daVinci EHR – API Guide

(g)(9) Application Access

Revision History

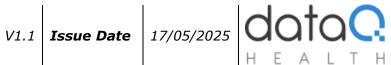| Version	Date	Author	Reviewer	Description
V1.1	2025-05-17	Muhammad Ali Malik	Abdur Rehman, Asad Ali	Draft
V1.1	2025-05-27	Muhammad Ali Malik	Abdur Rehman, Asad Ali	Final

Table of Contents

1 Introduction

This guide provides an overview of how patients can use standards-based APIs to access their health information, in accordance with the ONC's certification criterion §170.315(g)(8). It ensures that all relevant API documentation is openly available to the public without requiring registration or login. Patients can refer to the steps outlined below to securely and easily access their personal health data through these APIs.

1.1 API Tools

An **API (Application Programming Interface)** acts like a secure messenger that allows different software systems to communicate and exchange data. In this context, APIs let you, as a patient, retrieve your personal health records from your provider's system using standard tools.

You can use free and publicly available tools to connect with these APIs. These tools require a **basic understanding of APIs** and how to make requests such as **GET** (to retrieve data) — similar to interacting with a website but with more control.

While Postman is a commonly used option, you're free to use any API client of your choice. Download Postman.

2   API List

Executing the following list of APIs would successfully generate the CCDA data file.

2.1   Generate OTP

The first step is to generate an OTP using your MR number. The MR number is a unique patient identifier that is used by your practice/provider to identify you in the EHR system. This number can be found on payment receipts, medical records or lab results given to you by your practice. If you have access to your patient web portal or mobile app you can also find it there.

POST: https://qa.wmi360.com/EHR/api/main/api/v1/patienttoken/generateotp

This API body takes MR number as input to deliver an OTP to your registered email address. The OTP is valid for 5 minutes and expires.

Schema

Element Name	Data Type	Card	Description
patientMrn	string	1..1	MR number from practice records

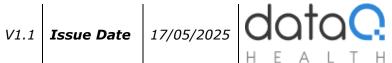**Sample Request**

Headers	POST (body)

Accept: */*
Content-Type: application/json-patch+json; x-api-version=1.0
	`{`
	  `"patientMrn": "AB001234"`
	`}`

Response

The response will show if the OTP (one-time password) was sent to the registered email. If the email is invalid, it will say the MRN (medical record number) entered is wrong.

```
{
    "message": "POST Request successful.",
    "result": "OTP sent successfully"
}
```

2.2 Generate Token

The second step after being authenticated through OTP is to generate a token. A token is a special code or key that proves your identity after entering the OTP (one-time password). It allows you to securely access a system or service, **acting like a digital pass** to confirm you're authorized.

POST: https://qa.wmi360.com/EHR/api/main/api/v1/patienttoken/generatetoken

This API takes MRN and OTP to generate a token.

Schema

Element Name	Data Type	Card	Description
patientMrn	string	1..1	MR number from practice records
Otp	string	1..1	OPT from registered email address.

Sample Request

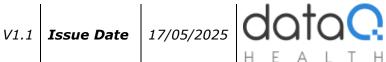| Headers	POST (body)

Accept: */*
 	
Content-Type: application/json-patch+json; x-api-version=1.0
	`{`
	`   "patientMrn": "AB001234"`
	`}`

Response

The response gives a token, the time it was created, and how long it's valid. The token stops working one hour after it's generated.

```
{
    "message": "POST Request successful.",
    "result": {
        "token": "HvyzW2kwD.CymG8ipCFkCE1eJwLALBXlja",
        "generatedOn": "2025-06-02T11:17:34.5178033-05:00",
        "validUpto": "2025-06-02T12:17:34.5178485-05:00",
        "error": null
    }
}
```

In case of invalid parameters, following message will be received.

```
    "Invalid MRN or OTP"
```

2.3 Get Patient Encounters

Here, the API gives a list of patient visits, sorted by the Date of Service (DOS), so patients can pick which visit's records they want to view.

POST: https://qa.wmi360.com/EHR/api/main/api/v1/patienttoken/patientencounters

This API takes token, from and to date parameters to return an array of patient encounter IDs during that date range.

Schema

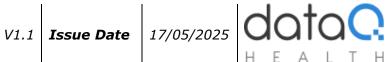| Element Name	Data Type	Card	Description
token	string	1..1	Token generated using MRN & OTP.
fromDate	dateTimeOffset	1..1	Start of date range
toDate	dateTimeOffset	1..1	End of date range

Sample Request

Headers	POST (body)
Accept: */* Content-Type: application/json-patch+json; x-api-version=1.0	```{ "token": "kwD.CymG8ipCFkCE1eJwLALBXlja", "fromDate": "2025-02-02T13:49:39.200Z", "toDate": "2025-06-02T13:49:39.200Z" }```

Response

The response is a list of patient visits (encounters). Extra details help find the right visit, but you only need the "id" to create a CCDA document.

```
{
    "message": "POST Request successful.",
    "result": [
        {
            "id": 86219,
            "patientId": 15906,
            "locationId": 9683,
            "providerId": 1,
            "encounterDateTime": "2025-05-26T14:00:00-05:00",
            ...
        ...
        ...

        {
            "id": 79253,
```

```
                "patientId": 15906,
                "locationId": 9683,
                "providerId": 1,
                "encounterDateTime": "2025-05-21T14:00:00-05:00",
                ...
                ...
                ...
            }
        ]
    }
```

In case of invalid parameters, following message will be received.

```
    "Invalid Token "
```

2.4   Generate CCDA for Encounter IDs

This API provides the CCDA document in XML format. Multiple encounter IDs could be passed to this API

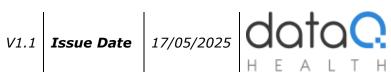**POST:** https://qa.wmi360.com/EHR/api/main/api/v1/patienttoken/generateccda

Schema

Element Name	Data Type	Card	Description
token	string	1..1	Token generated using MRN & OTP.
encounterIds	array of integers	1..1	Encounter IDs - comma separate if multiple.

Sample Request

Headers	POST (body)
Accept: */* Content-Type: application/json-patch+json; x-api-version=1.0	```{ "token": "kwD.CymG8ipCFkCE1eJwLABXlja ", "encounterIds": [86219, 79253] }```

Response

The response would be a CCDA file in UTF-8 encoded XML format.

```
{
    "message": "POST Request successful.",
    "result": "<?xml version=\"1.0\" encoding=\"utf-8\"?>\r\n<ClinicalDocument
xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" xmlns:voc=\"urn:hl7-
org:v3/voc\" xmlns:sdtc=\"urn:hl7-org:sdtc\" xmlns=\"urn:hl7-org:v3\">\r\n  <realmCode
code=\"US\
...
...
...
}
```