|  |  |  |  |
| --- | --- | --- | --- |
| Document filename: | **NHS login Interface Specification - Provisioning** | | |
| Directorate / Programme | **Citizen Identity Programme** |  |  |
| Document Reference | **CID-IS-P** | Status | **DRAFT** |
| Information Asset Owner | Melissa Ruscoe | Version | **V1.0c** |
| Author | Simon Gordon | Version issue date | **15 Dec. 18** |

**NHS login**

**Interface Specification - Provisioning**

**Document Management**

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Summary of Changes** |
| V1.0 Draft 1 | 01 November 2018 | First draft for review |
| V1.0 | 09 Nov 2018 | Issued |
| v1.0a | 3 Dec 2018 | Updates based on feedback. Uplift to latest supported version of SCIM |
| v0.1c | 16 Jan 2019 | Update following review comments from e-RS team |
|  |  |  |
|  |  |  |

**Reviewers**

This document must be reviewed by the following people:

|  |  |  |  |
| --- | --- | --- | --- |
| **Reviewer name** | **Title / Responsibility** | **Date** | **Version** |
| Matthew Brown | Portfolio Lead Architect |  |  |
| Priyanka Mittal | Citizen Identity Technical Architect |  |  |

**Approved by**

This document must be approved by the following people:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title / Responsibility** | **Date** | **Version** |
| Melissa Ruscoe | Programme Head |  |  |
|  |  |  |  |

**Document Control:**

The controlled copy of this document is maintained in the NHS Digital corporate network. Any copies of this document held outside of that area, in whatever format (e.g. paper, email attachment), are considered to have passed out of control and should be checked for currency and validity.

Contents

[1 Introduction 4](#_Toc535902253)

[1.1 Purpose of Document 4](#_Toc535902254)

[1.2 Audience 4](#_Toc535902255)

[1.3 Definitions 4](#_Toc535902256)

[1.4 Token Endpoint 5](#_Toc535902257)

[2 Scope / Constraints 9](#_Toc535902258)

[3 Information Flows 10](#_Toc535902259)

[3.1 Overview 10](#_Toc535902260)

[3.2 User Provisioning 11](#_Toc535902261)

[4 Message Data 22](#_Toc535902262)

[4.1 JWT for Client Authorisation 22](#_Toc535902263)

[4.2 Token Payload – Authorization Request 22](#_Toc535902264)

[4.3 Token Payload – Access Token response to client 23](#_Toc535902265)

[4.4 JWT Signing 23](#_Toc535902266)

[4.5 User Provisioning 24](#_Toc535902267)

[5 References 28](#_Toc535902268)

# Introduction

## Purpose of Document

The NHS Digital NHS login Platform implements the SCIM Service Provider role to manage and provide identity information on End-Users to a Service Consumer, enabling the RP to send and obtain basic profile information about the End-User in an interoperable manner.

This document defines the interfaces implemented by the Platform and describes the data flows supported by these interfaces.

## Audience

The primary audiences for this document are:

* NHS login Programme team
* NHS Digital – Digital Delivery Centre
* NHS Digital – Other Delivery teams
* NHS England
* 3rd Party Suppliers integrating with the NHS Digital NHS login Platform

## Definitions

Where used in this document set, the keywords MUST, SHOULD and MAY are to be interpreted as follows:

* **MUST**: This word, or the terms “**REQUIRED**” or “**SHALL**”, means that the definition is an absolute` requirement of the specification.
* **SHOULD**: This word, or the adjective “**RECOMMENDED**”, means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications **MUST** be understood and carefully weighed before choosing a different course.
* **MAY**: This word, or the adjective “**OPTIONAL**”, means that an item is truly optional. One implementer may choose to include the item because a particular implementation requires it or because the implementer feels that it enhances the implementation while another implementer may omit the same item. An implementation which does not include a particular option **MUST** be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option **MUST** be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides).

## Token Endpoint

### Token Request

References:

* RFC7523 - JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants [1]

A Token request is used to obtain an Access Token. The Service Consumer (client) sends a Token Request to the Token Endpoint to obtain a Token Response.

The Client sends the parameters to the Token Endpoint using the HTTP POST method and the application/x-www-form-urlencoded serialization – the request must be sent using TLS v1.2 or above

Table 1: Token Request Parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Req? | Description |
| grant\_type | Mand | Value MUST be set to “urn:ietf:params:oauth:grant-type:jwt-bearer“ |
| assertion | Mand | A signed JWT, using JSON Web Signature – details below |
| scope | Mand | A list of space-delimited, case-sensitive strings specifying the requested scope(s). For the provisioning service the following scope values are defined   * https://[service fqdn]/Users.retrieve * https://[service fqdn]/Users.add   Section 4.5.1 details additional scope values that determine the content of retrieval messages. Scopes are associated with clients as part of the registration and assurance process. |

For example (non-normative):

POST /token HTTP/1.1

Host: auth.login.nhs.uk

Content-Type: application/x-www-form-urlencoded

grant\_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-assertion-type%3Ajwt-bearer&scope=Users.retrieve

&assertion=PHNhbWxwOl … ZT

The client creates an assertion, a JWT that is RSA-SHA512 signed with the pre-agreed RSA private key and adds the following claims in the payload. This is as per sections 4.2 and 4.4.

### Token Request Validation

References:

* JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants [1]
* Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants [2]

The Token Request is validated as described in JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants

### Token Response

#### Successful Response

References:

* JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants [1]
* Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants [2]

After receiving and validating a valid and authorised Token request from the client, the Token Endpoint returns a response which includes an ID Token and an Access Token. The response uses the “application/json” media type.

Table 2: Token Response: HTTP headers & values

|  |  |
| --- | --- |
| Header Name | Value |
| Cache-Control | no-store |
| Pragma | no-cache |

Table 3: Token Response

| Parameter | Req? | Description |
| --- | --- | --- |
| access\_token | Mand | Signed JWT which encodes the Access Token |
| token\_type | Mand | Must be value “bearer” |
| refresh\_token | n/a | Not currently supported |
| expires\_in | Opt | Recommended.  The lifetime in seconds of the access token.  The authorization server WILL provide the expiration time via the “exp” claim within the token, as per section 4.3 |
| scope | Cond | OPTIONAL, if identical to the scope requested by the client;  otherwise, REQUIRED |
|  |  |  |

For example:

HTTP/1.1 200 OK

Content-Type: application/json

Cache-Control: no-store

Pragma: no-cache

{

“access\_token”: “SlAV32hkKG”,

“token\_type”: “Bearer”,

“expires\_in”: 3600

}

#### Error Response

References:

* RFC6749 – The OAuth 2.0 Authorization Framework [3], s5.2

If the Token Request is invalid or unauthorized, the Authorization Server constructs the error response. The parameters of the Token Error Response are defined as in Section 5.2 of OAuth 2.0 [RFC6749]. The HTTP response body uses the application/json media type with HTTP response code of 400.

The following is a non-normative example Token Error Response:

HTTP/1.1 400 Bad Request

Content-Type: application/json

Cache-Control: no-store

Pragma: no-cache

{

“error”: “invalid\_request”

}

Table 4: Token Error Response

|  |  |  |
| --- | --- | --- |
| Parameter | Req? | Description |
| error | Mand | Error code |
| error\_description | Opt | Human-readable ASCII encoded text description of the error |
| error\_uri | Opt | URI of a web page that includes additional information about the error |

Table 5: Error Codes for Authentication Error Response

|  |  |
| --- | --- |
| Code | Description |
| OAuth 2.0 error codes | |
| invalid\_request | The request is missing a required parameter, includes an unsupported parameter value (other than grant type), repeats a parameter, includes multiple credentials, utilizes more than one mechanism for authenticating the client, or is otherwise malformed. |
| invalid\_client | Client authentication failed (e.g., unknown client, no client authentication included, or unsupported authentication method).  The authorization server MAY return an HTTP 401 (Unauthorized) status code to indicate which HTTP authentication schemes are supported. If the client attempted to authenticate via the “Authorization” request header field, the authorization server MUST respond with an HTTP 401 (Unauthorized) status code and include the “WWW-Authenticate” response header field matching the authentication scheme used by the client. |
| invalid\_grant | The provided authorization grant (e.g., authorization code, resource owner credentials) or refresh token is invalid, expired, revoked, does not match the redirection URI used in the authorization request, or was issued to another client. |
| unauthorized\_client | The authenticated client is not authorized to use this authorization grant type. |
| unsupported\_grant\_type | The authorization grant type is not supported by the authorization server. |
| invalid\_scope | The requested scope is invalid, unknown, malformed, or exceeds the scope granted by the resource owner |

# Scope / Constraints

* HTTP is not supported – all HTTP-based flows must be HTTPs (using TLS v1.2 or above)
* All examples contained within this document are non-normative

# Information Flows

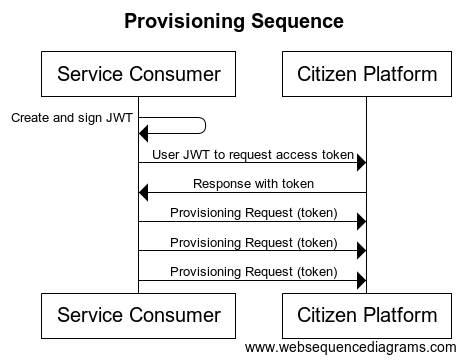
This section describes how the NHS login Platform supports information flows relating to Identity Provisioning.

## Overview

Broadly the pattern is:

1. Service Consumer provides its credentials to the NHS login Platform
2. NHS login Platform authenticates the Consumer and authorises access to the Provisioning endpoint
3. Service Consumer accesses the Provisioning endpoint with one or more messages, within the time constraint of the provided access token

The Service Consumer authenticates and is authorised to the platform using a standard OAuth Client Authorization Grant Flow – in this flow the Service Consumer is the OAuth Client



1. The Service Consumer prepares and signs a JWT to authenticate and authorise to the platform
2. The Service Consumer sends the request to the Citizen Platform token endpoint directly via HTTPs
3. The NHS login Platform validates the Authorization Request
4. The NHS login Platform authenticates the Service Consumer by validating the JWT signature
5. The NHS login Platform Authorizes the Service Consumer
6. The client receives a response that contains an Access Token in the response body
7. The client uses the Access token as a bearer token at the /Users endpoint to perform the provisioning actions. (next section)

## User Provisioning

References:

* RFC7644 System for Cross-domain Identity Management: Protocol [4]
* RFC6749 – The OAuth 2.0 Authorization Framework [3], s4.1.3

The User Provisioning Endpoint is an OAuth 2.0 Protected Resource that enables the core provisioning of an End-User. The endpoint uses SCIM as the underlying standard – this is basically a REST-based standard with conventions for attribute names and interface behaviour [4]

The endpoint for User Provisioning is /Users – the endpoint provides different behaviours for creating/amending and searching/retrieving users, along the general REST pattern.

|  |  |
| --- | --- |
| Message Flow | Messages to endpoint |
| Create Citizen Flow | 1. GET /Users?filter=nhsNumber eq “4444567890” 2. POST /Users |
| Retrieve Citizen by platform id | GET /Users/{id} |
| Retrieve Citizen by NHS Number | GET /Users?filter=nhsNumber eq “4444567890” |
| Amend Citizen status | 1. Retrieve, either    1. GET /Users/{id} or    2. GET /Users?filter=nhsNumber eq “4444567890” 2. POST /Users/{id} (with HTTP header X-HTTP-Method-Override: PUT) |
|  |  |

The Access Token from the Client Credentials grant MUST be sent as a Bearer Token using the Authorization header field, per Section 2 of OAuth 2.0 Bearer Token Usage.

### Create Citizen Flow

This flow is used when a Service Consumer is registering a new Citizen account within the Platform. The endpoint messages involved are:

|  |  |
| --- | --- |
| Message to Endpoint | Description |
| **Retrieve Citizen**  GET /Users?filter=nhsNumber eq “4444567890” | The Service first attempts to retrieve the details for a Citizen Account that the Citizen may already have |
| **Create Citizen**  POST /Users | (optional) If the retrieve finds no account, or only an account which is inactive or not verified, then the service creates the Citizen account |

#### Example Request – Retrieve Citizen by NHS Number

GET /Users?filter=nhsNumber eq “4444567890”

Host: example.com

Accept: application/json

Authorization: Bearer h480djs93hd8

#### Example Response – Retrieve Citizen by NHS Number

For a citizen which already exists in the Citizen Platform, a response would be as below

HTTP/1.1 200 OK

Content-Type: application/json

Location: <https://example.com/v1/Users/2819c223-7f76-453a-919d-413861904646>

Etag: W/”f250dd84f0671c3”

{

“schemas”: [“urn:ietf:params:scim:schemas:core:2.0:User”, “uk:nhs:login:auth:1.0:User”],

“id”: “2819c223-7f76-453a-919d-413861904646”,

“externalId”: “1294029928-001-222”,

“userName”: “bjensen@example.com”,

“emails”: [

{

“value”: “bjensen@example.com”,

“type”: “home”,

“primary”: true

},

{

“value”: “babs@jensen.org”,

“type”: “other”

}

],

“phoneNumbers”: [

{

“value”: “555-555-4444”,

“type”: “mobile”

}

],

“name”: {

“familyName”: “Jensen”

},

“active”: true,

“uk:nhs:login:auth:1.0:User”: {

“nhsNumber”: “9434760001”,

“delegators”: [

“4444567890”,

“4445555666”

],

“gpUserId”: “32498239048-3248734”,

“gpLinkageKey”: “YCRPyPSEUARu9edfjl”,

“gpOdsCode”: “A34123”,

“birthdate”: “1972-04-12”,

“vectorsOfTrust”{

“IdentityProofing”: “P9”

}

}

}

Since this is the create flow, it is usual for no Citizen resource to be found, hence the response below:

HTTP/1.1 404 NOT FOUND

{

“Errors”:[

{

“description”:”filter=nhsNumber eq &quot;4444567890&quot; not found”,

“code”:”404”

}

]

}

The citizen can now be created using the ‘Create Request’ message, an example is below

#### Example Request – Create Citizen

POST /Users HTTP/1.1

Host: auth.login.nhs.uk

Accept: application/json

Content-Type: application/json

Authorization: Bearer h480djs93hd8

Content-Length: …

{

“schemas”: [“urn:ietf:params:scim:schemas:core:2.0:User”, “uk:nhs:login:auth:1.0:User”],

“externalId”: “1294029928-001-222”,

“userName”: “bjensen@example.com”,

“emails”: [

{

“value”: “bjensen@example.com”,

“type”: “home”,

“primary”: true

},

{

“value”: “babs@jensen.org”,

“type”: “other”

}

],

“phoneNumbers”: [

{

“value”: “555-555-4444”,

“type”: “mobile”

}

],

“name”: {

“familyName”: “Jensen”

},

“active”: true,

“uk:nhs:login:auth:1.0:User”: {

“nhsNumber”: “9434760001”,

“delegators”: [

“4444567890”,

“4445555666”

],

“gpUserId”: “32498239048-3248734”,

“gpLinkageKey”: “YCRPyPSEUARu9edfjl”,

“gpOdsCode”: “A34123”,

“birthdate”: “1972-04-12”,

“verification”:{

“verificationStatus”: “verified”,

“verifiedBy”: “66781445561”,

“verifiedDatetime”: “2019-04-12T15:32:10.000Z”,

“verifiedMethod”: “1”,

“verificationEvidence”:[{

“evidenceIdentifier”: “349823098135497”,

“evidenceType”: “DrivingLicense”

}]

}

}

}

#### Example Response – Create Citizen

HTTP/1.1 201 Created

Content-Type: application/json

Location: <https://example.com/v1/Users/2819c223-7f76-453a-919d-413861904646>

{

“schemas”: [“urn:ietf:params:scim:schemas:core:2.0:User”, “uk:nhs:login:auth:1.0:User”],

“id”: “2819c223-7f76-453a-919d-413861904646”,

“externalId”: “1294029928-001-222”,

“userName”: “bjensen@example.com”,

“emails”: [

{

“value”: “bjensen@example.com”,

“type”: “home”,

“primary”: true

},

{

“value”: “babs@jensen.org”,

“type”: “other”

}

],

“phoneNumbers”: [

{

“value”: “555-555-4444”,

“type”: “mobile”

}

],

“name”: {

“familyName”: “Jensen”

},

“active”: true,

“uk:nhs:login:auth:1.0:User”: {

“nhsNumber”: “9434760001”,

“delegators”: [

“4444567890”,

“4445555666”

],

“gpUserId”: “32498239048-3248734”,

“gpLinkageKey”: “YCRPyPSEUARu9edfjl”,

“gpOdsCode”: “A34123”,

“birthdate”: “1972-04-12”,

“verification”:{

“verificationStatus”: “verified”,

“verifiedBy”: “66781445561”,

“verifiedDatetime”: “2019-04-12T15:32:10.000Z”,

“verifiedMethod”: “1”,

“verificationEvidence”:{

“evidenceIdentifier”: “349823098135497”,

“evidenceType”: “DrivingLicense”

}

},

“vectorsOfTrust”: {

“IdentityProofing”: “P9”

}

}

}

### Retrieve Citizen by platform ID Flow

This flow is used when a Service Consumer is registering a new Citizen account within the Platform. The endpoint messages involved are:

|  |  |
| --- | --- |
| Message to Endpoint | Description |
| **Retrieve Citizen**  GET /Users/{id} | The Service first attempts to retrieve the details for a Citizen Account that the Citizen may already have |

#### Example Request – Retrieve Citizen by ID

GET /Users/2819c223-7f76-453a-919d-413861904646

Host: example.com

Accept: application/json

Authorization: Bearer h480djs93hd8

#### Example Response – Retrieve Citizen by ID

For a citizen which already exists in the Citizen Platform, a response would be as below

HTTP/1.1 200 OK

Content-Type: application/json

Location: <https://example.com/v1/Users/2819c223-7f76-453a-919d-413861904646>

Etag: W/”f250dd84f0671c3”

{

“schemas”: [“urn:ietf:params:scim:schemas:core:2.0:User”, “uk:nhs:login:auth:1.0:User”],

“id”: “2819c223-7f76-453a-919d-413861904646”,

“externalId”: “1294029928-001-222”,

“userName”: “bjensen@example.com”,

“emails”: [

{

“value”: “bjensen@example.com”,

“type”: “home”,

“primary”: true

},

{

“value”: “babs@jensen.org”,

“type”: “other”

}

],

“phoneNumbers”: [

{

“value”: “555-555-4444”,

“type”: “mobile”

}

],

“name”: {

“familyName”: “Jensen”

},

“active”: true,

“uk:nhs:login:auth:1.0:User”: {

“nhsNumber”: “9434760001”,

“delegators”: [

“4444567890”,

“4445555666”

],

“gpUserId”: “32498239048-3248734”,

“gpLinkageKey”: “YCRPyPSEUARu9edfjl”,

“gpOdsCode”: “A34123”,

“birthdate”: “1972-04-12”,

“vectorsOfTrust”{

“IdentityProofing”: “P9”

}

}

}

If the Citizen no longer has an account in the Citizen Platform, a 404 is returned, as below

HTTP/1.1 404 NOT FOUND

{

“Errors”:[

{

“description”:”2819c223-7f76-453a-919d-413861904646 not found”,

“code”:”404”

}

]

}

### Retrieve Citizen by NHS Number Flow

This flow is used when a Service Consumer is registering a new Citizen account within the Platform. The endpoint messages involved are:

|  |  |
| --- | --- |
| Message to Endpoint | Description |
| **Retrieve Citizen**  GET /Users?filter=nhsNumber eq “4444567890” | The Service first attempts to retrieve the details for a Citizen Account that the Citizen may already have |

#### Example Request – Retrieve Citizen by NHS Number

Identical as per section 3.2.1.1

#### Example Response – Retrieve Citizen by NHS Number

Identical as per section 3.2.1.2

### Amend Citizen Flow

|  |  |
| --- | --- |
| Message to Endpoint | Description |
| **Retrieve Citizen**  GET /Users/{id}  Or  GET /Users?filter=nhsNumber eq “4444567890” | The Service first attempts to retrieve the details for a Citizen Account that the Citizen may already have |
| **Amend Request**  POST /Users  (with HTTP header X-HTTP-Method-Override: PUT) | (optional) If the retrieve finds an update is needed on the account, then the service amends the Citizen account |

#### Example Request – Retrieve Citizen by ID

Identical as per section 3.2.2.1

#### Example Response – Retrieve Citizen by ID

Identical as per section 3.2.2.2

#### Example Request – Retrieve Citizen by NHS Number

Identical as per section 3.2.1.1

#### Example Response – Retrieve Citizen by NHS Number

Identical as per section 3.2.1.2

#### Example Request – Amend Request

POST /Users/2819c223-7f76-453a-919d-413861904646 HTTP/1.1

Host: auth.login.nhs.uk

Accept: application/json

Content-Type: application/json

X-HTTP-Method-Override: PUT

Authorization: Bearer h480djs93hd8

Content-Length: …

{

“schemas”: [“urn:ietf:params:scim:schemas:core:2.0:User”, “uk:nhs:login:auth:1.0:User”],

“id”: “2819c223-7f76-453a-919d-413861904646”,

“externalId”: “1294029928-001-222”,

“username”: “test@tester.com”,

“emails”: [

{

“value”: “bjensen@example.com”,

“type”: “home”,

“primary”: true

},

{

“value”: “babs@jensen.org”,

“type”: “other”

}

],

“phoneNumbers”: [

{

“value”: “555-555-4444”,

“type”: “mobile”

}

],

“name”: {

“familyName”: “Jensen”

},

“delegators”: [

“4444567890”,

“4445555666”

],

“active”: true,

“uk:nhs:login:auth:1.0:User”: {

“nhsNumber”: “9434760001”,

“gpUserId”: “32498239048-3248734”,

“gpLinkageKey”: “YCRPyPSEUARu9edfjl”,

“gpOdsCode”: “A34123”,

“birthdate”: “1972-04-12”,

“verification”:{

“verificationStatus”: “verified”,

“verifiedBy”: “66781445561”,

“verifiedDatetime”: “2019-04-12T15:32:10.000Z”,

“verifiedMethod”: “1”,

“verificationEvidence”:{

“evidenceIdentifier”: “349823098135497”,

“evidenceType”: “DrivingLicense”

}

},

“vectorsOfTrust”{

“IdentityProofing”: “P9”

}

}

}

“

#### Example Response – Amend Response

HTTP/1.1 200 OK

Content-Type: application/json

Location: <https://example.com/v1/Users/2819c223-7f76-453a-919d-413861904646>

{

“schemas”: [“urn:ietf:params:scim:schemas:core:2.0:User”, “uk:nhs:login:auth:1.0:User”],

“id”: “2819c223-7f76-453a-919d-413861904646”,

“externalId”: “1294029928-001-222”,

“userName”: “bjensen@example.com”,

“emails”: [

{

“value”: “bjensen@example.com”,

“type”: “home”,

“primary”: true

},

{

“value”: “babs@jensen.org”,

“type”: “other”

}

],

“phoneNumbers”: [

{

“value”: “555-555-4444”,

“type”: “mobile”

}

],

“name”: {

“familyName”: “Jensen”

},

“active”: true,

“uk:nhs:login:auth:1.0:User”: {

“nhsNumber”: “9434760001”,

“delegators”: [

“4444567890”,

“4445555666”

],

“gpUserId”: “32498239048-3248734”,

“gpLinkageKey”: “YCRPyPSEUARu9edfjl”,

“gpOdsCode”: “A34123”,

“birthdate”: “1972-04-12”,

“verification”:{

“verificationStatus”: “verified”,

“verifiedBy”: “66781445561”,

“verifiedDatetime”: “2019-04-12T15:32:10.000Z”,

“verifiedMethod”: “1”,

“verificationEvidence”:{

“evidenceIdentifier”: “349823098135497”,

“evidenceType”: “DrivingLicense”

}

},

“vectorsOfTrust”{

“IdentityProofing”: “P9”

}

}

}

# Message Data

## JWT for Client Authorisation

References:

* RFC7519 – JSON Web Token (JWT) [5]
* RFC7515 – JSON Web Signature (JWS) [6]

The JWT header will contain the following claims:

Table 6: JWT Header

| Claim | Req? | Name | Description |
| --- | --- | --- | --- |
| alg | Mand | Algorithm used for signing the JWT | “RS512” – RSASSA-PKCS1-v1\_5 with the SHA-512 hash algorithm |
| typ | Mand | Type | “JWT” |

## Token Payload – Authorization Request

References:

* RFC7523 - JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants [1]

Table 7: Bearer token claims

|  |  |  |
| --- | --- | --- |
| Parameter | Example | Description |
| iss | myClientIdentifier1 | The identifier for the client system, as registered with the NHS login platform |
| sub | <https://auth.login.nhs.uk/provisioning> | The NHS login platform provisioning service |
| aud | <https://auth.login.nhs.uk/token> | The NHS login platform token URL |
| iat | 1311280970 | Time the JWT was created |
| exp | 1311281030 | Time the JWT will expire. How far in the future this can be will be agreed with each client |

## Token Payload – Access Token response to client

References:

* Access Tokens and Audit (JWT) [7]

Table 8: Access token claims

|  |  |  |
| --- | --- | --- |
| Parameter | Example | Description |
| iss | <https://auth.login.nhs.uk/> | The NHS login platform issuer identifier |
| sub | myClientIdentifier1 | The identifier for the client system, as registered with the NHS login platform |
| aud | <https://auth.login.nhs.uk/provisioning> | The NHS login platform provisioning service |
| iat | 1311280970 | Time the JWT was created |
| exp | 1311281030 | Time the JWT will expire. This claim MUST be used by clients to determine when new a new Access Token needs to be requested |
| scope | https://auth.login.nhs.uk/Users.retrieve | The scope(s) for which the access\_token is valid |
| Spine Core extensions | | |
| reason\_for\_request | directcare | The identified the purpose for which the request is being made.  Will contain the text “directcare” |
| requesting\_system | myClientIdentifier1 | Identifier for the system or device making the request |

## JWT Signing

References:

* RFC7519 – JSON Web Token (JWT) [5]
* RFC7515 – JSON Web Signature (JWS) [6]

All JWTs MUST be signed using the RSASSA-PKCS1-v1\_5 with the SHA-512 hash algorithm (“RS512”)

## User Provisioning

### User Attributes

References:

* RFC7644 System for Cross-domain Identity Management: Protocol [4]
* RFC7643 System for Cross-domain Identity Management: Core Schema [8]

The attributes in the above messages are specified below. All attributes are optional, except where listed otherwise. The schema is based upon:

https://tools.ietf.org/html/rfc7643

For “Retrieve Citizen” requests attributes will be returned based on the requested scope, with scopes defined in the NHS login Interface Specification – Provisioning. The mapping between requested scopes and returned attributes is as follows:

* “profile” scope will return “active”, “name.familyName”, “nhsNumber”, “birthdate” and “vectorsOfTrust.IdentityProofing”
* “email” scope will return “userName” and “emails.\*”
* “phone” scope will return “phoneNumbers.\*”
* “address” scope is not supported
* “gp\_registration\_details” will return “gpOdsCode”
* “gp\_integration\_credentials” will return “gpUserId” and “gpLinkageKey”
* “profile\_extended” will return “name.givenName”

Table 9: NHS login Provisioning Attribute Details

| **Attribute** | **Example value** | **Description** |
| --- | --- | --- |
| schemas | [“urn:ietf:params:  scim:schemas:core:  2.0:User”, “uk:nhs:login:  auth:1.0:User”] | Always present at this value  Note – newline characters inserted for formatting in this table |
| The following attributes are defined in the SCIM Schema for the “User” resource. The schema id is urn:ietf:params:scim:schemas:core:2.0:User | | |
| id | 2819c223-7f76-453a-919d-413861904646 | Identifier for the citizen profile/account within the NHS login Platform – mastered by the platform. Mandatory except for a create message. |
| externalId | 1294029928-001-222 | Identifier for the citizen profile/account within the client system |
| username | [bjensen@example.com](mailto:bjensen@example.com) | The friendly username for the citizen account – as mastered by the NHS login Platform. Mandatory for at least one value to be present. |
| emails[].value | [bjensen@example.com](mailto:bjensen@example.com) | The email address for the citizen. Mandatory for at least one value to be present. |
| emails[].type | “home”  “other” | home – this is the citizen’s own email address  other – the email address is known to be shared with other individuals |
| emails[].primary | true  false | One of the email addresses can be tagged as ‘true’ to denote the main address used by the citizen |
| name.familyName | Doe | Citizen’s family name |
| name.givenName | Jane | Citizen’s given name |
| phoneNumbers.value | 07900123456 | The citizen’s contact number |
| phoneNumbers.type | work  home  mobile  other | Note other is for contact number that is shared and/or owned by another individual |
| active | true | Citizen’s status within the GP System for digital access |
| The following attributes are NHS login extensions to the SCIM User resource. The NHS login extensions will be contained within the schema reference “uk:nhs:login:auth:1.0:User”. In accordance with the specification the following attributes will all be held within a JSON container with a name matching the schema reference. | | |
| nhsNumber | 9434760001 | The citizen’s NHS Number, if this has been verified |
| delegators | 4444567890,  4445555666 | Other users who have provided some level of proxy access to their accounts/records to the primary user. The numbers are NHS numbers |
| gpUserId | 32498239048-3248734 | Account number for GP Online services. Used in conjunction with the gpLinkageKey to register an app to access the patient’s GP information. Mastered by the GP system |
| gpLinkageKey | YCRPyPSEUARu9edfjl | May also be referred to as “passphrase” by the GP system. Mastered by the GP system |
| gpOdsCode | A12345 | The ODS Code for the citizen’s current GP system registration, if available |
| birthdate | 1972-04-12 | Citizen’s date of birth |
| verification | n/a | JSON object holding user identity verification details |
| verification.verificationStatus | verified | verified  not-verified |
| verification.verifiedBy | 66781445561 | National user role identifier |
| verification.verifiedDatetime | 2019-04-12T15:32:10.000Z | Timestamp |
| verification.verifiedMethod | 1 | Flag for method |
| verification.verifiedDetails | The user visited the practice | Extra details for the verification |
| verification. verificationEvidence | n/a | An array of verification evidence entries |
| verification. verificationEvidence. evidenceIdentifier | 349823098135497 | Identifier for the evidence |
| verification. verificationEvidence. evidenceType | DrivingLicense | Evidence type |
| vectorsOfTrust.IdentityProofing | P9 | Returned by the platform to indicate the level to which the user’s identity has been verified |

# References

|  |  |
| --- | --- |
| [1] | Internet Engineering Task Force (IETF), “RFC7523 - JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants,” [Online]. Available: https://tools.ietf.org/html/rfc7523. |
| [2] | Internet Engineering Task Force, “RFC7521: Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants,” [Online]. |
| [3] | Internet Engineering Task Force (IETF), “RFC6749 - The OAuth 2.0 Authorization Framework,” [Online]. Available: https://tools.ietf.org/html/rfc6749. |
| [4] | Internet Engineering Task Force (IETF), “RFC7644: System for Cross-domain Identity Management: Protocol,” [Online]. Available: https://tools.ietf.org/html/rfc7644. |
| [5] | Internet Engineering Task Force (IETF), “RFC7519: JSON Web Token (JWT),” [Online]. Available: https://tools.ietf.org/html/rfc7519. |
| [6] | Internet Engineering Task Force (IETF), “RFC7515 - JSON Web Signature (JWS),” [Online]. Available: https://tools.ietf.org/html/rfc7515. |
| [7] | NHS Digital, “Access Tokens and Audit (JWT),” [Online]. Available: https://developer.nhs.uk/apis/spine-core/security\_jwt.html. |
| [8] | Internet Engineering Task Force (IETF), “RFC7643: System for Cross-domain Identity Management: Core Schema,” [Online]. Available: https://tools.ietf.org/html/rfc7643. |
| [9] | OpenID Foundation, “OpenID Provider Authentication Policy Extension 1.0,” [Online]. Available: http://openid.net/specs/openid-provider-authentication-policy-extension-1\_0.html. |
| [10] | OpenID Foundation, “OpenID Connect Core 1.0 incorporating errata set 1,” [Online]. Available: https://openid.net/specs/openid-connect-core-1\_0.html. |
| [11] | OpenID Foundation, “Enhancing OAuth Security for Mobile Applications with PKCE,” [Online]. Available: http://openid.net/2015/05/26/enhancing-oauth-security-for-mobile-applications-with-pkse/. |
| [12] | OpenID Foundation, “OAuth 2.0 Multiple Response Type Encoding Practices,” [Online]. Available: http://openid.net/specs/oauth-v2-multiple-response-types-1\_0.html. |
| [13] | Internet Engineering Task Force (IETF), “RFC3339 - Date and Time on the Internet: Timestamps,” [Online]. Available: https://www.ietf.org/rfc/rfc3339.txt. |
| [14] | Cabinet Office, “GPG44 - Authentication credentials for online government services,” [Online]. Available: https://www.gov.uk/government/publications/authentication-credentials-for-online-government-services. |
| [15] | Cabinet Office, “GPG45 - Identity proofing and verification of an individual,” [Online]. Available: https://www.gov.uk/government/publications/identity-proofing-and-verification-of-an-individual. |
| [16] | Internet Engineering Task Force (IETF), “Vectors of Trust (Draft 0.9),” [Online]. Available: https://tools.ietf.org/html/draft-richer-vectors-of-trust-09. |
| [17] | OpenID Connect Foundation, “International Government Assurance Profile (iGov) for OpenID Connect 1.0,” [Online]. Available: https://xml2rfc.tools.ietf.org/cgi-bin/xml2rfc.cgi?Submit=Submit&format=ascii&mode=html&type=ascii&url=https://bitbucket.org/openid/igov/raw/master/openid-igov-profile.xml. |
| [18] | Internet Engineering Task Force (IETF), “RFC6750: OAuth 2.0 Bearer Token Usage,” [Online]. Available: https://tools.ietf.org/html/rfc6750. |
| [19] | NHS Digital, “DCB3051 Identity Verification and Authentication Standard for Digital Health and Care Services,” [Online]. Available: http://digital.nhs.uk/isce/publication/dcb3051. |
| [20] | NHS Digital, “DCB3051 Identity Verification and Authentication Standard for Digital Health and Care Services,” [Online]. Available: http://digital.nhs.uk/isce/publication/dcb3051. |