FHIR BULK DATA API

Dan Gottlieb
HIMSS 2018
Today

Three recent examples:

- Large AMC syncing progress notes from a third party clinic into EHR
- Integration population health system with EHR system
- Machine learning startup obtaining training data from cloud EHR
Today

Three recent examples:

- Large AMC syncing progress notes from a third party clinic into EHR
- Integration population health system with EHR system
- Machine learning startup obtaining training data from cloud EHR

Other common use cases:

- Payer database to assess care quality
- Claims in EHR to provide comprehensive view
- Internal clinical data warehouse for study cohort identification
- Reportable disease submission or other registry
Sharing bulk data is cumbersome

Healthcare organizations often use CSV EHR and data warehouse extracts to share clinical data (or don’t share it at all)

- **Proprietary data model**: overburdens IT organizations and data analysts with manual and repetitive work to map the data
- **Proprietary API (or manual)**: Data extraction scripts need to be built for each system and logistics like firewall support need to be configured each time
- **FHIR API (programming interface)** is great for small queries, but inefficient for this type of large queries
  - Synchronous
  - Paged
Let’s enhance FHIR to support bulk data access

- **Standard data model** to simplify data parsing and mapping
- **Standard API** across systems to initiate the data extracts
- **Standard security model** through SMART backend services authorization specification
Pipelines can support many scenarios

Create a deidentified view for researchers:

- EHR
  - Bulk Data API
  - Deidentification Transformation
  - Bulk Data API
  - ETL
  - Research DB

Share with a partner:

- EHR
  - Bulk Data API
  - Re-code Meds to RxNorm
  - Bulk Data API
  - SFTP
  - Data Load
  - Hadoop HDFS

Share with several partners:

- EHR
  - Bulk Data API
  - Encryption Transformation
  - Bulk Data API
  - AWS Upload
  - S3 Signed URLs
Design Goals

- Focus on enabling automated communication between backend services and EHRs/clinical systems
- Use mature, stable technologies wherever possible
- Small API surface area
  - Limit number of query parameters
  - Limit number of serialization formats
- Reuse as much of existing FHIR semantics as possible
  - Data models
  - API format and data types
  - Implementation guide structure
- Use existing standards based authentication and authorization
  - Base on widely used OAuth (SMART) standard
- Structure for efficiently generating and loading large datasets
  - Asynchronous operation
  - One data type per file
  - Streaming data
Focused Scope

Out of scope for initial version:

- Legal framework for sharing data between partners - BAAs, SLAs, DUAs continue to be negotiated and completed out-of-band
- Real-time data (although data loaded through bulk data can be supplemented at with synchronous FHIR REST API calls)
- Data transformation - different step of the ETL process
- Patient matching (although, it’s possible to include identifiers like subscriber number in FHIR resources)
Architecture
Kickoff Request

Backend Service

Kickoff Request

Bulk Data Server
Kick Off Request

- FHIR Operation for all data on all patients
  
  \[[FHIR \ Server \ Base]/Patient/$export\]

- FHIR Operation for all data on a group of patients:
  
  \[[FHIR \ Server \ Base]/Group/[group \ id]/$export\]

- Asynchronous requests with status polling
  
  Prefer: respond-async
## Query Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_outputFormat</code></td>
<td>The format for the generated bulk data files (currently, only <strong>ndjson</strong> is supported)</td>
</tr>
<tr>
<td><code>_since</code></td>
<td>FHIR resource modified date (FHIR instant timestamp)</td>
</tr>
<tr>
<td><code>_type</code></td>
<td>Comma delimited list of FHIR resource types</td>
</tr>
<tr>
<td>[group id]</td>
<td>Predefined set of patients (research cohort, plan members, employer)</td>
</tr>
</tbody>
</table>
Kickoff Response

Kickoff Request

Content Location

Backend Service

Bulk Data Server
Kick Off Response

Status: 202 Accepted
Content-Location: [URL for status or deleting request]
Status Request #1

- Backend Service
- Kickoff Request
- Content Location
- GET Content Location
- Bulk Data Server
Status Response

Status: 202 Accepted
X-Progress: “50% complete”
Retry-After: 120
Status Request #2

Kickoff Request

Content Location

GET Content Location

File Generation Status

GET Content Location
Status Response Header

Status: 200 OK
Expires: Mon, 12 Mar 2018 23:59:59 GMT
Status Response Body

{  
  "transactionTime": "[instant]",
  "request": "[base]/Patient/$export?_type=Patient,Observation",
  "secure": true,
  "output": [{
    "type": "Patient",
    "url": "http://serverpath2/patient_file_1.ndjson"
  },{
    "type": "Patient",
    "url": "http://serverpath2/patient_file_2.ndjson"
  },{
    "type": "Observation",
    "url": "http://serverpath2/observation_file_1.ndjson"
  }]
}
File Request

- Backend Service
- Kickoff Request
- Content Location
- GET Content Location
- File Generation Status
- GET Content Location
- File Links
- GET File (e.g. 0001.Observation.ndjson)
- FHIR Resources File

Bulk Data Server
FHIR Resources

Data models representing discrete clinical and administrative units (patient, practitioner, allergy, medication order, etc.)

- Currently around 100 have been defined
- Can reference other resources by their URL
- Don’t include the kitchen sink

“We only include data elements if we are confident that most normal implementations using that resource will make use of the element”
– Grahame Grieve (FHIR Product Director)

- But, support extensions
- MU3 Common Clinical Dataset defines subset
[{
  "id": "06eb35fc-09e6-48 ... "given": ["Lucille"],"family": "Bluth"}
},
{
  "id": "cf53f382-6eb6-4f ... "given": ["George", "Oscar"],"family": "Bluth","suffix": ["Senior"]}
},
{
  "id": "406a9c3e-50f9-4c ... "given": ["Michael"],"family": "Bluth"}
]
SMART Authorization

- Out-of-band app registration (can use Dynamic Client Registration or portal)
- Token requests signed with private key
- System level scope (parallels SMART “user” scope)
  
  \[
  \text{system/}[\text{resourceType}].\text{read}
  \]
  
- Short-lived access tokens
Security Flow (SMART Backend Services)

Backend Service Admin

- Configure Public Key and other OAuth settings
- OAuth Client Id

Bulk Data Server

- Signed Token Request
- Short Lived Access Token
Tools and Resources
SMART Reference Implementation
SMART Sample GUI Client

FHIR Bulk Downloader

Resources to Download
- Allergy/Intolerance (30)
- DiagnosticReport (375)
- MedicationRequest (337)
- Procedure (732)
- CarePlan (239)
- Encounter (1,369)
- Observation (5,157)
- Claim (1,706)
- Goal (200)
- Organization (131)
- Condition (431)
- Immunization (726)
- Patient (100)

Patients Group
No Group (include all the patients)

Filter by Modification Date/Time
No Time Filter (include everything)

Download Link
http://localhost:9443/eyJlcml0OiIsLCJwYXJlcjoxMDAwMCwiZHVyY2thcGxGx0iLCJ0aW1lMS5tZXh0LWNsdXN0cyIsIFdpZHRoZ2V0cyIsIFdpZHRoaWJvYXQh

Files to Download
- 1.AllergyIntolerance.ndjson
- 1.DiagnosticReport.ndjson
- 1.MedicationRequest.ndjson
- 1.Procedure.ndjson
- 1.CarePlan.ndjson
- 1.Encounter.ndjson
- 1.Observation.ndjson
- 1.Claim.ndjson
- 1.Goal.ndjson
- 1.Organization.ndjson
- 1.Condition.ndjson
- 1.Immunization.ndjson
- 1.Patient.ndjson

Prepare Download!
SMART Sample Command Line Client

Waiting for the server to generate the files...
100% ⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬤⬜
Next Steps
## Timeline

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2017</td>
<td>ONC Meeting on Bulk Data</td>
</tr>
<tr>
<td>January 2018</td>
<td>SMART Bulk Data Reference Implementation Server</td>
</tr>
<tr>
<td></td>
<td>Initial Connectathon for interested parties to start experimenting with proposed technical specification</td>
</tr>
<tr>
<td>2018</td>
<td>FHIR Connectathon events to refine and develop the technical specification</td>
</tr>
<tr>
<td></td>
<td>Argonaut Project to bring together pilot implementations</td>
</tr>
<tr>
<td></td>
<td>Publish Draft Implementation Guide</td>
</tr>
<tr>
<td>2019</td>
<td>Publish version 1.0 of FHIR Implementation Guide For Bulk Data</td>
</tr>
</tbody>
</table>
Open Questions

- How do the APIs perform for real-world use cases?
- Are other forms of bulk data APIs required?
- Is ndjson sufficient or do we need other export formats (e.g. Apache Parquet)?
- Implementation details
Potential for Community Ecosystem

- Open source implementations of bulk data server and backend services client
- Open source test suite providing feedback on implementations (including error conditions and any optional bells/whistles)
- Open source reference implementation of components like de-identification that can act as nodes in an export pipeline
Links

- Draft Bulk Data Implementation Guide:

- Authentication approach with SMART backend services specification:
  [http://docs.smarthealthit.org/authorization/backend-services/](http://docs.smarthealthit.org/authorization/backend-services/)

- SMART Server Reference Implementation:
  [https://bulk-data.smarthealthit.org](https://bulk-data.smarthealthit.org)

- SMART Client Reference Implementation:

- Other implementations:

- Discussion Group:
  [https://chat.fhir.org/#narrow/stream/bulk.20data](https://chat.fhir.org/#narrow/stream/bulk.20data)