# Smart on FHIR

Federated Authentication / Authorization
FHIR and HTML5

# What is SMART on FHIR

- Collection of Standards,
- JS & Python API for FHIR, and
- Profiles for Standard FHIR Resource Support + Coding Systems

Standards

OAUTH2 + OpenID for Authentication and Authorization

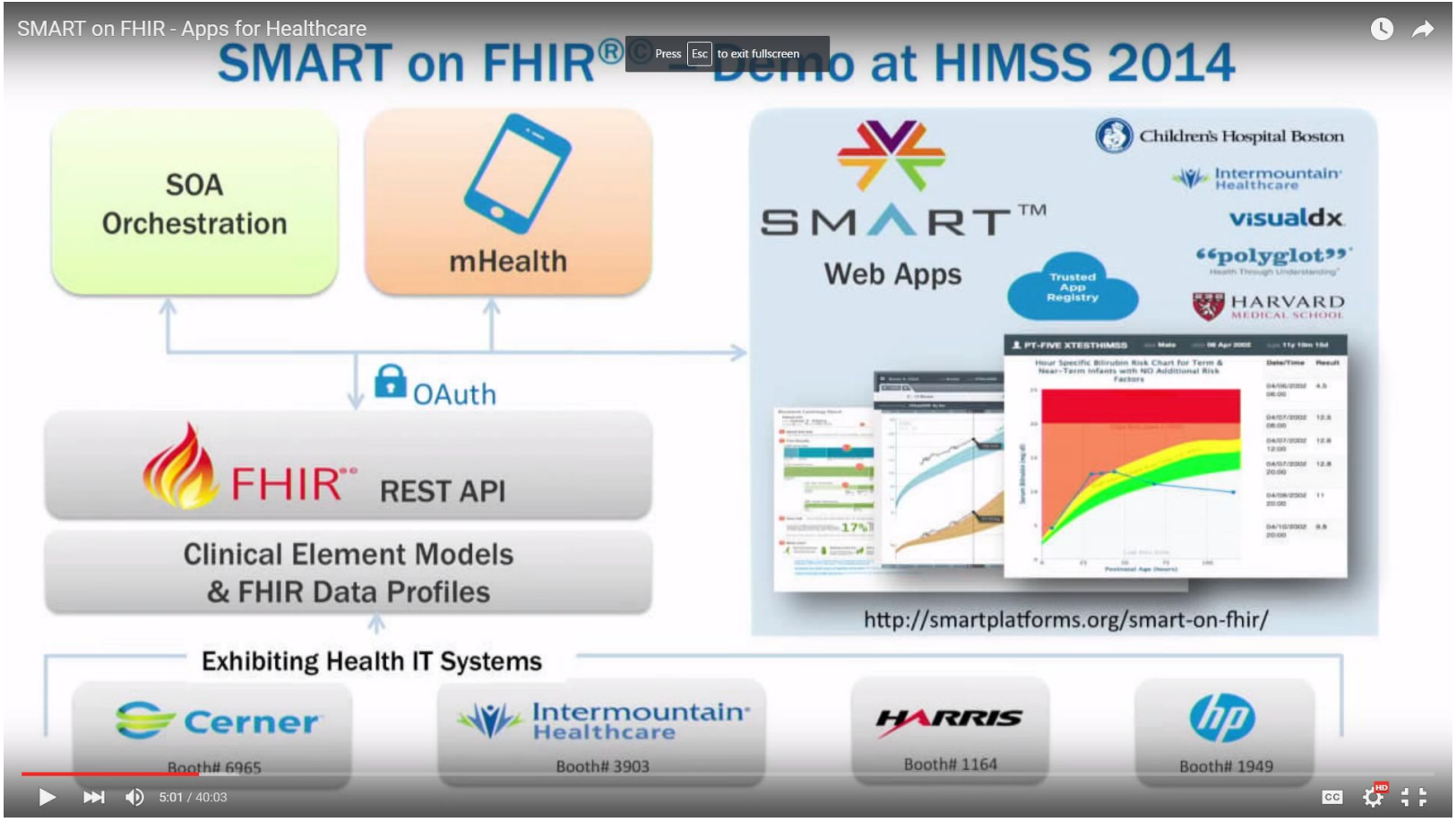
FHIR DSTU2

HTML5

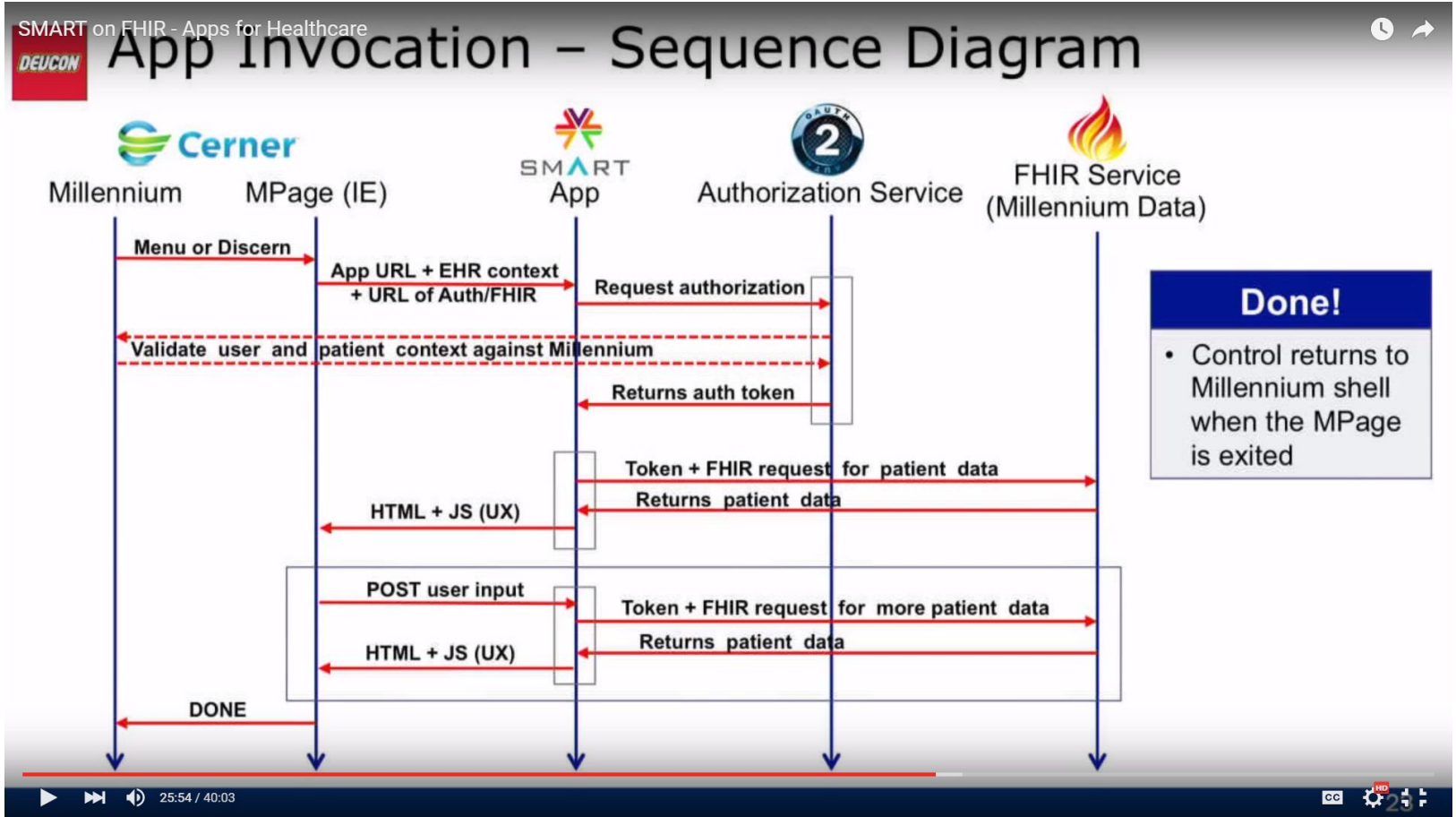
APIs for JS & Python (for Java use Hapi-FHIR)

Profiles

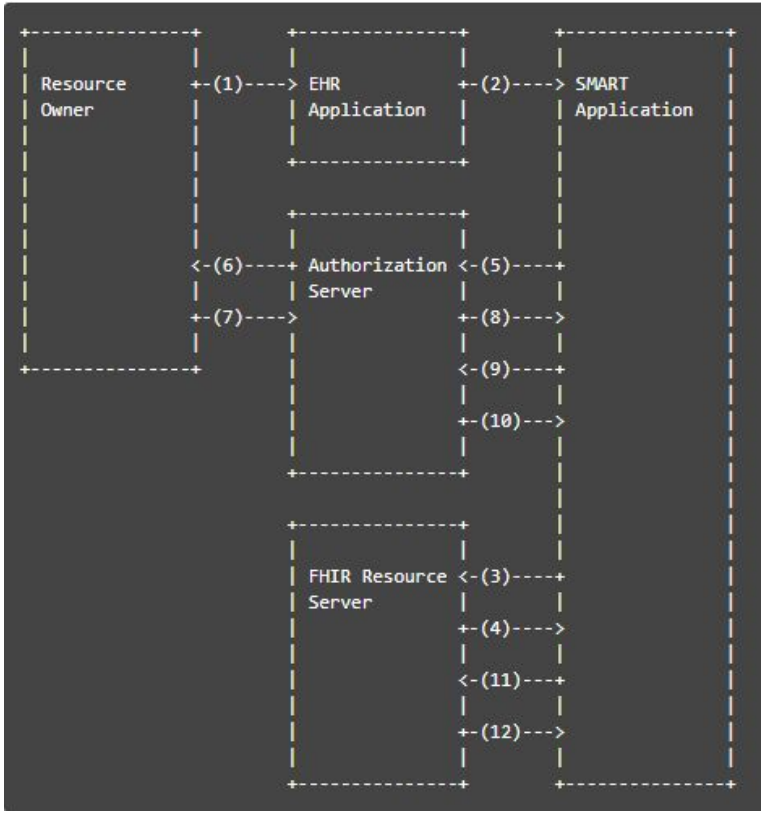
# Smart Arcitecture Overview



# OAUTH2


SMART on FHIR - Apps for Healthcare
App Invocation – Sequence Diagram
DEUCON
Cerner
Millennium
MPage (IE)
SMART
App
Authorization Service
FHIR Service
(Millennium Data)
Menu or Discern
App URL + EHR context
+ URL of Auth/FHIR
Request authorization
Validate user and patient context against Millennium
Returns auth token
Token + FHIR request for patient data
Returns patient data
HTML + JS (UX)
POST user input
Token + FHIR request for more patient data
Returns patient data
HTML + JS (UX)
DONE
Done!
• Control returns to Millennium shell when the MPage is exited
25:54 / 40:03

# OAuth2 Authentication




1. The end user selects to launch a SMART application from within an EHR application.
2. The EHR directs the user to a URI endpoint registered for the SMART application containing a reference to the current context information, and the location of the EHRs FHIR® API.
3. The SMART application performs discovery by requesting the FHIR® server's conformance statement.
4. The FHIR® server returns the conformance statement, which provides the needed endpoints for steps 5 and 9.
5. The SMART application creates an OAuth 2.0 authorization grant request, then directs the end user to the authorization server's authorization endpoint via a browser with said request. This request contains a request for the appropriate scopes necessary to access the FHIR® resource.
6. The authorization server interacts with the resource owner to verify identity or other information required by the authorization server.
7. The end user provides any information needed by the authorization server to proceed.
8. An authorization grant is sent via the OAuth 2.0 framework back to the SMART application.
9. The SMART application requests an access token using the authorization code.
10. The authorization server returns the access token.
11. The SMART application utilizes the access token to request a FHIR® resource.
12. The FHIR® resource server returns the desired resource.

# OAuth2 / OpenId

## Scopes and permissions: OAuth2

When an EHR user launches your app, you get a “launch request” notification. Just ask for the permissions you need using OAuth scopes like `patient/*.read` and once you’re authorized you’ll have an access token with the permissions you need – including access to clinical data and context like:

- which patient is in-context in the EHR
- which encounter is in-context in the EHR
- the physical location of the EHR user

## Simple sign-in: OpenID Connect

If your **app needs to authenticate the EHR end-user**, OpenID Connect is there to help. Just ask for one **additional scope** (`openid`) when you request authorization, and you’ll have access to a `UserInfo` **endpoint** that exposes **structure claims about the user**, including **name and NPI**.

## Lightweight UI integration: HTML5

Need to hook your app into an existing EHR user interface? SMART on FHIR allows web apps to run inside browser widgets or inline frames, so users can interact without leaving the EHR environment. Of course, native and mobile apps are supported too – so you can choose the level of integration that makes sense for you.

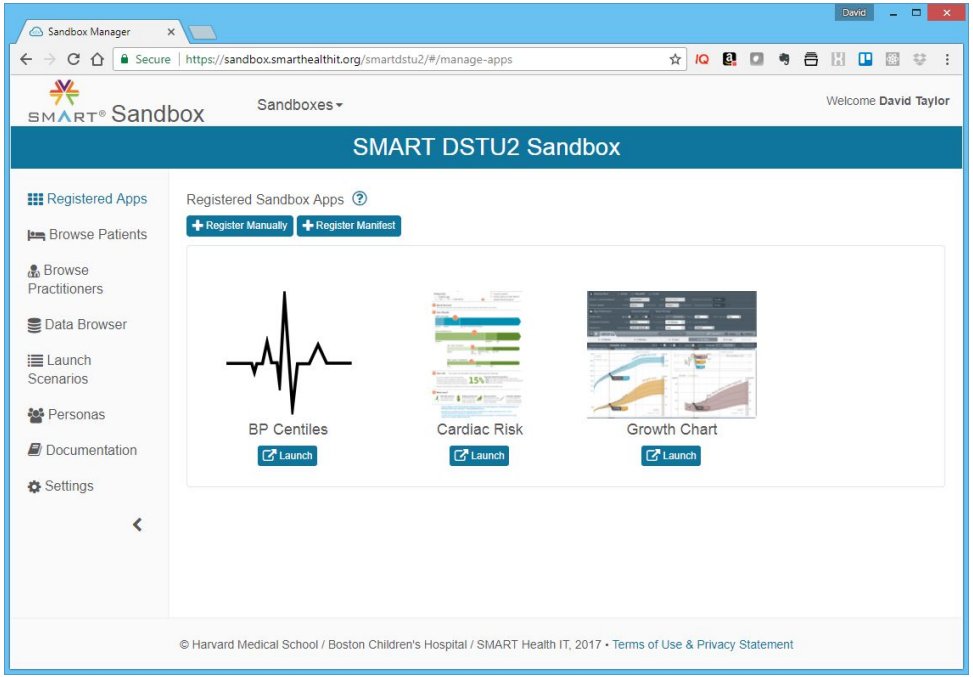
# SmartHealthIT SMART Tutorials

http://docs.smarthealthit.org/tutorials/

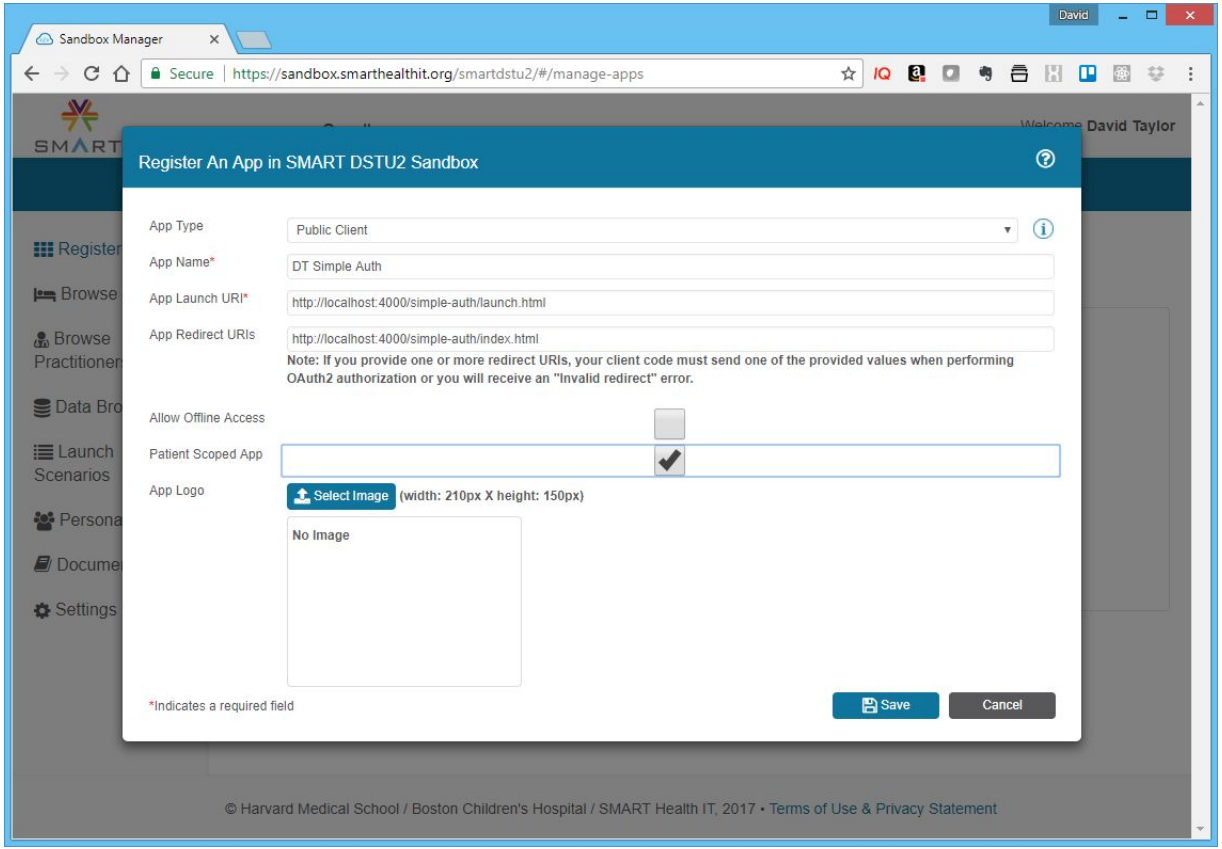
http://docs.smarthealthit.org/tutorials/javascript/

http://docs.smarthealthit.org/tutorials/authorization/

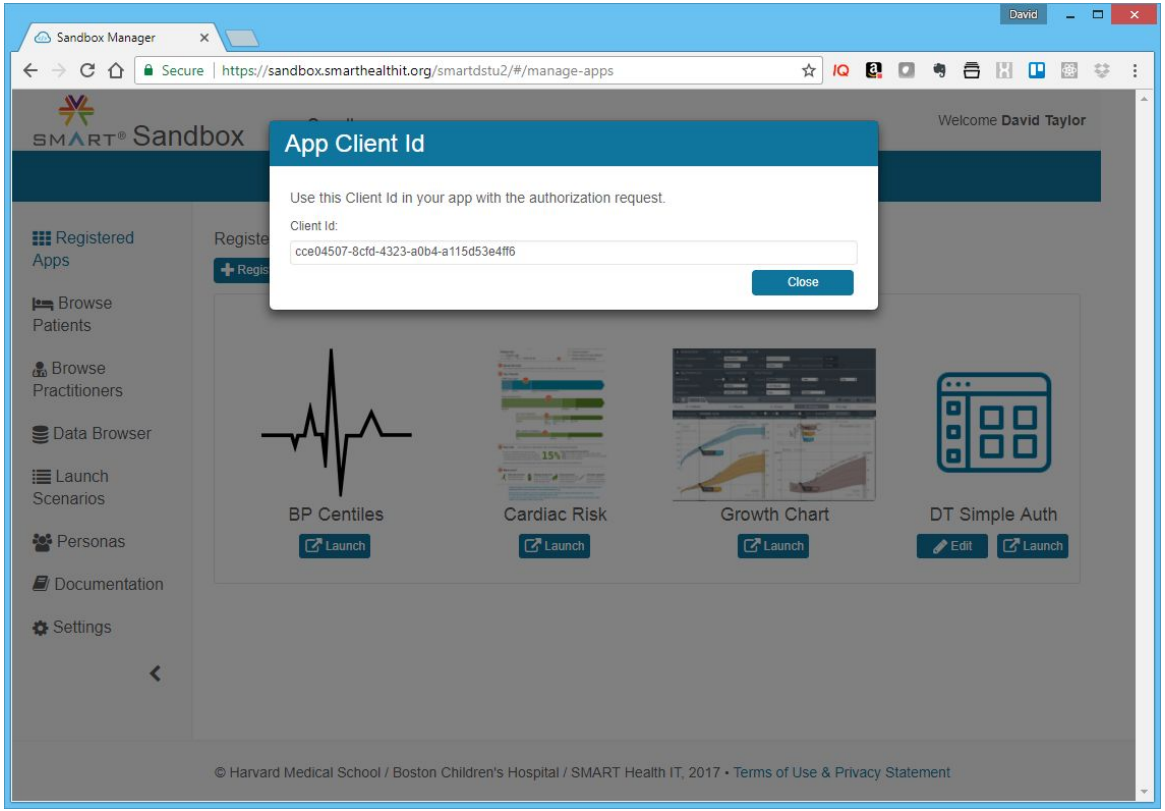
# Login to Smarthealthit.org SMART FHIR Sandbox



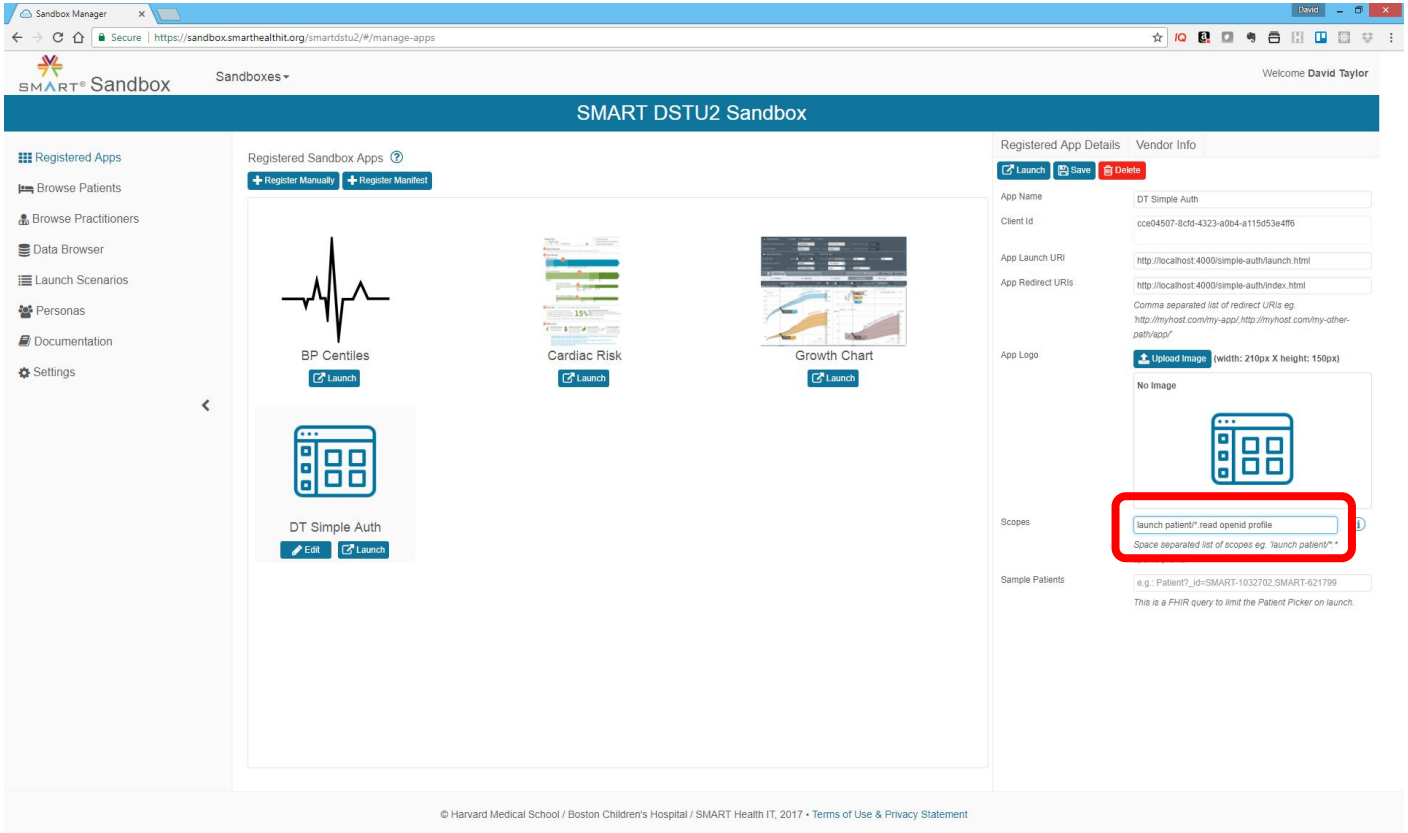
# Add App “Register Manually”



# Copy Client Id



# Modify App to fix Patient Scope (Patient/*.read)

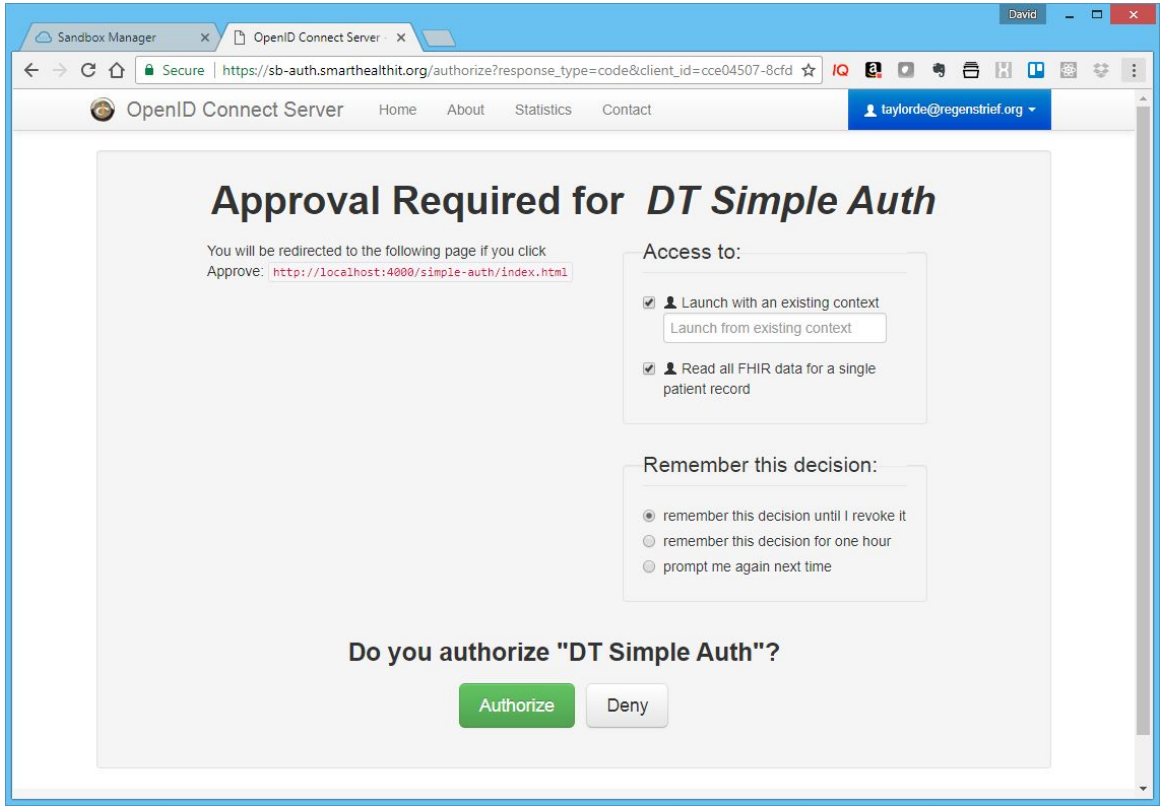


# Install http-server via npm and start Sample files

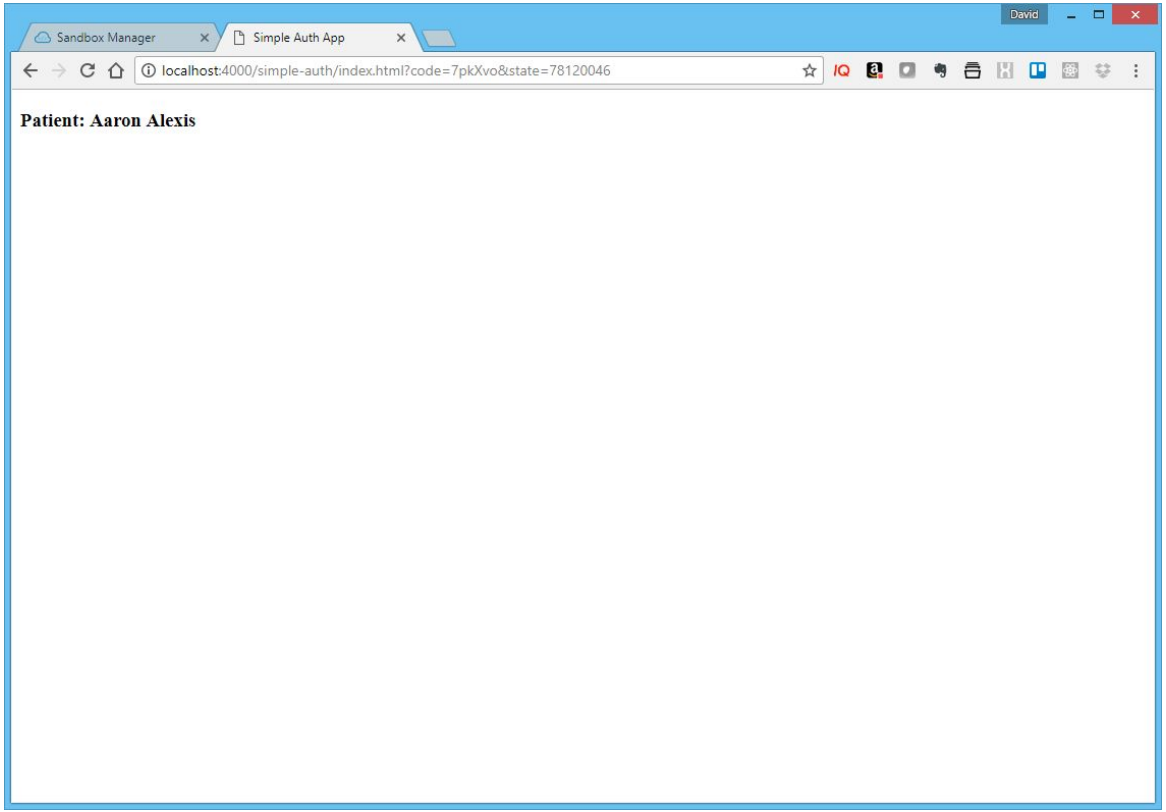
- http://docs.smarthealthit.org/tutorials/authorization/
- Install http-server via npm
  - npm install http-server -g
- Create a folder called simple-auth
- Create 2 files in that folder
  - launch.html
    - (Remember to update the client_id var)
  - Index.html
- Start http-server in folder above simple-auth
  - `http-server -p 4000`
- Turn on Dev-Tools

- Launch App from Sandbox

# Choose Patient from List - Prompted to Authorize



# Patient Displayed



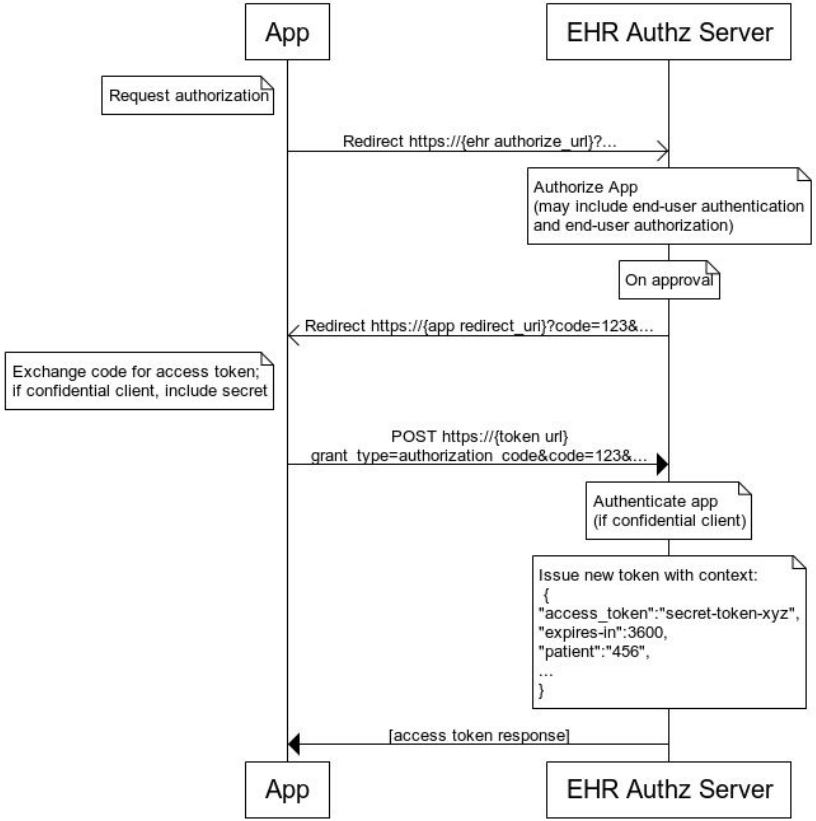
# Server Calls from Sandbox to our Smart Server

GET
/simple-auth/launch.html?**iss=https%3A%2F%2Fsb-fhir-dstu2.smarthealthit.org%2Fsmartdstu2%2Fdata&launch=gEVUxa**

**Fhir_base_url = iss =** https://sb-fhir-dstu2.smarthealthit.org/**smartdstu2/data**
Identifies the EHR's FHIR endpoint, which the app can use to obtain additional details about the EHR, including its authorization URL.

**launch = gEVUxa**
Opaque identifier for this specific launch, and any EHR context associated with it. This parameter must be communicated back to the EHR at authorization time by passing along a `launch=123` parameter.

# Redirect back to index.html post Authent/Author




GET
/simple-auth/index.html?**code=E0fVo3&state=29839352**

**code=E0fVo3**
The authorization code generated by the authorization server. The authorization code *must* expire shortly after it is issued to mitigate the risk of leaks.

**state=29839352**
The exact value received from the client.

# Client calls back to Sandbox/FHIR server

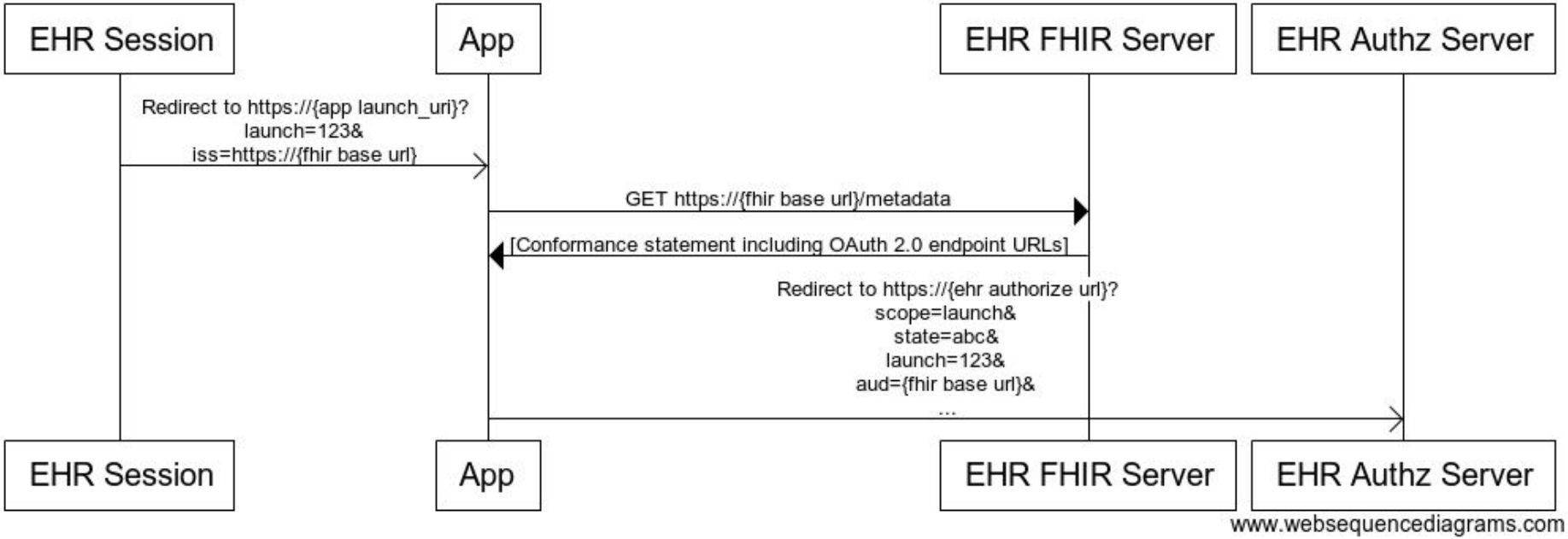
GET https://sandbox.smarthealthit.org/REST/**launchScenario?appId=772**

GET https://sandbox.smarthealthit.org/REST/**app/772**

GET https://sb-fhir-dstu2.smarthealthit.org/**smartdstu2/data/Patient?_sort:asc=family&_sort:asc=given&name=&_count=10**

# SMART App Authorization Guide

http://docs.smarthealthit.org/authorization/



# Smart on FHIR Profiles (Project Argonaut)

http://www.fhir.org/guides/argonaut/r2/

http://www.fhir.org/guides/argonaut/r2/profiles.html

Provides Standards for Structure of Resources and Standard Coding Systems

FHIR Coding Systems

https://www.hl7.org/fhir/dstu2/terminologies-systems.html

Argonaut AllergyIntolerance Profile
Argonaut CarePlan Profile
Argonaut CareTeam Profile
Argonaut Condition Profile
Argonaut Device Profile
Argonaut DiagnosticReport Profile
Argonaut DocumentReference Profile
Argonaut Goal Profile
Argonaut Immunization Profile
Argonaut Medication Profile
Argonaut MedicationOrder Profile
Argonaut MedicationStatement Profile
Argonaut Observation Results Profile
Argonaut Patient Profile
Argonaut Procedure Profile
Argonaut Smoking Status Observation Profile
Argonaut Vital Signs Observation Profile

# Useful Links - FHIR

http://hl7.org/fhir/index.html

http://jamesagnew.github.io/hapi-fhir/

https://open.epic.com/Interface/FHIR

https://open.epic.com/AppExchange/Sandbox

http://fhir.cerner.com/smart/

http://fhir.cerner.com/authorization/authorization-specification/

https://sandbox.smarthealthit.org/

# Useful Links - Smart on FHIR + Profiles

http://docs.smarthealthit.org/

http://docs.smarthealthit.org/clients/javascript/

https://github.com/smart-on-fhir/client-js

http://docs.smarthealthit.org/tutorials/testing/

http://docs.smarthealthit.org/authorization/scopes-and-launch-context/

http://docs.smarthealthit.org/profiles/

http://argonautwiki.hl7.org/index.php?title=Main_Page

http://hl7.org/fhir/daf/daf.html

https://gallery.smarthealthit.org/

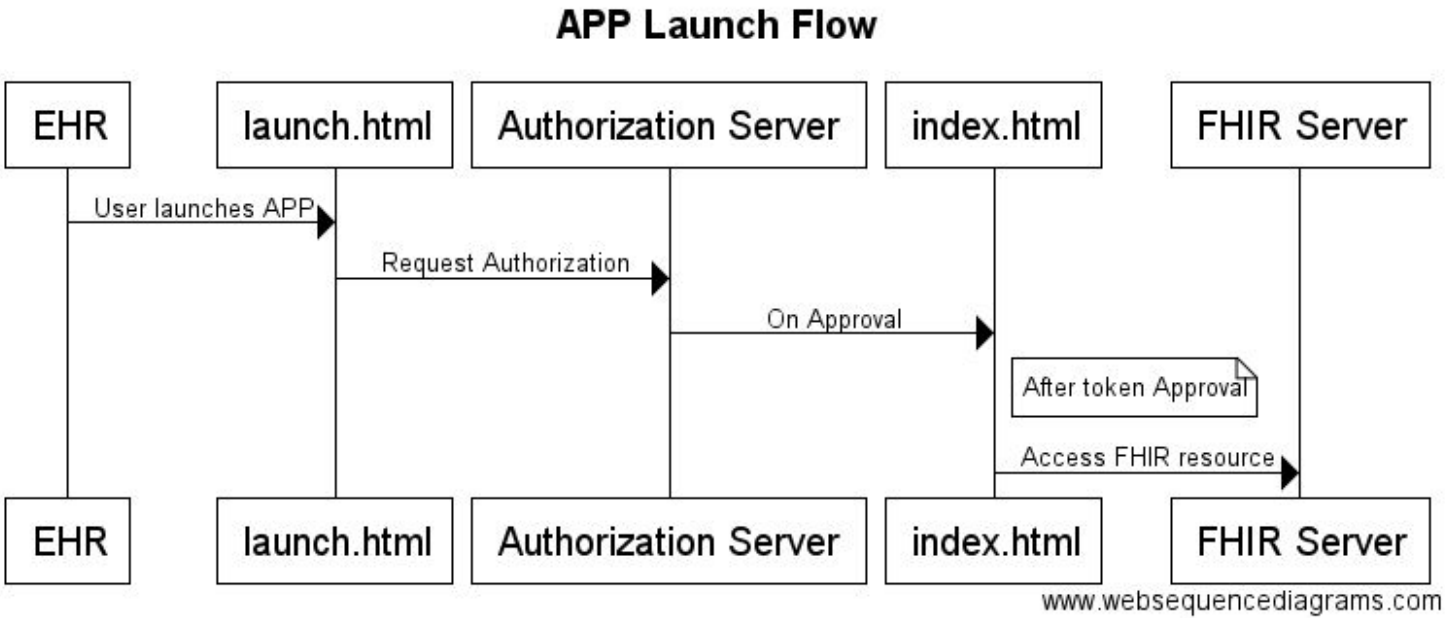
# Next Meetings

- 9/6 - Cerner Tutorials
- 10/4 - Epic Tutorials
- 11/1 - Java Spring OAuth2 / OpenId Security + Hapi FHIR Server

Thank you for coming tonight!

# Cerner Smart FHIR Tutorial

http://engineering.cerner.com/smart-on-fhir-tutorial/#github-pages



# FHIR OAuth2 Authorize

```
FHIR.oauth2.authorize({
        'client_id': '<enter your client id here>',
        'scope':  'patient/Patient.read patient/Observation.read launch online_access openid profile'
      });
);
```

**EHR Launch Request**

In the EHR application launch flow, the end user chooses to “launch” a SMART application from the EHR. To receive such a launch, the SMART application implements an endpoint at a specific URI that accepts the following query parameters:

- **iss**
  Identifies the **EHR’s FHIR® endpoint**, which the app can use to obtain additional details about the EHR, including its authorization URL.
- **launch**
  An **opaque identifier for this specific launch**, and any EHR context associated with it.

# OAuth2 Sample Messages

https://example.org/launch?iss=https%3A%2F%2Fehr%2FFHIR&launch=ef1e6860-db06-4572-b311-02881d01d03d

# Sample Sandbox Queries - Open Epic Patient

Open.epic

https://open-ic.epic.com/FHIR/api/FHIR/DSTU2/Patient?given=Jason&family=Argonaut

Patients:

- Jessica Argonaut
- Flapjacks Ragsdale
- Pancakes Ragsdale
- Waffles Ragsdale
- Bacon Ragsdale
- Emily Williams (three of them!)
- James Kirk

# Sample Sandbox Queries - Open Epic Allergy

https://open-ic.epic.com/FHIR/api/FHIR/DSTU2/AllergyIntolerance?patient=Tbt3KuCY0B5PSrJvCu2j-PlK.aiHsu2xUjUM8bWpetXoB

# Videos Presentations

Cerner presentation: SMART on FHIR - Apps for Healthcare

https://www.youtube.com/watch?v=BbBZbo2fMus&list=PLda47xJVlvw4gndPMTrQQkI2CCt0Xi1R2

SMART on FHIR Presentation for CajunCodeFest 4.0

https://www.youtube.com/watch?v=sb7RzWpW_nE

# Sample Sandbox Queries - RMRS Fhir

http://10.234.33.43/fhir/Patient?_id=1+620&_include=DiagnosticReport:result&_revinclude=DiagnosticReport:subject&_revinclude=QuestionnaireResponse:subject&_revinclude=Condition:patient&_revinclude=AllergyIntolerance:patient&_revinclude=Immunization:patient

# Project Organization

Base Project Template

- Maven
- Spring-Boot
- Hapi-FHIR
- Spring-JPA
- Embedded Jetty server

Repository: https://tools.regenstrief.org/stash/projects/CON

Accounts have been / are being setup

# Sample Sandbox Queries - RMRS Fhir

http://10.234.33.43/fhir/MedicationOrder?patient=1+620&_include=MedicationOrder:prescriber

http://10.234.33.43/fhir/MedicationDispense?patient=1+620

# SMART App for Connectathon

- Support App Launch URL
- Support retrieving Patient Data using FHIR (Hapi-FHIR, JS, Python)
- Display data / provide interaction via HTML5 Web Page(s)
    - Angular, JQuery, etc.

# Testing / Sandbox Support

http://10.234.33.43/fhir/

RMRS Resource Provider targeting a test RMRS database NON-PHI Test Data

Sample URLs above

https://open-ic.epic.com/FHIR/api/FHIR/DSTU2/

Open.epic sandbox

https://fhir-dstu2.smarthealthit.org/#/ui/select-patient

Smart FHIR Sandbox (working on a local targeting RMRS - not ready yet)