FHIRcast


Intro

January WGM, 2019
Timeline

- Starting testing a context sync spec based on FHIR Subscriptions in Sept, 2017.
- Following significant implementer feedback in Jan, 2018, refactored the spec to move away from FHIR Subscriptions and instead used a common rest-hook communication model.
- Tested successfully in May and September, 2018 at FHIR Connectathons.
- Have been adding new features based upon feedback.
- Tutorial at Boston DevDays in June.
- Dedicated time this summer to fixing bugs and improving the downloadable sandbox.
- Great open source community contributions.
- Connectathon track this weekend.
SMART App Launch protocol

• HL7 standard
  • Authorization (patient, provider, backend system)
  • OAuth2 scopes
  • Access to FHIR server

• Standalone launch
  • User authenticates

• “EHR Launch”
  • Single sign-on.

www.hl7.org/fhir/smart-app-launch/
SMART App launched from the EHR

- Embedded
  - iframe or embedded browser
  - web app
- “Sidecar” or external
  - Browser external to EHR or native app (mobile or desktop)
- Multiple machines
  - Simultaneous: desktop EHR + mobile app
  - Sequential: EHR then mobile app

www.hl7.org/fhir/smart-app-launch/
EHR-hosted browser control

Embedded SMART app
App context synchronization

- Widely used in healthcare
- Typically proprietary
- Disparate implementations
- May only support desktop apps
- May require apps to be on same machine
FHIRcast: modern, simple app context sync

• Based on http, webhook, json
• Extends SMART on FHIR
• Doesn’t require context manager

http://fhircast.org/
FHIRcast: EHR launches SMART on FHIR App

SMART launch parameters include:
1) hub base url
2) session identifier

{
   "access_token": "i8hweunweunweofiwweoijewiwe",
   "token_type": "bearer",
   "expires_in": 3600,
   "patient": "123",
   "encounter": "456",
   "imagingstudy": "789",
   "cast-hub": "https://hub.example.com",
   "cast-session": "https://hub.example.com/7jaa86kgdudewiaq0wtu"
}
## FHIRcast: App subscribes to session

Subscriber HTTP POSTs to the hub base url to create subscription

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hub.callback</td>
<td>Subscriber’s callback url.</td>
</tr>
<tr>
<td>hub.mode</td>
<td>The literal string “subscribe”.</td>
</tr>
<tr>
<td>hub.topic</td>
<td>Uri of user’s session.</td>
</tr>
<tr>
<td>hub.secret</td>
<td>Unique secret string, used to verify subscription request.</td>
</tr>
<tr>
<td>hub.events</td>
<td>List of events to subscribe to.</td>
</tr>
<tr>
<td>hub.lease_seconds</td>
<td>Length of subscription in seconds.</td>
</tr>
</tbody>
</table>
FHIRcast: App subscribes to session

Subscriber HTTP POSTs to the hub base url to create subscription

POST https://hub.example.com
Host: hub.example.com
Authorization: Bearer i8hweunweunweofiwweoiwie
Content-Type: application/x-www-form-urlencoded
hub.callback=https%3A%2F%2Fapp.example.com%2Fsession%2Fcallback%2Fv7tfwuk17a&hub.mode=subscribe&hub.topic=https%3A%2F%2Fhub.example.com%2F7jaa86kgdudewiaq0wtu&hub.secret=shhh-this-is-a-secret&hub.events=patient-open-chart,patient-close-chart
**FHIRcast: Hub verifies callback url**

The hub performs verification of intent of the subscriber.

GET https://app.example.com/session/callback/v7tfwuk17a?hub.mode=subscribe&hub.topic=7jaa86kgdu
dewiaq0wtu&hub.events=patient-open-chart,patient-close-chart&hub.challenge=meu3we944ix80ox HTTP 1.1
Host: subscriber

The subscriber confirms.

HTTP/1.1 200 Success
Content-Type: text/html

meu3we944ix80ox
FHIRcast: Workflow event occurs and subscriber is notified
Hub POSTs event notification with relevant FHIR resources to subscriber.

POST https://app.example.com/session/callback/v7tfwuk17a HTTP/1.1
Host: subscriber
X-Hub-Signature: sha256=dce85dc8dfde2426079063ad413268ac72dcf845f9f923193285e693be6ff3ae

{
    "timestamp": "2018-01-08T01:37:05.14",
    "id": "q9v3jubddqt63n1",
    "event": {
        "hub.topic": "https://hub.example.com/7ja86kgdudewiaq0wtu",
        "hub.event": "open-patient-chart",
        "context": [
            {
                "key": "patient",
                // Patient FHIR Resource
            }
        ]
    }
}
**FHIRcast: App unsubscribes from session**

Same as subscription request, but for `hub.mode='unsubscribe'`

POST https://hub.example.com
Host: hub
Authorization: Bearer i8hweunweunweofiwweoijewiwe
Content-Type: application/x-www-form-urlencoded

hub.callback=https%3A%2F%2Fapp.example.com%2Fsession%2Fcallback%2Fv7tfwuk17a&hub.mode=unsubscribe&hub.topic=https%3A%2F%2Fhub.example.com%2F7jaa86kgdudewiaq0wtu&hub.secret=shhh-this-is-a-secret&hub.events=patient-open-chart,patient-close-chart
Recent updates

• Add capability for the client to query for current context
• **Informative description of security considerations for FHIRcast**
  Exploring optional support for websockets in addition to web hooks
• **Informative description of synchronization failure considerations**
• **implementers**
Get involved

• Read the spec
  • http://fhircast.org

• Download, test, contribute to the sandbox
  • https://github.com/fhircast/sandbox

• Log issues, participate via github
  • https://github.com/fhircast/docs

• Ask questions, learn, chat
  • https://chat.fhir.org/#narrow/stream/118-FHIRcast
Connectathon Update

January, 2019
San Antonio, TX

Nuance, Sectra, Epic, Philips, Siemens Healthineers
Connectathon report

- Testing with the sandboxes
- Philips published their prototypes
  - https://github.com/PhilipsOnFhir/fhir-cast
- Design discussions:
  - websockets - issue 33
  - Retrieve context without event - issue 28
  - Non-OAuth2 launching and session id generation
  - websockets vs SSE vs HTML5 Web Messaging
- Implementers want a standard, and the standard needs implementers
FHIRcast report

Two sandboxes - js, .Net, open source implementations, experimental websockets

Ballot in May
Next steps

1) Experimental websocket support in specification.
2) Polish WebSub spec
3) Ballot