NextGen® Office Bulk FHIR API Developer Guide

for NextGen® Office

Prepared by NextGen R&D
Feb 20th, 2025

Contents

Overview

The NextGen® Office Bulk Data API is an API based on the standards for FHIR Bulk Data Access (Flat FHIR) (v1.0.1: STU 1) based on FHIR R4.

The API provides the ability to read/download resources in bulk for the entire practice or a subset of the practice using the FHIR standard.

JWK Setup

Authentication for Bulk FHIR API utilizes the methods detailed in the SMART Backend Services Authorization Guide.

Step 1: Create a JWK Keyset according to the following requirements and host the public key as your JWK URL.

Create a JWK keyset (using OpenSSL, etc.) with one or more keys, with the following attributes for each key:

- Encryption Algorithm: RS384
- Key Use: Signature
- Key Size: 3072 or 4096 bits
- Key ID MUST be a GUID (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx)

Secure the private key components. You will use both the public and private keys to sign the required JWT in Step 3.

Host the public keyset JSON at a publicly accessible URL – preferably on your organization's domain – such that the JWK keyset JSON is an array of keys that adheres to the following schema (only the "kid" and "n" values will differ per key).

Share the publicly accessible URL with the Nextgen Support team to be configured with the registered Bulk Client.

This is an implementation of the "URL to JWK Set" technique described in 4.1 of the SMART Backend Services Auth Guide linked above.

The following example contains an array of 2 keys; your JWK must contain an array with at least one key but can contain as many as you wish to use. The use of multiple keys is recommended for security (rotating through the keyset each time you sign a JWT).

Example Schema for the JWK JSON file you must host as your JWK URL
(e.g. https://example.domain.com/path/JWKS.json)

```
{
    "keys": [
    {
            "kty": "RSA",
            "alg":"RS384",
            "x": "<string>", //Modulus value for
            Key 1
            "y": "<string>",
            "key_ops":
            ["verify"],
            "use": "sig",
            "ext": true,
            "kid": "<guid>" //Key ID for Key 1 – this value MUST be a guid. },
    {
            "kty": "RSA",
            "alg": "RS384",
            "n": "<string>", //Modulus value for
            Key 2
             "e": "AQAB",
            "use": "sig",
            "key_ops":
            ["verify"],
            "ext": true,
            "kid": "<guid>" //Key ID for Key 2 – this value MUST be a guid.
            }
    ]
}
```

Bulk FHIR Authentication

Step 2: Create & Sign client_assertion JWT, then pass the signed JWT with other required parameters to the token endpoint to obtain an access_token for subsequent use when authorizing Bulk FHIR endpoint requests.

Construct and sign a JWT that meets the following requirements:

JWT Headers:

```
{
   "kid": "<guid>",
   "typ": "JWT",
   "alg": "RS384"
}
```

JWT Payload:

```
{
   "iss": "YOUR_CLIENT_ID", //client_id of the application configured for Bulk FHIR per Step 2
   "sub": "YOUR_CLIENT_ID ", //client_id of the application configured for Bulk FHIR per Step 2
   "aud":"https://idp-prod.prod.ngo.nextgenaws.net/auth/realms/nextgen/protocol/openid-connect/token",
    "exp": "<epoch_timestamp +300>", //JWT expiry, expressed as epoch timestamp; must be no more than 5 minutes into the future
   "jti": "<nonce>" //Random non-sequential, non-repeating string uniquely identifying this JWT
}
```

Sign this JWT with the public & private JWK corresponding to the “kid” specified in the JWT header.

This signed JWT will be passed as the value of the client_assertion parameter in the token request along with client_assertion_type, grant_type, and scope (see example below).

i. **Method**: POST
ii. **Endpoint**: https://idp-prod.prod.ngo.nextgenaws.net/auth/realms/nextgen/protocol/openid-connect/token
iii. **Headers**:
    1. Content-Type=application/x-www-form-urlencoded
iv. **Body**: (x-www-form-urlencoded)
    1. grant_type=client_credentials
    2. client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer
    3. client_assertion=<Signed_JWT>
    4. scope=system/*.read

Sample curl request for requesting token:

```
curl --location --request POST 'https://idp-prod.prod.ngo.nextgenaws.net/auth/realms/nextgen/protocol/openid-connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_assertion=<Signed_JWT>' \
--data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'scope=system/*.read'
```

A 200 response will include your access_token (example below) and list the authorized FHIR resource scope(s). The "expires_in" value is expressed in seconds (3600=1 hour).

```
{
    "access_token": "<generated_access_token>",
    "expires_in": 7200,
    "refresh_expires_in": 0,
    "token_type": "Bearer",
    "not-before-policy": 1661487077,
    "scope": "system/*.read"
}
```

Use this token to authorize (via 'Authorization: Bearer <access_token>' header) requests to the Bulk FHIR endpoints. (see also the HL7 Bulk FHIR Data Access Implementation Guide).

The NextGen Bulk FHIR base URL is https://fhir.meditouchehr.com/api/bulkfhir/r4

Please Refer to the Swagger Documentation for Bulk FHIR APIs at https://petstore.swagger.io/?url=https://hfstatic.s3.amazonaws.com/swagger/Bulk-API-R4-Swagger.yml#/

Bulk FHIR APIs

Patient Search API

An API that assists clients in searching for active patients within the practice to include in a group.

(GET) https://fhir.meditouch.com/api/bulkfhir/r4/$patient-search

Supported Params:

- page – A response will contain a maximum of 1,000 patients. To retrieve the next set of patients, use the page parameter.
- encDate – Get the list of patients whose encounter is created with certain date conditions.
  Date format should be <yyyy-MM-dd>
  Supported attributes:
    - eq (equal)
    - gt (greater than)
    - ge (greater than equal)
    - lt (less than)
    - le (less than equal)
- patientChartNo – Retrieve patient details with patient chart number.

Notes:

- All the params listed above are optional.
- The key of above parameters is case-sensitive.
- Search parameters can be used in combination, functioning as an AND operator.
- The response will include only active patients from the practice.
- The response will be in the FHIR R4 Patient resource format.

Examples:

(GET) https://fhir.meditouch.com/api/bulkfhir/r4/$patient-search?page=10

(GET) https://fhir.meditouch.com/api/bulkfhir/r4/$patient-search?encDate=gt<mm-dd-yyyy>&patientChartNo=HF12345

Group APIs

- **Create a Group:**
  (POST) https://fhir.meditouch.com/api/bulkfhir/r4/Group

  Request: *Content-Type: application/json*

```
{
 “resourceType”: “Group”,
 “name”: “Diabetes Patients”,
 “patientIds”: ["434334","122223"]
}
```

  Success response will contain group-id for the newly created group which can be used in the further requests.

  Note:
    - ‘**Group**’ keyword is case sensitive in the request URL.
    - A client can create a group with a maximum of 1,000 patients.
    - There must be at least one patient in the group for it to be created successfully.
    - A client can create up to 1,000 groups per practice.
    - The '**name**' field is optional and can contain any user-defined string to identify the group.

- **Update a Group:**
  (PUT) https://fhir.meditouch.com/api/bulkfhir/r4/Group/<group-id>

  Request: *Content-Type: application/json*

```
{
 “resourceType”: “Group”,
 “id”: “<group-id>”,
 “name”: “Diabetes Patients with age greater than 50”,
 “patientIdsToInclude”: ["122225","122228"],
 “patientIdsToExclude”: ["434334"]
}
```

  Note:
    - At least one of the following must be present in the request to update the group successfully: name, patientIdsToInclude, or patientIdsToExclude.
    - Removing all patients from the group is not allowed; instead, use the delete group API.

- **Delete a Group:**
  (DELETE) https://fhir.meditouch.com/api/bulkfhir/r4/Group/<group-id>

- **Get a Group:**
  (GET) https://fhir.meditouch.com/api/bulkfhir/r4/Group/<group-id>

- **Get All Groups:**
  (GET) https://fhir.meditouch.com/api/bulkfhir/r4/Group

Export APIs

- **Group of Patients Export:**
  An API to export data of patients present in the group.
  example:
  (GET) http://fhir.meditouch.com/api/bulkfhir/r4/Group/<group-id>/$export
  where <group-id> is id of group created using Group API.

- **Bulk Data Status Request:**
  Call URL is present at content location returned from export kick-off request
  example:
  (GET) http://fhir.meditouch.com/api/bulkfhir/r4/$export-poll-status?_jobId={jobID}

- **Bulk Data Delete Request:**
  Call URL is present at content location returned from export kick-off request
  example:
  (DELETE) http://fhir.meditouch.com/api/bulkfhir/r4/$export-poll-status?_jobId={jobID}

