New York State Medicaid Management Information System (MMIS)

EVV Submitter
EP6720 - EVV Data Aggregator

Interface Control Document (ICD):
Electronic Visit Verification (EVV) Data API

Version 1.1
7/13/2020

Document Number: EMD.CLM.1180.01.0420
Contract Number: C014305
# Table of Contents

Table of Contents........................................................................................................... ii
List of Figures .................................................................................................................... iii
List of Tables ..................................................................................................................... iii

1. Purpose of Interface Control.......................................................................................... 1
2. Introduction ..................................................................................................................... 2
3. Overview ......................................................................................................................... 3
4. Assumptions/Constraints/Risks ..................................................................................... 4
   4.1 Assumptions ............................................................................................................... 4
   4.2 Constraints ................................................................................................................ 4
   4.3 Risks ......................................................................................................................... 4
5. General Interface Requirements ................................................................................... 5
   5.1 Interface Overview .................................................................................................... 5
   5.2 Functional Allocation ............................................................................................... 5
   5.3 Data Transfer ............................................................................................................ 6
   5.4 Transactions .............................................................................................................. 6
   5.5 Security and Integrity ............................................................................................... 8
   5.6 Operational Support & Service Levels ................................................................. 8
      5.6.1 Service Level Agreements (SLA) ................................................................... 8
      5.6.2 Operational Expectations ............................................................................. 8
6. Detailed Interface Requirements .................................................................................. 10
   6.1 Requirements for Electronic Visit Verification (EVV) Data API ............................ 10
      6.1.1 Assumptions .................................................................................................... 10
      6.1.2 General Processing Steps .............................................................................. 10
      6.1.3 Interface Processing Time Requirements ...................................................... 10
      6.1.4 File Naming Convention .............................................................................. 11
      6.1.5 Message Format (or Record Layout) and Required Protocols ...................... 11
      6.1.6 Communication Methods ............................................................................ 19
      6.1.7 Security Requirements ................................................................................ 19
7. Qualification Methods ................................................................................................... 20

Appendix A – Record of Changes..................................................................................... 21
Appendix B – Acronyms ................................................................................................... 22
Appendix C – Glossary ....................................................................................................... 23
Appendix D – Referenced Documents ............................................................................. 24
Appendix E – Approvals ................................................................................................... 25
Appendix F – Contact Information .................................................................................. 26
Appendix G – Additional Appendices .............................................................................. 27
List of Figures

Figure 1 – Data Transfer Diagram

List of Tables

Table 1 – EVV Request Data Model Table
Table 2 – Error Data Model Table
Table 3 – HTTP Response Status Code to Data Type Mapping Table
Table 4 – Record of Changes
Table 5 – Acronyms
Table 6 – Glossary
Table 7 – Referenced Documents
Table 8 – Approvals
Table 9 – Contact Information
1. **Purpose of Interface Control**

The intended audience of the Electronic Medicaid of New York (eMedNY) Project Interface Control Document (ICD) is all project stakeholders, including the project sponsor, senior leadership, and the project team.

This ICD defines the eMedNY system’s Electronic Visit Verification (EVV) interface with EVV Submitters. It is worth noting that EVV Submitters may consist of EVV Vendors, Providers, and Managed Care Organizations (MCO). ICD is to communicate all inputs and outputs from the system for all potential actions. Its intended audience is the project managers, project teams, development teams, and stakeholders interested in interfacing with the system utilizing this eMedNY interface.
2. Introduction

Electronic Medicaid of New York (eMedNY) is the New York State (NYS) Medicaid program claims processing system. The system allows NYS Medicaid providers to submit claims and receive payments for Medicaid-covered services provided to eligible clients.

This Interface Control Document (ICD) describes the relationship between the Electronic Medicaid of New York (eMedNY) information system and EVV Submitters and specifies the requirements of both participating systems. This includes the concept of operations, the file structure and protocols that govern the interchange of data, and the communication paths along which the data is expected to flow.

This is the current and official version of the interface. Other versions in development represent proposed changes until fully approved.

For each interface, the following information will be provided:

- A general description of the interface
- Assumptions where appropriate
- A description of the data exchange format and protocol for exchange
- Estimated size and frequency of data exchange
3. **Overview**

The EVV interface will enable providers to easily and securely transmit EVV data to eMedNY which will be sent to the Medicaid Data Warehouse (MDW) for aggregation. Collecting and aggregating this EVV data is a necessary step for New York state to achieve compliance with the 21st Century Cures Act (the Cures Act) and avoid Federal Medical Assistance Percentages (FMAP) penalties. Ultimately, the data stored will be mapped to claims and encounters which will provide new fraud, waste, and abuse detection capabilities.
4. **Assumptions/Constraints/Risks**

4.1 **Assumptions**

- None

4.2 **Constraints**

- eMedNY’s web service is Internet facing.
- No coding changes that interrupts connectivity between the two systems may be performed on this interface without NYSDOH approval.
- All web service request/response activity must be logged, correlated and reviewed to ensure compliance with New York State (NYS) auditing practices.
- ICD will be reviewed and updated based on enhancements or maintenance activities and will be posted to the eMedNY website.

4.3 **Risks**

- None
5. **General Interface Requirements**

5.1 **Interface Overview**

The eMedNY EVV Interface is an internet facing Representational State Transfer (REST) Application Programming Interface (API). The primary end user of this interface will be EVV Submitters. This may include EVV Vendors, Providers, and Managed Care Organizations. The service will allow the EVV Submitters to submit specific electronic visit verification data for Medicaid personal care services (PCS) and home health care services (HHCS) that require an in-home visit by a provider.

The EVV interface is built leveraging REST design patterns, utilizing JSON as the information exchange structures.

5.2 **Functional Allocation**

EVV submitters will initiate a service request in an on-demand manner as part of their normal course of operations. The EVV service is designed to support the collection of electronic visit verification data for Medicaid personal care services (PCS) and home health care services (HHCS) that require an in-home visit by a provider.

The EVV interface will accept requests from EVV Submitters as part of their business processes and provide a response, as per the interface requirements.
5.3 Data Transfer

EVV Submitters will initiate a service request over a secure HyperText Transfer Protocol (HTTP) connection (HTTPS) to an eMedNY hosted RESTful API. All information will be transferred between both parties as JSON documents or HTTP Uniform Resource Identifiers (URIs) using REST Design best practices.

![Data Transfer Diagram](image)

Figure 1 – Data Transfer Diagram

5.4 Transactions

The eMedNY EVV Service exposes a single REST HTTPS endpoint for EVV Submitters to send member specific details for Medicaid personal care services (PCS) and home health care services (HHCS) that require an in-home visit by a provider.

If the EVV Data (service payload) do not pass validation, the records will be rejected with appropriate reason code.

- Providers will be able submit multiple EVV records per submission.
- Provider EVV Proxys (MCO, VO, Aggregator, Vendor, etc.) will be able to submit for multiple providers which can include multiple EVV records per provider per submission.
- Error handling will be able to accept successful rows and reject only bad rows (with an appropriate reject reason).
- The service will identify previously accepted records and reject duplicated data with an appropriate reject reason.
<table>
<thead>
<tr>
<th>Use Case</th>
<th>HTTP Operation</th>
<th>URI Path</th>
<th>Request Payload</th>
<th>Request Response</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Request Submission</td>
<td>POST</td>
<td>/claims/submitter/{submitterId} /evv</td>
<td>payload</td>
<td>error[]</td>
<td>This can be used for submitting one or many transactions</td>
</tr>
<tr>
<td>Individual Submission</td>
<td>PUT</td>
<td>/claims/submitter/{submitterId} /evv/{transactionId}</td>
<td>visit</td>
<td>error[]</td>
<td>Can be used for submitting only one transaction</td>
</tr>
<tr>
<td>Update</td>
<td>PUT</td>
<td>/claims/submitter/{submitterId} /evv/{transactionId}</td>
<td>visit</td>
<td>error[]</td>
<td>Can also be used for updating the transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Note: Updates to a previously accepted transaction will overwrite the previous record and may be subject to restrictions.</td>
</tr>
<tr>
<td>Delete</td>
<td>DELETE</td>
<td>/claims/submitter/{submitterId} /evv/{transactionId}</td>
<td>visit</td>
<td>error[]</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>Get</td>
<td>GET</td>
<td>/claims/submitter/{submitterId} /evv/{transactionId}</td>
<td>visit</td>
<td>error[]</td>
<td>Reserved for future use</td>
</tr>
</tbody>
</table>
5.5 Security and Integrity

The EVV interface will contain both Protected Health Information (PHI) and Personally Identifiable Information (PII). The security approach for this interface will fall into two areas of concern: encryption, and authentication and authorization.

The EVV Service will utilize HTTPS with Security Socket Layer (SSL) encryption and the Transport Layer Security (TLS) version 1.2.

The EVV Service will leverage API Keys for authentication and authorization to enforce identity verification and service authorization. The eMedNY operations team will provide the client systems their client ID with security tokens.

Security and data validation checks will be performed on the EVV web service requests in this interface. When issues with this interface are discovered that prevent data exchange, EVV Submitters will be notified immediately by eMedNY operations team.

5.6 Operational Support & Service Levels

This section discusses the Operational Level expectations of the interface regarding availability, performance, and scale. Interfaces do not inherit eMedNY’s “continuously available” SLA’s unless specifically mentioned in the section below.

5.6.1 Service Level Agreements (SLA)

This interface does not have any Service Level Agreements.

5.6.2 Operational Expectations

5.6.2.1 Usage Policy

This interface is designed to meet the use cases as mentioned in the requirements document. Extending the usage of interfaces beyond the original intent requires the approval of NYSDOH and GDIT.

The interface user/consumer might be disabled if their usage pattern presents security risks or operational expectations are not met.

5.6.2.2 Availability & Performance

Below is the general guidance on the operational levels on this SLA. These are delivered on a best efforts basis.

- 99.98 with the exception of planned outages
- Average response is < 10 seconds/transaction excluding bulk record submission
- Maximum number of concurrent connections/API client is 5; more than 5 concurrent threads will be throttled

5.6.2.3 Support

- If you face issues with the interface in the production environment, call the GDIT Command Center at +1 318-383-4000.
- Non production issues need to be communicated to the onboarding team/project team during normal business hours.
6. Detailed Interface Requirements

This section refers and/or describes details for the EVV interface. This ICD defines the conditions under which the EVV interface is to be leveraged.

6.1 Requirements for Electronic Visit Verification (EVV) Data API

The EVV interface supports the need for EVV Submitters to submit Electronic Visit Verification data for Medicaid personal care services (PCS) and home health care services (HHCS) that require an in-home visit by a provider.

The eMedNY program exposes the EVV service for the sole purpose of meeting the New York State Department of Health (DOH) requirement under the 21st Century Cures Act (the Cures Act), mandating that states implement Electronic Visit Verification (EVV) for all Medicaid personal care services (PCS) and home health care services (HHCS) that require an in-home visit by a provider.

- The API consumers (EVV Submitter) and provider (eMedNY API Platform) are expected to adhere to REST best practices in API access and operations.
- REST Enabled Client with the ability to consume and produce Content-Type: application/json
- API Consumers should be able to produce JSON payload that meet the service specification ([Refer to RESTful API Modeling Language (RAML) document](#)).
- API Consumers should accurately use the appropriate URI patterns recommended for each transaction along with HTTP verb.

The following URL’s will be used as the base URI for the eMedNY EVV web service for production and test environments:

Production: https://api.emedny.io
Test: https://api.emednytest.io

6.1.1 Assumptions

None

6.1.2 General Processing Steps

EVV Submitters can invoke the EVV service as needed to submit electronic visit verification data for Medicaid personal care services (PCS) and home health care services (HHCS) that require an in-home visit by a provider.

6.1.3 Interface Processing Time Requirements

Electronic Visit Verification (EVV) Data API offers services on a “Best Effort” basis. There is no guarantee on availability or performance but every reasonable effort would be taken to provide a highly available and responsive service.

On average service is expected to respond within 10 seconds excluding bulk record submission.

A maximum on 5 concurrent calls/second is allowed / consumer. API Access will be revoked or temporarily denied if service policies are violated.
6.1.4 File Naming Convention
N/A

6.1.5 Message Format (or Record Layout) and Required Protocols

API Message format is documented and managed through a RAML (www.raml.org) and method to model and document REST API. Message interchange format is JSON. Both inbound and outbound message format should comply with the object model provided in the RAML specification. See Appendix G.

6.1.5.1 HTTP Protocol

The Electronic Visit Verification (EVV) Data service will expose its endpoints over secure HTTP (HTTPS). The service will leverage HTTP status codes to inform the consumer of the response being returned.

At no point will the service be executed by EVV Submitters over an insecure protocol.

6.1.5.1.1 HTTP Response Status Codes

The Electronic Visit Verification (EVV) Data API adheres to REST design principles in that the service will return an HTTP response status code which provides clients of their request’s overarching result. At a high level the following series of status codes can be categorized as follows:

- **2xx**: Success – Indicates that the client’s request was accepted successfully
- **4xx**: Client Error – Indicates that the client must take some additional action in order to complete their request.
- **5xx**: Server Error – Indicates that the server takes responsibility for these error status codes

Generally speaking, client error codes in the 4xx range are a result of an error on the client side and will require that the client, in this case EVV Submitters, take an action to resolve the returned error. Whereas 5xx range status codes require that the service, in this case eMedNY, must take an action to resolve the error.

The following are the HTTP Response Status Codes which will be returned by the service and their associated meaning. With each Response Status Code, a specific response data structure will be returned, data structures will be addressed in the next section of this document.

<table>
<thead>
<tr>
<th>HTTP STATUS CODE</th>
<th>DESCRIPTION</th>
<th>EXAMPLE SCENARIOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Success</td>
<td>OK</td>
</tr>
<tr>
<td>201</td>
<td>Created</td>
<td>Created</td>
</tr>
<tr>
<td>204</td>
<td>No Content</td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>Partial Content</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>Client Error</td>
<td>Bad Request</td>
</tr>
<tr>
<td>Code</td>
<td>Status</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
<td>The service has denied your request due to a failure in authentication. Review your security credentials to ensure they are accurate.</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
<td>The service has denied your request due to the client not having the proper authorization to invoke the service. Unlike a 401, a retry will not resolve the issue. Contact eMedNY operations support to grant permission for the client to access the service.</td>
</tr>
<tr>
<td>404</td>
<td>Not Found</td>
<td>The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible.</td>
</tr>
<tr>
<td>500</td>
<td>Server Error</td>
<td>An error occurred within the application and the application could not process the requests. This error implies an issue with the service and not with the client or the data which was passed to the service by the client. This error implies that the request can be tried again once the service issues have been resolved.</td>
</tr>
</tbody>
</table>

### 6.1.5.2 Data Assembly Characteristics

The EVV Service defines a single endpoint for consumption. As this service is a REST service, the interface is governed by a RAML specification. This specification defines the input and output behaviors of the service including: endpoint mappings, request data types, and response detail like: each http response code and the associated response data type. Attached in Appendix G you will find the RAML specification which describes the EVV service.

The next sections will discuss the details of the request and response data structures.
### Table 1 – EVV Request Data Model Table

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Data Element Number</th>
<th>Length</th>
<th>Format</th>
<th>Required</th>
<th>Description/Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visit Object</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>transactionId</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Unique transaction ID per visit generated by the EVV system when the EVV record is generated. Transaction ID must not be generated outside the EVV system including during submission. The recommendation would be to use a UUID/GUID Compliant ID if available.</td>
</tr>
<tr>
<td>memberId</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Medicaid Id for the recipient receiving the service. A unique identifier assigned to each Medicaid Member by the Welfare Management System (WMS) or NYSoh. It serves to identify the medical data pertaining to the individual as the unique permanent identifier. Must pass Client ID Check Digit. Client ID must exist on the B_DETAIL_TB.</td>
</tr>
<tr>
<td>dateOfBirth</td>
<td>date-only</td>
<td></td>
<td></td>
<td>YYYY-MM-DD</td>
<td>Yes</td>
<td>Date of Birth of the recipient receiving the service. Cannot be greater than the current date (future date). Must match the date of birth on the B_DETAIL_TB.</td>
</tr>
<tr>
<td>providerName</td>
<td>string</td>
<td></td>
<td></td>
<td>Max: 35</td>
<td>No</td>
<td>Provider Name is the name of a provider of Medicaid services as used on official State records.</td>
</tr>
<tr>
<td>nationalProviderId</td>
<td>string</td>
<td></td>
<td></td>
<td>10</td>
<td>Situational</td>
<td>National Provider Identifier (NPI) is the nationally recognized provider identifier assigned by the Center for Medicare &amp; Medicaid Services (CMS). Required if MMIS Identifier is not present. Must Pass NPI Billing Check Digit.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Data Element Number</td>
<td>Length</td>
<td>Format</td>
<td>Required</td>
<td>Description/Validation</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
<td>---------------------</td>
<td>--------</td>
<td>--------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>providerId</td>
<td>string</td>
<td></td>
<td>8</td>
<td></td>
<td>Situational</td>
<td>When NPI and Provider ID are both present, they must be a valid combination in eMedNY. MMIS Identifier is a unique number generated by the eMedNY system and assigned to each provider enrolled to provide services to Members of the Medicaid program. This number is the primary method of identifying a provider. Required if National Provider Identifier (NPI) is not present. Must pass MMIS Billing Check Digit. Must be active on Date of Service. When NPI and Provider ID are both present, they must be a valid combination in eMedNY.</td>
</tr>
<tr>
<td>taxpayerId</td>
<td>string</td>
<td></td>
<td>9</td>
<td></td>
<td>Yes</td>
<td>Federal Employer Identification Number (FEIN)</td>
</tr>
<tr>
<td>providerAddress</td>
<td>Address</td>
<td></td>
<td></td>
<td>No</td>
<td></td>
<td>Providers most current street address, city, state and zip code.</td>
</tr>
<tr>
<td>providerRateCode</td>
<td>string</td>
<td></td>
<td>4</td>
<td></td>
<td>No</td>
<td>Rate Code specifies a medical service or product that utilizes a rate reimbursement technique processed by the eMedNY system. All Institutional claims are paid by rate code and they include: Clinic, Managed Care, Inpatient, ICF/DD, Child Care, Home Health and Nursing Home claims. Must be a valid rate code.</td>
</tr>
<tr>
<td>procedureCode</td>
<td>string</td>
<td></td>
<td>Min: 5</td>
<td>Max: 5</td>
<td>Yes</td>
<td>Procedure Code for the service rendered to the recipient by the provider. Must be a valid (HCPCS) procedure code.</td>
</tr>
<tr>
<td>procedureModCode</td>
<td>array</td>
<td></td>
<td>2</td>
<td></td>
<td>No</td>
<td>Two character number modifying the procedure code for the service rendered to the recipient by the provider.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Data Element Number</td>
<td>Length</td>
<td>Format</td>
<td>Required</td>
<td>Description/Validation</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------</td>
<td>---------------------</td>
<td>--------</td>
<td>-----------------</td>
<td>----------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Must be a valid modifier, up to 4 occurrences.</td>
</tr>
<tr>
<td>serviceStartDateTime</td>
<td>datetime-only</td>
<td></td>
<td>YYYY-M-DDThh:mm:ss</td>
<td>Yes</td>
<td>Begin date/time of the recipient receiving the service. Must be a valid date/time. Cannot be greater than the current date (future date).</td>
<td></td>
</tr>
<tr>
<td>serviceEndDateTime</td>
<td>datetime-only</td>
<td></td>
<td>YYYY-M-DDThh:mm:ss</td>
<td>Yes</td>
<td>End date/time of the recipient receiving the service. Must be a valid date/time. Must be greater than Begin date/time. Cannot be greater than the current date (future date).</td>
<td></td>
</tr>
<tr>
<td>serviceAddress</td>
<td>Address</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Current street address, city, state and zip code where the recipient is receiving the service.</td>
</tr>
<tr>
<td>serviceProviderFirstName</td>
<td>string</td>
<td>Min: 1 Max: 35</td>
<td></td>
<td></td>
<td>Yes</td>
<td>First name of the servicing worker.</td>
</tr>
<tr>
<td>serviceProviderLastName</td>
<td>string</td>
<td>Min: 1 Max: 60</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Last name of the servicing worker.</td>
</tr>
<tr>
<td>serviceProviderPhoneNumber</td>
<td>string</td>
<td></td>
<td>10</td>
<td>99999999999</td>
<td>No</td>
<td>Phone number of the servicing worker.</td>
</tr>
<tr>
<td>serviceProviderSSN</td>
<td>string</td>
<td></td>
<td>9</td>
<td>9999999999</td>
<td>Situational</td>
<td>SSN of the servicing worker. Required if Tax Identifier of the servicing worker is not present.</td>
</tr>
<tr>
<td>serviceProviderTaxPayerId</td>
<td>string</td>
<td></td>
<td>9</td>
<td>9999999999</td>
<td>Situational</td>
<td>Tax Identifier of the servicing worker. Required if SSN of the servicing worker is not present.</td>
</tr>
<tr>
<td>eVisitMethod</td>
<td>string</td>
<td></td>
<td>Max: 9</td>
<td></td>
<td>No</td>
<td>Method by which data was entered into the E.V.V. system (T-Telephone, F-Fob, M-Mobile App, O-Other). Must be a valid Clock in/out method code.</td>
</tr>
</tbody>
</table>

Address Object

ICD Version 1.1
Appendix xx - DRAFT (EP6720) Electronic Visit Verification (EVV) Data API ICD.docx
7/13/2020
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Data Element Number</th>
<th>Length</th>
<th>Format</th>
<th>Required</th>
<th>Description/Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>street</td>
<td>string</td>
<td></td>
<td>Max: 40</td>
<td></td>
<td></td>
<td>Building number and Street Name</td>
</tr>
<tr>
<td>city</td>
<td>string</td>
<td></td>
<td>Max: 25</td>
<td></td>
<td></td>
<td>City</td>
</tr>
<tr>
<td>state</td>
<td>string</td>
<td></td>
<td>Max: 2</td>
<td></td>
<td></td>
<td>State</td>
</tr>
<tr>
<td>zip</td>
<td>string</td>
<td>Min: 5 Max: 9</td>
<td></td>
<td>999999 or 999999</td>
<td></td>
<td>Zip Code</td>
</tr>
</tbody>
</table>

**Submitter ID**

| submitterId | string | 8 | Yes | The organization submitting the EVV transactions on behalf of the Provider. The Submitter ID will be in the URI and is not required as a payload, since it will be same for a given submitter. This is also in line with the REST Design and allows us to apply security rules based on the submitter. |

ICD Version 1.1
Appendix xx - DRAFT (EP6720) Electronic Visit Verification (EVV) Data API ICD.docx

7/13/2020
6.1.5.2.1.1  Error Data Model

In the event that the EVV service encounters an error the service will respond with an error message. The error message along with the HTTP Response Status Code will provide EVV Submitters with detail related to the issue encountered.

Table 2 – Error Data Model Table

<table>
<thead>
<tr>
<th>Object</th>
<th>Property</th>
<th>Data Element Number</th>
<th>Length</th>
<th>Format</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A simple message returned from the service upon encountering an error</td>
</tr>
<tr>
<td></td>
<td>transactionId</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Unique transaction ID used by the EVV vendor to submit the transaction</td>
</tr>
<tr>
<td></td>
<td>code</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Error Code</td>
</tr>
<tr>
<td></td>
<td>message</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Error Message</td>
</tr>
</tbody>
</table>

6.1.5.3  HTTP Response Status Code to Data Type Mapping

In this section we will describe each of the possible data types and conditions where they will be returned based on the HTTP Response Status Code. The table provided will map the status code to the name of the data element, both defined in the above sections of the document.

Table 3 – HTTP Response Status Code to Data Type Mapping Table

<table>
<thead>
<tr>
<th>HTTP Response Status Code</th>
<th>Data Model</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Response</td>
<td>The service will return a 200 so long as a technical error is not encountered.</td>
</tr>
<tr>
<td>201</td>
<td>Response</td>
<td>The request has been fulfilled and resulted in a new resource being created.</td>
</tr>
<tr>
<td>204</td>
<td>Response</td>
<td>The server has successfully fulfilled the request and that there is no additional content to send in the response payload body.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Response</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>206</td>
<td>Response</td>
<td>The server is successfully fulfilling a range request for the target resource by transferring one or more parts of the selected representation that correspond to the satisfiable ranges found in the request's Range header field.</td>
</tr>
<tr>
<td>400</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
<tr>
<td>401</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
<tr>
<td>403</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
<tr>
<td>404</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
<tr>
<td>500</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
</tbody>
</table>
6.1.6 Communication Methods

The Electronic Visit Verification (EVV) Data API service will be exposed as a RESTful secure HTTP (HTTPS) web service for the EVV Submitter system to consume. The service will leverage the HTTP Methods POST and PUT as part of Phase 1.

6.1.6.1 Interface Initiation

Interface can be invoked by issuing a REST API Call.

6.1.6.2 Flow Control

eMedNY will notify EVV Submitters should the Electronic Visit Verification (EVV) Data API service be unavailable or any access issues are encountered.

6.1.7 Security Requirements

1) API Key and secret
   a. The HTTP request should have the following key values in the header
      i. api_id = API_KEY issued by eMedNY API Gateway
      ii. api_secret = API_SECRET Issued by the API Gateway

2) All communications with eMedNY should be on TLS 1.2
7. Qualification Methods

Qualification methods to be used to verify that the requirements for the interfaces defined in Section 6 “Detailed Interface Requirements” have been met include:

- During Construction Phase, unit test packages are created with visual examination and approval of test results.
- During System Integration Phase, test packages are created with visual examination and approval of test results. Test results are then presented to NYSDOH for their sign-off.
- During Quality Assurance Phase, automated scripts are executed to determine results are as expected.
- Post Implementation Validate is performed after a project is moved into Production where visual examination of file processing is reviewed for accuracy.
# Appendix A – Record of Changes

## Table 4 – Record of Changes

<table>
<thead>
<tr>
<th>Version Number</th>
<th>Date</th>
<th>Author/Owner</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4/1/2020</td>
<td>Cyd Hoen-Shears/GDIT</td>
<td>Initial</td>
</tr>
</tbody>
</table>

ICD Version 1.1  
eMedNY – EP6720  
Appendix xx - DRAFT (EP6720) Electronic Visit Verification (EVV) Data API ICD.docx  
7/13/2020
# Appendix B – Acronyms

**Table 5 – Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Literal Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BDD</td>
<td>Business Design Document</td>
</tr>
<tr>
<td>eMedNY</td>
<td>Electronic Medicaid of New York</td>
</tr>
<tr>
<td>EVV</td>
<td>Electronic Visit Verification</td>
</tr>
<tr>
<td>FMAP</td>
<td>Federal Medical Assistance Percentages</td>
</tr>
<tr>
<td>GDIT</td>
<td>General Dynamics Information Technology</td>
</tr>
<tr>
<td>HHCS</td>
<td>Home Health Care Services</td>
</tr>
<tr>
<td>HTTP/HTTPS</td>
<td>HyperText Transfer Protocol/HyperText Transfer Protocol Secure</td>
</tr>
<tr>
<td>ICD</td>
<td>Interface Control Document</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>MCO</td>
<td>Managed Care Organization</td>
</tr>
<tr>
<td>MDW</td>
<td>Medicaid Data Warehouse</td>
</tr>
<tr>
<td>NYS</td>
<td>New York State</td>
</tr>
<tr>
<td>NYSDOH</td>
<td>New York State Department of Health</td>
</tr>
<tr>
<td>PCS</td>
<td>Personal Care Services</td>
</tr>
<tr>
<td>PHI</td>
<td>Protected Health Information</td>
</tr>
<tr>
<td>PII</td>
<td>Personally Identifiable Information</td>
</tr>
<tr>
<td>RAML</td>
<td>RESTful API Modeling Language</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SIT</td>
<td>Systems Integration Testing</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreements</td>
</tr>
<tr>
<td>SSL/TSL</td>
<td>Secure Sockets Layer/Transport Layer Security</td>
</tr>
<tr>
<td>URL/URI</td>
<td>Uniform Resource Locator/Uniform Resource Identifier</td>
</tr>
<tr>
<td>VO</td>
<td>Verification Organization</td>
</tr>
</tbody>
</table>
## Appendix C – Glossary

**Table 6 – Glossary**

<table>
<thead>
<tr>
<th>Term</th>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Term&gt;</td>
<td>&lt;Acronym&gt;</td>
<td>&lt;Definition&gt;</td>
</tr>
<tr>
<td>&lt;Term&gt;</td>
<td>&lt;Acronym&gt;</td>
<td>&lt;Definition&gt;</td>
</tr>
<tr>
<td>&lt;Term&gt;</td>
<td>&lt;Acronym&gt;</td>
<td>&lt;Definition&gt;</td>
</tr>
<tr>
<td>&lt;Term&gt;</td>
<td>&lt;Acronym&gt;</td>
<td>&lt;Definition&gt;</td>
</tr>
<tr>
<td>&lt;Term&gt;</td>
<td>&lt;Acronym&gt;</td>
<td>&lt;Definition&gt;</td>
</tr>
</tbody>
</table>
# Appendix D – Referenced Documents

## Table 7 – Referenced Documents

<table>
<thead>
<tr>
<th>Document Name</th>
<th>Document Location and/or URL</th>
<th>Issuance Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix E – Approvals

The undersigned acknowledge that they have reviewed the ICD and agree with the information presented within this document. Changes to this ICD will be coordinated with, and approved by, the undersigned, or their designated representatives.

Table 8 – Approvals

<table>
<thead>
<tr>
<th>Document Approved By</th>
<th>Date Approved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: Patric Dempster, CISO and Privacy Officer - GDIT</td>
<td></td>
</tr>
<tr>
<td>Name: &lt;Name&gt;, &lt;Job Title&gt; - &lt;Company&gt;</td>
<td></td>
</tr>
<tr>
<td>Name: &lt;Name&gt;, &lt;Job Title&gt; - &lt;Company&gt;</td>
<td></td>
</tr>
<tr>
<td>Name: &lt;Name&gt;, &lt;Job Title&gt; - &lt;Company&gt;</td>
<td></td>
</tr>
<tr>
<td>Name: &lt;Name&gt;, &lt;Job Title&gt; - &lt;Company&gt;</td>
<td></td>
</tr>
</tbody>
</table>
# Appendix F – Contact Information

Table 9 – Contact Information

<table>
<thead>
<tr>
<th>Entity</th>
<th>Contact Description/Name</th>
<th>Phone</th>
<th>Email Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDIT</td>
<td>RDC Command Center</td>
<td>318-549-4333</td>
<td><a href="mailto:RDC_Command_Center@gdit.com">RDC_Command_Center@gdit.com</a></td>
</tr>
</tbody>
</table>
Appendix G – Additional Appendices

RAML Specification

```yaml
#%RAML 1.0

title: Electronic Visit Verification API

description: This service will allow Healthcare Providers, Issuers and Vendors to upload the Electronic Visit Verification information to eMedNY to comply with 21st Century Cures Act.

version: '1.0'

mediaType: application/json

protocols:
  - HTTPS

baseUri: /emedny

traits:
  api-key-required:
    headers:
      api_key:
        type: string
      api_secret:
        type: string

types: # Global types that will be used in this API
visit: # Electronic Visit Verification Records
  type: object
  properties:
    transactionId:
      required: true
      type: string
      description: Unique transaction id used by the EVV vendor. 1) This ID Must be unique within a submitter 2) Could be used to track back to the submitter’s source for audit control 3) we recommed using a UUID Or GUID
    minLength: 10
    maxLength: 150

# Member (Medicaid Recipient Information)
memberId:
  type: string
```
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medicaid Id</td>
<td>string</td>
<td>Description: Medicaid Id for the recipient receiving the service.</td>
</tr>
<tr>
<td>dateOfBirth</td>
<td>date-only</td>
<td>Description: Date of Birth of the recipient receiving the service.</td>
</tr>
<tr>
<td>providerName</td>
<td>string</td>
<td>Description: Provider Name is the name of a provider of Medicaid services as used on official State records.</td>
</tr>
<tr>
<td>nationalProviderId</td>
<td>string</td>
<td>Description: National Provider Identifier (NPI) is the nationally recognized provider identifier assigned by the Center for Medicare &amp; Medicaid Services (CMS).</td>
</tr>
<tr>
<td>providerId</td>
<td>string</td>
<td>Description: MMIS Identifier is a unique number generated by the eMedNY system and assigned to each provider enrolled to provide services to Members of the Medicaid program. This number is the primary method of identifying a provider.</td>
</tr>
<tr>
<td>taxPayerId</td>
<td>string</td>
<td>#Please provide either tax payer ID or FEIN</td>
</tr>
</tbody>
</table>
type: string
description: Tax Identifier of the provider or Federal Employer Identification Number (FEIN). format:############
required: true

providerAddress:
type: address
description: Providers most current address
required: false

providerRateCode:
type: string
minLength: 4
maxLength: 4
required: false
description: Rate Code specifies a medical service or product that utilizes a rate reimbursement technique processed by the eMedNY system. All Institutional claims are paid by rate code and they include Clinic, Managed Care, Inpatient, ICF/DD, Child Care, Home Health and Nursing Home claims.

#Service Information
procedureCode:
type: string
minLength: 5
maxLength: 5
required: true
description: Procedure Code for the service rendered to the recipient by the provider

procedureModCode:
type: string[]
required: false
description: Array of Two character string modifying the procedure code for the service rendered to the recipient by the provider

serviceStartDateTime:
type: datetime-only
description: Begin date timestamp of the recipient receiving the service.
required: true
serviceEndDateTime:
  type: datetime-only
  description: End date timestamp of the recipient receiving the service.
  required: true

serviceAddress:
  type: address
  description: Current address where recipient is receiving the service
  required: true

#Service Provider Information

serviceProviderFirstName:
  type: string
  description: First name of the servicing worker
  minLength: 1
  maxLength: 35
  required: true

serviceProviderLastName:
  type: string
  description: Last name of the servicing worker
  minLength: 1
  maxLength: 60
  required: true

serviceProviderPhoneNumber:
  type: string
  description: Phone Number of the service worker. format:#########
  required: false

serviceProviderSSN:
  type: string
  description: SSN of the servicing worker. format:#########
  required: false

serviceProviderTaxPayerId:
  type: string
  description: Tax Identifier of the servicing worker/provider.
  format:#########
  required: false
eVisitMethod:

type: string
enum:
  - Telephone
  - Mobile
  - Fob
  - Other

description: Method by which data was entered into the E.V.V. system (T-Telephone, F-Fob, M-Mobile App, O-Other)
required: false

eexample:

  transactionId: 8ec389ea-dcf6-4584-85ef-8a06a8653ef2
  memberId: 123456AB
  dateOfBirth: 2015-05-23
  providerName: Upstate Hospital LLC.
  nationalProviderId: "1234567890"
  providerId: "12345678"
  taxPayerId: "123456789"
  providerAddress:
    street: Uptown St
    city: Albany
    state: NY
    zip : "12204"
  providerRateCode: "1234"
  procedureCode: A1234
  procedureModCode: ["AB","CD","EF","GH"]
  serviceStartDateTime: 2015-07-04T12:00:00
  serviceEndDateTime: 2015-07-04T13:00:00
  serviceAddress:
    street: 150 Broadway
    city: Menands
    state: NY
    zip : "12204"
  serviceProviderFirstName: George
serviceProviderLastName: Washington
serviceProviderPhoneNumber: "7879849883"
serviceProviderProviderSSN: "123456789"
serviceProviderTaxPayerId: "123456789"
eVisitMethod: Telephone

payload: # A wrapper Object is to used to avoid JSON hijacking.
  type: object
  properties:
    visits: visit[]

error:
  type: !include error.raml

address:
  type: object
  properties:
    street: #addressLine1
      type: string
      description: Building Number and Street Name
    city:
      type: string
      description: City
    state:
      type: string
      description: State
    zip :
      type: string
      description: Zip Code

/claims/submitter/{submitterId}/evv:
  post:
    is: [api-key-required]
    body:
      application/json:
        type: payload
responses:
"200":
  description: OK
"206":
  description: Partial Content
  body:
    application/json:
      type: error
"401":
  description: Unauthorized
  body:
    application/json:
      type: error
"403":
  description: Forbidden
  body:
    application/json:
      type: error
"500":
  description: Internal Server Error

/claims/submitter/{submitterId}/evv/{transactionId}:
get:
  is: [api-key-required]
  description: Retrieve responses:
"200":
  description: OK
  body:
    application/json:
      type: visit
"400":
  description: Bad Request
  body:
    application/json:
type: error
"401":
  description: Unauthorized
  body:
    application/json:
      type: error
"403":
  description: Forbidden
  body:
    application/json:
      type: error
"404":
  description: Not Found
  body:
    application/json:
      type: error
"500":
  description: Internal Server Error
delete:
  is: [api-key-required]
resoses:
  "204":
    description: No Content
  "400":
    description: Bad Request
    body:
      application/json:
        type: error
  "401":
    description: Unauthorized
    body:
      application/json:
        type: error
  "403":

description: Forbidden
body:
  application/json:
    type: error
"404":
  description: Not Found
  body:
    application/json:
      type: error
"500":
  description: Internal Server Error
put:
  is: [api-key-required]
body:
  application/json:
    type: visit
responses:
  "200":
    description: OK
"201":
    description: Created
"400":
    description: Bad Request
    body:
      application/json:
        type: error
"401":
    description: Unauthorized
    body:
      application/json:
        type: error
"403":
    description: Forbidden
    body:
application/json:
  type: error

"500":
  description: Internal Server Error
#%RAML 1.0 DataType

description: Error response

type: object

properties:

  transactionId:
    type: string
    description: Unique transaction id used by the EVV vendor to submit the transaction
    example: "80b755d8-ee01-4bc7-b8a5-5e5b4cc6862c"

error-details:
  type: array
  items:
    description: Error detail
    type: object
    properties:
      code:
        type: string
        description: Error Code
        example: "A435430"

      message:
        type: string
        description: Error Message
        example: "Unable to Process the request"