MEDENT API Documentation
Version: This document applies only to MEDENT version 23.5 and above.
Software Requirements:
Entities that wish to use the MEDENT API do not need to install any additional MEDENT affiliated software, nor do they need to configure any MEDENT affiliated software.

Dynamic Registration

This is for the Standalone Patient Launch, Bulk Data/Multi Patient API, and EHR Practitioner applications.

Registering your application:

This must be done programmatically using the Dynamic Registration standard. Registration must be done via HTTP POST to the dynamic registration URL below.
Reference: HL7.FHIR.US.DAVINCI-HREX\Dynamic Registration for SMART Apps - FHIR v4.0.1
References https://datatracker.ietf.org/doc/html/rfc7591#section-3.1

Dynamic Registration URL: https://fhir.medent.com/fhir/R4/dynamicregistration/

Once your application is processed, a client_id will be provided.

Parameters and Definitions

For Redirect URLs, Launch URLs, and Contacts: if providing multiple entries, Use JSON Array

Ex.

```
"redirect_uris":[
      "https://test.com/redirect",
      "https://test.com/redirect"
  ]

"initiate_login_uri":[
    "https://test.com/launch",
    "https://test.com/launch2"
  ]

"contacts":[
    "contact@test.com",
    "contact2@test.com",
  ]
```

The JWK URI should point to one or more JWK with example variable structure.

Ex.
"alg":"RS384",
"n": "0vx7a...",
"e":"AQAB",
One JWK must use RS384 at this time.

Reference: https://datatracker.ietf.org/doc/html/rfc7517

Scope Formatting based on Smart on FHIR Scopes V1

Example patient scope: patient/*.read
Example user scope: user/Patient.read user/AllergyIntolerance.read
Example bulk scope: system/*.*

Multiple scopes should be expressed as a space delimited single string, NOT a JSON array.

Patient / App Launch

token_endpoint_auth_method – optional
client_secret_basic registration with secret - default
none: registration without secret
grant_types – optional
if included, value must be authorization_code
if omitted, value will be authorization_code
response_types - required
value: code
client_name – required
This must be unique across all registrations – suggestion for format is to indicate the vendor/platform name in addition to application name to avoid user confusion
client_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]
logo_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]
scope - required
MUST contain "patient" SMART on FHIR scopes. Can contain multiple resources [e.g. "patient/*.read" or "patient/Patient.read patient/AllergyIntolerance.read"]
contacts: required
MUST contain valid email address.
Valid formats include
“contacts":”email@email.com”
or
"contacts:["email@email.com", "email2@email.com"]
tos_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]
policy_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]
software_id – optional
software_version – optional
jwks_uri – optional
jwks – optional and not stored
initiate_login_uri – required
redirect_uris – required
Must be valid [e.g. not return http 400 error]

User / EHR Launch

token_endpoint_auth_method – optional
client_secret_basic registration with secret - default
none: registration without secret
grant_types – optional

if included, value must be authorization_code
if omitted, value will be authorization_code

response_types – required
value: code

client_name – required
This must be unique across all registrations – suggestion for format is to indicate the vendor/platform name in addition to application name to avoid user confusion

client_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]

logo_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]

scope – required
MUST contain "user" SMART on FHIR scopes. Can contain multiple resources [e.g. "user/*.read" or "user/Patient.read user/AllergyIntolerance.read"]

contacts: required
MUST contain valid email address.
Valid formats include
“contacts":”email@email.com”
or
"contacts:["email@email.com", "email2@email.com"]

tos_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]

policy_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]

software_id – optional

software_version – optional

jwks_uri – optional

jwks – optional and not stored

initiate_login_uri – required
Must be valid [e.g. not return http 400 error]

redirect_uris – required
Must be valid [e.g. not return http 400 error]

Bulk / System

token_endpoint_auth_method – optional
client_secret_basic registration with secret - default

grant_types – required
value: client_credentials

response_types – optional

client_name – required
This must be unique across all registrations – suggestion for format is to indicate the vendor/platform name in addition to application name to avoid user confusion

client_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]

logo_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]

scope – required
MUST contain "system" SMART on FHIR scopes. Can contain multiple resources [e.g. "system/*.read" or "system/Patient.read system/AllergyIntolerance.read"]

contacts: required
MUST contain valid email address.

Valid formats include
“contacts":”email@email.com”
or
"contacts:["email@email.com", "email2@email.com"]

tos_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]

policy_uri – optional but highly recommended
If given, must be valid [e.g. not return http 400 error]

software_id – optional

software_version – optional

jwks_uri – required

jwks – optional and not stored

Errors

All error codes will contain a **code** and a description
All error codes will return a 400 http code.

invalid_redirect_uri
Redirect URL required by server.
Valid Redirect URLs required by server.

invalid_client_metadata
Registration required by server. - No registration was submitted
Json registration required by server. - There was a message submitted, but was not JSON format.
UDAP software_statement not supported.
Response Type code required by server.
Valid Client URL required by server.
Valid Logo URL required by server.
Valid Terms of Service URL required by server.
Valid Policy URL required by server.
SMART on FHIR scope required by server.
Patient or User Smart on FHIR scope is required by server.
Patient and User scopes must be registered separately.
Launch URL required by server.
JWKS URI required by server.
This application's registration is currently under review or the name is already being used. Please contact MEDENT Support: https://www.medent.com/api_support_form/"
Note this error message will appear if there is registration with the same client_name value as the attempted registration.

App Responsibilities

If an app is found to violate terms of service or use policy, the app first be disabled, and an email will be sent using the contact information provided during registration.
The app can be re-enabled if the conflict is resolved.

MEDENT API Additional Requirements

In order to connect to a MEDENT practice, App must obtain knowledge of practice IDs:

To View:
https://fhir.medent.com/fhir/resources/viewpracticelist.php
This page will require a clientid to view, given at Registration.

To Download, send a POST request to:
https://fhir.medent.com/fhir/resources/practices.php
With Parameter 'clientid', matching clientid given at Registration.
This will return a JSON container with name, baseUrl.

These queries will return FHIR Enabled practices that are ready to be queried.

Service Base URL Information
An HTTP GET request can be sent to https://fhir.medent.com/fhir/resources/ServiceBaseURL.php to receive a Bundle including the Endpoint and Organization information for Active MEDENT FHIR practices. This is an open endpoint that does not require authentication.

Patient SMART On FHIR Standalone Launch

Authentication will follow the OAuth 2.0 workflow.
The following assumes App has completed Dynamic Registration for system FHIR Scopes, and has valid client_id, and has queried for PracticeID's required.

Authorization

The request will be sent by the App redirecting to the authorize URL with the parameters listed below:

URL:
https://fhir.medent.com/fhir/R4/PracticeID/authorize

Expected Parameters sent from App:
response_type: Value must be “code”
client_id: This should be assigned by MEDENT at Registration
redirect_uri: must match a Redirect URL from Registration
scope: Reference SMART App Launch: Scopes and Launch Context (hl7.org)
state: Up to the application to determine
aud: URL of the FHIR server app would like to connect to

After receiving the above information, MEDENT will launch a Patient Portal Authentication and login where the user will be able to verify their credentials and confirm the scopes for the application access. This process also includes granting offline access permissions.

MEDENT will redirect to the application using the following information:
redirect url: redirect_uri from registration
url parameters:
code: randomly generated access code
state: code sent by App to Authorization Endpoint

Access Token
After obtaining an authorization code, the application trades the code for an access token via HTTP POST to the EHR authorization server’s token endpoint URL.

For public applications, authentication is not possible (and thus not required), since the app cannot be trusted to protect a secret. For confidential apps, an Authorization header using HTTP Basic authentication is required, where the username is the application's client_id and the password is the application's client_secret

Token Endpoint Locations:
https://fhir.medent.com/fhir/R4/token/index.php?medent_practice_id=PRACTICEID

Expected Parameters sent from the application:
grant_type: Value must be "authorization_code"
code: MUST be Authorization Code generated during authentication
redirect_uri MUST match a registered application that also has a matching client_id

For Public Apps:
client_id: should match client_id from registration

For Confidential Apps:
The client_id and client_secret will be sent in an Authorization: Basic header.
Reference: http://hl7.org/fhir/smart-app-launch/basic-auth-example/index.html

If the client_id is "my-app" and the client_secret is "my-app-secret-123", then the header uses the value B64Encode("my-app:my-app-secret-123"), which converts to bXktYXBwOm15LWFwcC1zZWNyZXQtMTIz. This gives the app the Authorization token for "Basic Auth".
Example GET header:
Authorization: Basic bXktYXBwOm15LWFwcC1zZWNyZXQtMTIz

client_id MUST have an active registration and be the owner of both the authorization code and client_secret.

Information returned to application:
access_token: generated token
token_type: Bearer
expires_in: 900 [lifetime in seconds, 15 minutes]
scope: scope of items you are allowed to access
refresh_token: Refresh tokens can be used to keep the access 'alive' for an amount of time.
need_patient_banner: false
smart_style_url: https://fhir.medent.com/fhir/resources/smart-style.php
If launch/patient scope requested:
patient: patient MRN the requestor is allowed to access

With Headers:
Cache-Control: no-store
Pragma: no-cache

Refresh Token
Refresh tokens are issued to enable sessions to last longer than the validity period of an access token. Expiration on a refresh token is 24 hours.

Token Endpoint Locations:
https://fhir.medent.com/fhir/R4/PracticeID/token

Expected Parameters sent from the application:
grant_type: MUST be refresh_token
refresh_token: MUST match the refresh_token sent back with the access_token
scope [optional]: Should match the originally requested scope. We will always send back a new access token with the originally requested scope.

For Public Apps:
client_id: should match client_id from registration

For Confidential Apps:
The client_id and client_secret will be sent in an Authorization: Basic header.
Reference: http://hl7.org/fhir/smart-app-launch/basic-auth-example/index.html
If the client_id is “my-app” and the client_secret is “my-app-secret-123”, then the header uses the value B64Encode(“my-app:my-app-secret-123”), which converts to bXktYXBwOm15LWFwcC1zZWNyZXQtMTIz. This gives the app the Authorization token for “Basic Auth”.
Example GET header:
Authorization: Basic bXktYXBwOm15LWFwcC1zZWNyZXQtMTIz

client_id MUST have an active registration and be the owner of both the authorization code and client_secret.

Information returned to application:
access_token: generated token.
token_type: value will be “bearer”
expires_in: 900 [lifetime in seconds, 15 minutes]
scope: scope of items you are allowed to access
refresh_token: Refresh tokens can be used to keep the access 'alive' for an amount of time.
If launch/patient scope requested:
patient: patient MRN the requestor is allowed to access.

With Headers:
Cache-Control: no-store
Pragma: no-cache

Practitioner SMART On FHIR EHR Launch

Authentication will follow the OAuth 2.0 workflow.
The following assumes App has completed Dynamic Registration for Patient FHIR Scopes, and has an enabled client_id, and has queried for PracticeID's required.

The application is launched from the EHR by calling the launch URL. The following parameters are included:
launch: A randomly generated token to be used once and will allow for exchange of an authorization code.
iss: The base FHIR URL for the MEDENT practice

Authorization
The App will launch a request to the Authorization Endpoint to verify the application identity and desired scopes.

The request will be sent by the App redirecting to the authorize URL with the parameters listed below:

Authorize Endpoint Locations:
https://fhir.medent.com/fhir/R4/PracticeID/authorize

Expected Parameters sent from the application:
response_type: Value must be “code”
client_id: This should be assigned by MEDENT at Registration
redirect_uri: must match a Redirect URL from Registration
scope: Reference SMART App Launch: Scopes and Launch Context (hl7.org)
state: Up to the application to determine
aud: URL of the fhir server they would like to connect to, should be:
https://fhir.medent.com/fhir/R4/PracticeID
launch: This will be the launch token we sent to the Launch URL.

MEDENT will redirect to the application using the following information:
redirect url: redirect_uri from registration
url parameters:
code: randomly generated access code
state: sent by App to Authorization Endpoint

Access Token
After obtaining an authorization code, the application trades the code for an access token via HTTP POST to the EHR authorization server’s token endpoint URL

For public apps, authentication is not possible (and thus not required), since the app cannot be trusted to protect a secret. For confidential apps, an Authorization header using HTTP Basic authentication is required, where the username is the application’s client_id and the password is the application’s client_secret.

Token Endpoint Locations:
https://fhir.medent.com/fhir/R4/token/index.php?medent_practice_id=PRACTICEID

Expected Parameters sent from the application:
grant_type: MUST be authorization_code
code: MUST be Authorization Code sent from Authentication endpoint
redirect_uri MUST match a registered App that also has a matching client_id

For Confidential Apps:
The client_id and client_secret will be sent in an Authorization: Basic header.
Reference: http://hl7.org/fhir/smart-app-launch/basic-auth-example/index.html
If the client_id is “my-app” and the client_secret is “my-app-secret-123”, then the header uses the value B64Encode(“my-app:my-app-secret-123”), which converts to bXktYXBwOm15LWFwcC1zZWNyZXQtMTIz. This gives the app the Authorization token for “Basic Auth”.
Example GET header:
Authorization: Basic bXktYXBwOm15LWFwcC1zZWNyZXQtMTIz

client_id MUST have an active registration and be the owner of both the authorization code and client_secret.

Information returned to application:
access_token: generated token
token_type: Bearer

expires_in: 900 [lifetime in seconds, 15 minutes]
scope: scope of items you are allowed to access
refresh_token: Refresh tokens can be used to keep the access 'alive' for an amount of time.
need_patient_banner: false
smart_style_url: https://fhir.medent.com/fhir/resources/smart-style.php
If launch/patient scope requested:
patient: patient MRN the requestor is allowed to access

Refresh Token
Refresh tokens are issued to enable sessions to last longer than the validity period of an access token. Expiration on a refresh token is 24 hours.

Token Endpoint Locations:
https://fhir.medent.com/fhir/R4/PracticeID/token

Expected Parameters sent from the application:
grant_type: MUST be refresh_token
refresh_token: MUST match the refresh_token sent back with the access_token
scope [optional]: Should match the originally requested scope. We will always send back a new access token with the originally requested scope.

For Confidential Apps:

The client_id and client_secret will be sent in an Authorization: Basic header.
Reference: http://hl7.org/fhir/smart-app-launch/basic-auth-example/index.html

If the client_id is “my-app” and the client_secret is “my-app-secret-123”, then the header uses the value B64Encode(“my-app:my-app-secret-123”), which converts to bXktYXBwOm15LWFwcC1zZWNyZXQtMTIz. This gives the app the Authorization token for “Basic Auth”.
Example GET header:
Authorization: Basic bXktYXBwOm15LWFwcC1zZWNyZXQtMTIz

client_id MUST have an active registration and be the owner of both the authorization code and client_secret.

Information returned to application:
access_token: generated token.
token_type: value will be “bearer”
expires_in: 900 [lifetime in seconds, 15 minutes]
scope: scope of items you are allowed to access
refresh_token: Refresh tokens can be used to keep the access 'alive' for an amount of time.
If launch/patient scope requested:
patient: patient MRN the requestor is allowed to access.

Bulk Data Access SMART On FHIR (Flat FHIR)

Allows applications to access bulk FHIR resources Asynchronously. MEDENT does not support real-time Bulk queries.

Authentication will follow the OAuth 2.0 workflow.
The following assumes App has completed Dynamic Registration for Patient FHIR Scopes, and has an enabled client_id, and has queried for PracticeID.

This also requires the App provide a raw JWT or a URL pointing to a JWT at the time of Registration.

Token Endpoint

The app will launch a request to the Token endpoint with a Signed JWT requesting data.

Token Endpoint Locations:
https://fhir.medent.com/fhir/R4/token/index.php?medent_practice_id=PRACTICEID

Reference:
https://hl7.org/fhir/uv/bulkdata/STU1.0.1/authorization/index.html#obtaining-an-access-token

JWT will contain:

Iss	required	Issuer of the JWT -- the client's client_id, as determined during registration with the FHIR authorization server (note that this is the same as the value for the sub claim)
sub	required	The service's client_id, as determined during registration with the FHIR authorization server (note that this is the same as the value for the iss claim)
aud	required	The FHIR authorization server's "token URL" (the same URL to which this authentication JWT will be posted -- see below)
exp	required	Expiration time integer for this authentication JWT, expressed in seconds since the "Epoch" (1970-01-01T00:00:00Z UTC). This time SHALL be no more than 5 minutes in the future.
jti	required	A nonce string value that uniquely identifies this authentication JWT.

Request will contain headers:

alg	required	The JWA algorithm (e.g., RS384, ES384) used for signing the authentication JWT.
kid	required	The identifier of the key-pair used to sign this JWT. This identifier SHALL be unique within the client's JWK Set.
typ	required	Fixed value: JWT.

Request will contain POSTed information:

scope	required	The scope of access requested. See note about scopes below
grant_type	required	Fixed value: client_credentials
client_assertion_type	required	Fixed value: urn:ietf:params:oauth:client-assertion-type:jwt-bearer
client_assertion	required	Signed authentication JWT value (see above)

Verification Rules:

- iss and sub must match and match a registered client id, given during Dynamic Registration.
- aud must match token endpoint
https://fhir.medent.com/fhir/R4/token/index.php?medent_practice_id=PRACTICEID
- exp must be 5 minutes or less.
- jti must be unique for each request
- alg must match RS384 or ES384
- kid in header must match kid in the pre-registered JWT.
- typ must be JWT
- jku is an optional parameter and not checked.

- scope must match registered scopes, or be a smaller subset of scopes.
- grant_type must be client_credentials
- client_assertion_type must be urn:ietf:params:oauth:client-assertion-type:jwt-bearer
- client_assertion will be run through a signature verification using the value of alg header.

MEDENT supports both ES384 and RS384 Encryption.

Return with information:

Information in Json format:
access_token: generated token.
token_type: Bearer [fixed]
expires_in: 300 [lifetime in seconds, 5 minutes]
scope: scope of items you are allowed to access: none

With Headers:
Cache-Control: no-store
Pragma: no-cache
Content-Type: application/json

Group Endpoint

How to Obtain GroupID:
At this time, each MEDENT Practice will need to approve application access to Groups.

Available Practice Filters for Groups: Insurance, Provider, Location, and Age

Using the authentication token, an application can query for Active and Inactive groups at the Group Endpoint.
Ex. https://medentfhir.com/fhir/R4/PRACTICEID/Group?active=true

Alternatively, each MEDENT Practice can be contacted to set up custom groups and /or obtain group IDs for relevant information.

Asynchronous Requests

Following the Bulk Authentication will result in an Access Token used to access the Group Endpoint.

After authentication and confirmation of a GROUPID are completed, an $export request needs to be sent to the Group Endpoint.

Export request:
Ref: http://hl7.org/fhir/uv/bulkdata/export.html#endpoint---group-of-patients

Currently: Only the [fhir base]/Group/[id]/$export function is available, without additional filtering.
By default, all requested scopes [at the time of registration] will be generated and available after an export request. Optionally, a MEDENT Practice may restrict information available by adjusting the scopes.

As search parameters become available, they will be communicated via updated CapabilityStatement.

After the $export request has been successfully sent to the Group Endpoint, a Content-Location header will point to the appropriate Async endpoint to remove a request, check the status of the request, or pickup completed response files. These operations are described below.

Delete request:
This will be processed when the async URL is accessed with a DELETE request_method.

This will send a message to the practice's dataset to remove a request. The request cannot be removed if it has already started processing.

Success:
HTTP response code: 202
Operation Outcome: Informational, Delete request processed successfully.

Errors:
HTTP response code: 424
Operation Outcome: Error, Request Already Started, Cannot Remove.

HTTP response code: 404
Operation Outcome: Error, Request Not Found
This is if the group ID given is incorrect or could not be found.

Status check request:
This will be processed when the async URL is accessed with a GET request_method.

In-Progress:
HTTP response code: 202
X-Progress header: Calculated based on number of patients processed.
Operation Outcome: Informational, Status update request processed successfully.

Errors:
HTTP response code: 404
Operation Outcome: Error, Request Not Found
This is if the group ID given is incorrect or could not be found.

Complete:
X-Progress header: 100%
Body information:
transactionTime: The date and time the files started generating.
request: The original request query sent in, including Group ID.
requiresAccessToken: true [this will require the use of the authentication token when retrieving the file content]
output: for each file:
url: endpoint to retrieve a file.[ex: https://fhir.medent.com/fhir/R4/async/?medent_practice_id=PRACTICEID&filename=2.76.Patient.0]
error: for each file that failed:
url: endpoint to retrieve a file, this file will contain errors in OperationOutcome format.
type: FHIR Resource Name [ex: Patient]

File Request:

This will be processed when the output:url from the Status Check is accessed with a GET request_method.

If Requested File Found / Completed:
HTTP response code: 200
contentType header: application/fhir+ndjson
Body: requested content in newline delimited json format [ndjson]
example:
{"active":true,"address":[{"city":"SYRAC ...
{"active":true,"address":[{"city":"SYRAC ....
{"active":true,"address":[{"city":"SYRAC .....
...

Errors:
HTTP response code: 404
Operation Outcome: Error, Request Not Found
This is if the filename given is incorrect or could not be found.

Async Options / Bulk File Content

Obtain Asynchronous Files:
At this time, the MEDENT Practice controls how often the information requested will be available. This will be able to be provided in 1-7 days based on practice settings.

Request Frequency:
Requests will be denied with a 429 "Too Many Requests" if there is already one in progress for each Requesting App and Group combination.

If a request comes in, and the request is Complete, it will Auto-Expire the Completed item and allow the new request to be added.

File Format:
newline delimited json format [ndjson]
example:
{"active":true,"address":[{"city":"SYRAC ...
{"active":true,"address":[{"city":"SYRAC ....
{"active":true,"address":[{"city":"SYRAC .....
…

Requests will be paged at 50 entries per resource.

Expiration:
After a request is completed, the expiration date will be set to one day after.

If a request is Expired, and a new request comes in, it will allow the new request.

For Resource definitions, please refer to the appropriate document found here:
https://www.MEDENT.com/onc

Referenced Specifications:
HL7 FHIR US Core Implementation Guide – STU3 Release 3.1.1 - http://hl7.org/fhir/us/core/STU3.1.1/

SMART Application Launch Framework Implementation Guide - Release 1.0.0 - http://www.hl7.org/fhir/smart-app-launch/1.0.0/
OpenID Connect Core – 1.0 - https://openid.net/specs/openid-connect-core-1_0.html