v/r4/fhir/Observation?subject=Patient%2Fd00b766a-f5fa-4730-a64d-e8574994f460&category=http%3A%2F%2Fterminology.hl7.org%2FCodeSystem%2Fobservation-category%7Cvital-signs>. Got “HTTP 403 Forbidden”.
4. [FAILED?] [TEST NOT PLANNED IN CONNECTATHON] We requested an invalid scope ([http://terminology.hl7.org/CodeSystem/observation-category|**vitl-signs**](http://terminology.hl7.org/CodeSystem/observation-category|vitl-signs)) and got a token. Is this the expected behavior?

Here is the body of the response (**200 OK**)

{

"need\_patient\_banner":false,

"smart\_style\_url":"https://smart.argo.run//smart-style.json",

"val\_method":"cc-asym",

"jwks":"{\"keys\":[{\"kty\":\"RSA\",\"e\":\"AQAB\",\"use\":\"sig\",\"kid\":\"cde745f7-dbfd-4a9f-b110-331abcb85778\",\"alg\":\"RS384\",\"n\":\"gfpwdxF2XuUiC5f82xD41yogS3ZKaQOqMlT747QzllarJPSTMYGcJAOzfjo1eKruwxpBKCS1Yl60TUQdHu52by-oEavyR353pThoZ5pRAeRBeWf5MDPmfkNzj0EV6CnFysF3eRQm8jkoKCm3iSUQH\_TWnURNoonzOMBicimbr6G4qXBBGutigKiTMFLgdtdGX3siJaPjRfeyP8cjEie0bXCNHZsc50UVJ\_3eCl87hiL7U8J7bxEindjTMY5dl-DogRUUARQoKX7RIboT4rVdiWdg5gmqn5InC3NRiwRK1KwwGa3oXMyVLgh6oVOwdaHRMehw5FPniEagA2RbVuN2Jw\"}]}",

"patient":"d00b766a-f5fa-4730-a64d-e8574994f460",

"token\_type":"bearer",

"scope":"launch/patient patient/Observation.rs?category=http://terminology.hl7.org/CodeSystem/observation-category|**vitl-signs**",

"client\_id":"427c99e5-e6a0-49b1-8201-4f18dc75ef1d",

"expires\_in":3600, "access\_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..pCNcK-NxcZZv6qVeK2h5iA2LA7ckrxZHx3LLZRDiItw"

}

1. [SUCCESS] GET Observation?patient={}. We got “HTTP 403 Forbidden”. But everything seems to be forbidden – See 2 and 3 :)
2. [SUCCESS] GET Observation?patient={}&category=laboratory. We got “HTTP 403 Forbidden”. But everything seems to be forbidden – See 2 and 3 :)

## Scenario 4: Like Scenario 2 but with POST-based "authorize" API call

 Not tested

## Scenario 5: Like Scenario 2 but with PKCE enabled

## 

**Results**

All our previous tests were done with PKCE.

## Scenario 6: Token Introspection

*SMART Client requests token introspection and server responds with details including in-context patient:*

*GET .well-known/smart-configuration  
POST $introspection-url  
token=$token*

**Results**

Functionality not implemented in our app. So, not tested.