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 1.4

1/1/2021

Document Number: MMIS.ICD.0022.01.0720
Contract Number: C014305
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>ii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>iii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>...</td>
</tr>
<tr>
<td>1. Purpose of Interface Control</td>
<td>1</td>
</tr>
<tr>
<td>2. Introduction</td>
<td>2</td>
</tr>
<tr>
<td>3. Overview</td>
<td>3</td>
</tr>
<tr>
<td>4. Assumptions/Constraints/Risks</td>
<td>4</td>
</tr>
<tr>
<td>4.1 Assumptions</td>
<td>4</td>
</tr>
<tr>
<td>4.2 Constraints</td>
<td>4</td>
</tr>
<tr>
<td>4.3 Risks</td>
<td>4</td>
</tr>
<tr>
<td>5. General Interface Requirements</td>
<td>5</td>
</tr>
<tr>
<td>5.1 Interface Overview</td>
<td>5</td>
</tr>
<tr>
<td>5.2 Functional Allocation</td>
<td>5</td>
</tr>
<tr>
<td>5.3 Data Transfer</td>
<td>6</td>
</tr>
<tr>
<td>5.4 Transactions</td>
<td>6</td>
</tr>
<tr>
<td>5.5 Security and Integrity</td>
<td>8</td>
</tr>
<tr>
<td>5.6 Operational Support &amp; Service Levels</td>
<td>8</td>
</tr>
<tr>
<td>5.6.1 Service Level Agreements (SLA)</td>
<td>8</td>
</tr>
<tr>
<td>5.6.2 Operational Expectations</td>
<td>8</td>
</tr>
<tr>
<td>6. Detailed Interface Requirements</td>
<td>9</td>
</tr>
<tr>
<td>6.1 Requirements for Electronic Visit Verification (EVV) Data API</td>
<td>9</td>
</tr>
<tr>
<td>6.1.1 General Processing Steps</td>
<td>9</td>
</tr>
<tr>
<td>6.1.2 Interface Processing Time Requirements</td>
<td>10</td>
</tr>
<tr>
<td>6.1.3 File Naming Convention</td>
<td>10</td>
</tr>
<tr>
<td>6.1.4 Message Format (or Record Layout) and Required Protocols</td>
<td>10</td>
</tr>
<tr>
<td>6.1.5 Communication Methods</td>
<td>19</td>
</tr>
<tr>
<td>6.1.6 Security Requirements</td>
<td>19</td>
</tr>
<tr>
<td>7. Qualification Methods</td>
<td>20</td>
</tr>
<tr>
<td>Appendix A – Record of Changes</td>
<td>21</td>
</tr>
<tr>
<td>Appendix B – Acronyms</td>
<td>22</td>
</tr>
<tr>
<td>Appendix C – Glossary</td>
<td>23</td>
</tr>
<tr>
<td>Appendix D – Referenced Documents</td>
<td>24</td>
</tr>
<tr>
<td>Appendix E – Approvals</td>
<td>25</td>
</tr>
<tr>
<td>Appendix F – Contact Information</td>
<td>26</td>
</tr>
<tr>
<td>Appendix G – Additional Appendices</td>
<td>27</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1 – Data Transfer Diagram ................................................................. 6

List of Tables

Table 1 – EVV Request Data Model Table .................................................. 12
Table 2 – Error Data Model Table ................................................................. 17
Table 3 – Error Data Model Table (Provider Attestation) .......................... 17
Table 4 – HTTP Response Status Code to Data Type Mapping Table .......... 18
Table 5 – Record of Changes .................................................................... 21
Table 6 – Acronyms .................................................................................. 22
Table 7 – Glossary ................................................................................... 23
Table 8 – Referenced Documents ............................................................... 24
Table 9 – Approvals ................................................................................ 25
Table 10 – Contact Information ................................................................. 26
1. **Purpose of Interface Control**

The Electronic Medicaid of New York (eMedNY) Project Interface Control Document (ICD) defines the eMedNY system’s Electronic Visit Verification (EVV) interface with EVV Submitters. EVV Submitters are entities that will use the interface described in this document to share information on electronically verified in-home visits on behalf of a New York State Medicaid provider, and may consist of EVV Vendors, Providers, and Managed Care Organizations (MCO). The ICD communicates all inputs and outputs from eMedNY for all potential actions. Its intended audience is project managers, project teams, development teams, and stakeholders interested in submitting data to eMedNY using this interface.
2. Introduction

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 eMedNY 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.

Version 1.4 of this ICD that is published on the eMedNY website is the current and official version of the interface. Other versions in development represent proposed changes until fully approved.

In this document, 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 analysis. 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 the EVV Submitters. EVV Submitters 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.
5.3 Data Transfer
EVV Submitters will initiate a service request over a secure Hyper Text Transfer Protocol (HTTPS) connection to an eMedNY hosted RESTful API. All information will be transferred between both the parties as JSON documents or HTTP Uniform Resource Identifiers (URIs) using REST Design best practices.

![Data Transfer Diagram](image)

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. **Rejected records should be reviewed for accuracy by the EVV Submitter.** Corrected EVV records can be resubmitted via batch or individual submission method.

- Submitters will be able to submit multiple EVV records per submission.
- An EVV Submitter acting as an EVV proxy (Managed Care Organization, Verification Organization, 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>This can be used to delete a transaction</td>
</tr>
<tr>
<td>Get</td>
<td>GET</td>
<td>/claims/submitter/{submitterId} /evv/{transactionId}</td>
<td>visit</td>
<td>error[]</td>
<td>This can be used to retrieve a transaction</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 Key for authentication and authorization to enforce identity verification and service authorization. The provider will be able to obtain the API Key from the API Developer Portal (Accessed using the eMedNY Web Portal ID and password).

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 sections above. 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 or non production environments, contact the eMedNY Call Center at (800) 343-9000.
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 Submitters) and API 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 payloads that meet the service specification (Refer to Open API Specification document).
- API Consumers should accurately use the appropriate URI patterns recommended for each transaction along with HTTP verb.

If you are already enrolled in eMedNY, you may sign up for an eMedNY Web Portal account at https://portal.emedny.org

Your eMedNY Web Portal account credentials can be used to access the following eMedNY API Developer Portal:

Production API Developer Portal: https://developer.emedny.io

Test API Developer Portal: https://developer.emednytest.io

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 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.
6.1.2 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.

6.1.3 File Naming Convention

N/A

6.1.4 Message Format (or Record Layout) and Required Protocols

API Message format is documented and managed through an Open API Specification 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 Open API Specification document. See Appendix G.

6.1.4.1 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.

6.1.4.1.1 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.

6.1.4.1.1.1 Summary of HTTP Response Status Codes

<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>Status Code</td>
<td>Status Code Description</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>201</td>
<td>Created</td>
<td>The request has been fulfilled and resulted in a new resource being created.</td>
</tr>
<tr>
<td>204</td>
<td>No Content</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>206</td>
<td>Partial Content</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>Client Error</td>
<td>The request cannot be fulfilled due to bad syntax. The message requested did not adhere to the structure defined inside the Open API Specification.</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>

### 6.1.4.2 Data Assembly Characteristics

The EVV Service defines a single endpoint for consumption. As this service is a REST service, the interface is governed by an Open API 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 Open API 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>transactionId</td>
<td>string</td>
<td></td>
<td>Min: 1 Max: 150</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>8</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 eMedNY.</td>
</tr>
<tr>
<td>dateOfBirth</td>
<td>date-only</td>
<td></td>
<td>YYYY-MM-DD</td>
<td></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 eMedNY.</td>
</tr>
<tr>
<td>providerName</td>
<td>string</td>
<td></td>
<td>Max: 35</td>
<td></td>
<td>No</td>
<td>Provider Name is the name of a provider of Medicaid services as used on official State records. Provider Name should match the name used on Medicaid claims and encounters. This represents the name of the Billing Provider.</td>
</tr>
<tr>
<td>nationalProviderId</td>
<td>string</td>
<td></td>
<td>10</td>
<td></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).</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>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. The MMIS ID, if populated, should match what is on the claim or encounter that corresponds to the service. 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. This represents the MMIS ID of the Billing Provider.</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). This represents the TaxPayer ID of the Billing Provider.</td>
</tr>
<tr>
<td>providerAddress</td>
<td>Address</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>Providers most current street address, city, state and zip code. This represents the address of the Billing 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>providerRateCode</td>
<td>string</td>
<td></td>
<td>4</td>
<td></td>
<td>Situational</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. Required if Procedure Code is not present. Must be a valid rate code. Applicable billing codes can be found at <a href="https://www.health.ny.gov/health_care/medicaid/redesign/evv/repository/app_billing_codes.htm">https://www.health.ny.gov/health_care/medicaid/redesign/evv/repository/app_billing_codes.htm</a></td>
</tr>
<tr>
<td>procedureCode</td>
<td>string</td>
<td></td>
<td>Min: 5 Max: 5</td>
<td></td>
<td>Situational</td>
<td>Procedure Code for the service rendered to the recipient by the provider. Required if Rate Code is not present. Must be a valid (HCPCS) procedure code. Applicable billing codes can be found at <a href="https://www.health.ny.gov/health_care/medicaid/redesign/evv/repository/app_billing_codes.htm">https://www.health.ny.gov/health_care/medicaid/redesign/evv/repository/app_billing_codes.htm</a></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. Must be a valid modifier, up to 4 occurrences.</td>
</tr>
<tr>
<td>serviceStartDate</td>
<td>datetime-only</td>
<td>YYYY-MM-DDTh:mm:ss</td>
<td></td>
<td>Yes</td>
<td>Begin date/time of the service received by the recipient. Must be a valid date/time. Cannot be greater than the current date (future date). Timestamp is EST.</td>
<td></td>
</tr>
<tr>
<td>serviceEndDate</td>
<td>datetime-only</td>
<td>YYYY-MM-DDTh:mm:ss</td>
<td></td>
<td>Yes</td>
<td>End date/time of the service received by the recipient. Must be a valid date/time.</td>
<td></td>
</tr>
</tbody>
</table>

ICD Version 1.4

eMedNY – EP6720

1/1/2021
<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></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></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>serviceStartLocation</td>
<td>string</td>
<td></td>
<td>Min: 4</td>
<td>Max: 9</td>
<td>Yes</td>
<td>Must be greater than Begin date/time. Cannot be greater than the current date (future date). Timestamp is EST.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>serviceEndLocation</td>
<td>string</td>
<td></td>
<td>Min: 4</td>
<td>Max: 9</td>
<td>Yes</td>
<td>Service start location describes the place where the visit began at the service start time. (Home, Community) Must be a valid service start location.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>serviceProviderFirstName</td>
<td>string</td>
<td></td>
<td>Min: 1</td>
<td>Max: 35</td>
<td>Yes</td>
<td>First name of the servicing worker. This should match employment records maintained by the billing provider. This represents the first name of the caregiver providing the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>serviceProviderLastName</td>
<td>string</td>
<td></td>
<td>Min: 1</td>
<td>Max: 60</td>
<td>Yes</td>
<td>Last name of the servicing worker. This should match employment records maintained by the billing provider. This represents the last name of the caregiver providing the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>serviceProviderPhoneNumber</td>
<td>string</td>
<td></td>
<td>10</td>
<td>9999999999</td>
<td>No</td>
<td>Phone number of the servicing worker. This represents the phone number of the caregiver providing the service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>caregiverId</td>
<td>string</td>
<td></td>
<td>Min: 1</td>
<td>Max: 128</td>
<td>Yes</td>
<td>The Caregiver ID is the ID used to uniquely identify the person providing the service within the Provider's EVV System and/or solution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Object</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>address1</td>
<td>string</td>
<td></td>
<td>Max: 40</td>
<td></td>
<td>Yes</td>
<td>Building Number or Street Line 1</td>
</tr>
<tr>
<td>address2</td>
<td>string</td>
<td></td>
<td>Max: 40</td>
<td></td>
<td>No</td>
<td>Building Number or Street Line 2</td>
</tr>
<tr>
<td>city</td>
<td>string</td>
<td></td>
<td>Max: 25</td>
<td></td>
<td>Yes</td>
<td>City</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>state</td>
<td>string</td>
<td>Max: 2</td>
<td></td>
<td></td>
<td>Yes</td>
<td>State</td>
</tr>
<tr>
<td>zip</td>
<td>string</td>
<td>Min: 5 Max: 9</td>
<td>99999</td>
<td>999999999</td>
<td>Yes</td>
<td>Zip Code</td>
</tr>
<tr>
<td>Submitter ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>submitterId</td>
<td>string</td>
<td>8</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
6.1.4.2.1.1  **Error Data Models**

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>

Table 3 – Error Data Model Table (Provider Attestation)

<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. This applies only when the Provider Attestation is missing or expired, and does not prevent the request from being accepted as long as all other validations are passed.</td>
</tr>
<tr>
<td></td>
<td>providerId or nationalProviderId</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Provider Identifier or National Provider Identifier (NPI) used by the EVV vendor to submit the transaction. When both the Provider ID and National Provider Identifier (NPI) are used by the EVV vendor to submit the transaction, the Provider Identifier will be returned.</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>
6.1.4.3 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 4 – HTTP Response Status Code to Data Type Mapping Table
(Refer to the Electronic Visit Verification Technical User Guide for field validation error messages)

<table>
<thead>
<tr>
<th>HTTP Response Status Code</th>
<th>Data Model</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 Response</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 Response</td>
<td>Response</td>
<td>The request has been fulfilled and resulted in a new resource being created.</td>
</tr>
<tr>
<td>204 Response</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>206 Response</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 Error</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
<tr>
<td>401 Error</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
<tr>
<td>403 Error</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
<tr>
<td>404 Error</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
<tr>
<td>500 Error</td>
<td>Error</td>
<td>Appropriate message will be returned</td>
</tr>
</tbody>
</table>
6.1.5 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, PUT, DELETE, and GET.

6.1.5.1 Interface Initiation

Interface can be invoked by issuing a REST API Call.

6.1.5.2 Flow Control

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

6.1.6 Security Requirements

1) API Key
   a. The HTTP request should have the following key value in the header
      i. api_key = API_KEY issued by eMedNY 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 5 – 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>
<tr>
<td>1.1</td>
<td>7/13/2020</td>
<td>Cyd Hoen-Shears/GDIT</td>
<td>Removal of serviceAddress in lieu of serviceStartLocation and serviceEndLocation. Added sample transactions. Removal of eVisitEdited field</td>
</tr>
<tr>
<td>1.2</td>
<td>8/4/2020</td>
<td>Cyd Hoen-Shears/GDIT</td>
<td>Clarification of fields related to serviceStartDateTime and serviceEndDateTime to indicate Timestamp is EST</td>
</tr>
<tr>
<td>1.3</td>
<td>9/3/2020</td>
<td>Cyd Hoen-Shears/GDIT</td>
<td>Removal of serviceProviderSSN and serviceProviderTaxPayerId in lieu of caregiverId. Clarification of fields related to Billing Provider and Servicing Provider</td>
</tr>
<tr>
<td>1.4</td>
<td>10/1/2020</td>
<td>Cyd Hoen-Shears/GDIT</td>
<td>Addition of Error Data Model for Provider Attestation. Removal of eVisitMethod. Change/addition of address1 and address2</td>
</tr>
</tbody>
</table>
### Table 6 – 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>HTTPS</td>
<td>Hyper Text 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>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 7 – 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 8 – 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>ICD Version 1.4</td>
<td>eMedNY – EP6720</td>
<td>1/1/2021</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 9 – 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>Date</td>
</tr>
<tr>
<td>Name: &lt;Name&gt;, &lt;Job Title&gt; - &lt;Company&gt;</td>
<td>Date</td>
</tr>
<tr>
<td>Name: &lt;Name&gt;, &lt;Job Title&gt; - &lt;Company&gt;</td>
<td>Date</td>
</tr>
<tr>
<td>Name: &lt;Name&gt;, &lt;Job Title&gt; - &lt;Company&gt;</td>
<td>Date</td>
</tr>
<tr>
<td>Name: &lt;Name&gt;, &lt;Job Title&gt; - &lt;Company&gt;</td>
<td>Date</td>
</tr>
</tbody>
</table>

ICD Version 1.4
25
eMedNY – EP6720
1/1/2021
## Appendix F – Contact Information

### Table 10 – 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>eMedNY Call Center</td>
<td>800-343-9000</td>
<td><a href="mailto:emednyproviderservices@gdit.com">emednyproviderservices@gdit.com</a></td>
</tr>
</tbody>
</table>
Appendix G – Additional Appendices

Open API Specification

{
    "openapi": "3.0.0",
    "info": {
        "title": "Electronic Visit Verification API",
        "version": "1.0",
        "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.",
        "termsOfService": "http://developer.emedny.io/portal/terms",
        "contact": {
            "name": "API Support",
            "url": "https://www.emedny.org/contacts/emedny.aspx",
            "email": "api-support@emedny.org"
        }
    },
    "tags": [{
        "name": "New EVV Record",
        "description": "Create a new EVV Record, will overwrite existing records with the same transaction-id and submitter id"
    }, {
        "name": "Batch Load EVV Records",
        "description": "Batch Electronic Visit Verification Entry, Batch size limited to 100 per Call"
    }, {
        "name": "Delete an EVV Record",
        "description": "Delete an existing EVV Record"
    }, {
        "name": "Retrieve EVV Record",
        "description": "Get and existing EVV Record using the transaction Id"
    }],
    "servers": [{

}]}
"description": "Test environment",
"url": "https://api.emednytest.io/emedny"
},

"paths": {
  "/claims/submitter/{submitterId}/evv": {
    "post": {
      "tags": ["Batch Load EVV Records"],
      "responses": {
        "200": {
          "description": "OK"
        },
        "206": {
          "description": "Partial Content",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "/components/schemas/error"
                }
              }
            }
          }
        },
        "401": {
          "description": "Unauthorized",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "/components/schemas/error"
                }
              }
            }
          }
        }
      }
    }
  }
}"description": "Test environment",
"url": "https://api.emednytest.io/emedny"
},

"paths": {
  "/claims/submitter/{submitterId}/evv": {
    "post": {
      "tags": ["Batch Load EVV Records"],
      "responses": {
        "200": {
          "description": "OK"
        },
        "206": {
          "description": "Partial Content",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "/components/schemas/error"
                }
              }
            }
          }
        },
        "401": {
          "description": "Unauthorized",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "/components/schemas/error"
                }
              }
            }
          }
        }
      }
    }
  }
}"description": "Test environment",
"url": "https://api.emednytest.io/emedny"
},

"paths": {
  "/claims/submitter/{submitterId}/evv": {
    "post": {
      "tags": ["Batch Load EVV Records"],
      "responses": {
        "200": {
          "description": "OK"
        },
        "206": {
          "description": "Partial Content",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "/components/schemas/error"
                }
              }
            }
          }
        },
        "401": {
          "description": "Unauthorized",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "/components/schemas/error"
                }
              }
            }
          }
        }
      }
    }
  }
}"description": "Test environment",
"url": "https://api.emednytest.io/emedny"
},

"paths": {
  "/claims/submitter/{submitterId}/evv": {
    "post": {
      "tags": ["Batch Load EVV Records"],
      "responses": {
        "200": {
          "description": "OK"
        },
        "206": {
          "description": "Partial Content",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "/components/schemas/error"
                }
              }
            }
          }
        },
        "401": {
          "description": "Unauthorized",
          "content": {
            "application/json": {
              "schema": {
                "type": "array",
                "items": {
                  "$ref": "/components/schemas/error"
                }
              }
            }
          }
        }
      }
    }
  }
}
"403": {
    "description": "Forbidden",
    "content": {
        "application/json": {
            "schema": {
                "type": "array",
                "items": {
                    "$ref": "#/components/schemas/error"
                }
            }
        }
    }
},
"500": {
    "description": "Internal Server Error"
}

"operationId": "POST_claims-submitter-submitterId-evv",
"requestBody": {
    "content": {
        "application/json": {
            "schema": {
                "$ref": "#/components/schemas/payload"
            }
        },
        "required": true
    },
    "security": [{
        "api_key": []
    }]}
"parameters": [{
   "name": "submitterId",
   "in": "path",
   "required": true,
   "schema": {
      "type": "string"
   }
}]

"/claims/submitter/{submitterId}/evv/{transactionId}": {
   "get": {
      "tags": ["Retrieve EVV Record"],
      "responses": {
         "200": {
            "description": "OK",
            "content": {
               "application/json": {
                  "schema": {
                     "$ref": "#/components/schemas/visit"
                  }
               }
            }
         },
         "400": {
            "description": "Bad Request",
            "content": {
               "application/json": {
                  "schema": {
                     "$ref": "#/components/schemas/error"
                  }
               }
            }
         }
      }
   }
}
"401": {
   "description": "Unauthorized",
   "content": {
      "application/json": {
         "schema": {
            "$ref": "#/components/schemas/error"
         }
      }
   }
},
"403": {
   "description": "Forbidden",
   "content": {
      "application/json": {
         "schema": {
            "$ref": "#/components/schemas/error"
         }
      }
   }
},
"404": {
   "description": "Not Found",
   "content": {
      "application/json": {
         "schema": {
            "$ref": "#/components/schemas/error"
         }
      }
   }
},
"500": {
"description": "Internal Server Error"
}
},
"description": "Retrieve",
"operationId": "GET_claims-submitter-submitterId-evv-transactionId",
"security": [{
  "api_key": []
}
],
"delete": {
  "tags": ["Delete an EVV Record"],
  "responses": {
    "204": {
      "description": "No Content"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/error"
          }
        }
      }
    },
    "401": {
      "description": "Unauthorized",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/error"
          }
        }
      }
    }
  }
}
"put": {
  "tags": ["New EVV Record"],
  "responses": {
    "200": {
      "description": "OK"
    },
    "201": {
      "description": "Created"
    },
    "400": {
      "description": "Bad Request",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/error"
          }
        }
      }
    },
    "401": {
      "description": "Unauthorized",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/error"
          }
        }
      }
    },
    "403": {
      "description": "Forbidden",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/error"
          }
        }
      }
    }
  }
}

"$ref": "/components/schemas/error"

\}
\}
\}

"500": {
   "description": "Internal Server Error"
\}
\}

"operationId": "PUT_claims-submitter-submitterId-evv-transactionId",

"requestBody": {
   "content": {
      "application/json": {
         "schema": {
            "$ref": "/components/schemas/visit"
         }
      }
   },
   "required": true
},

"security": [{
   "api_key": []
}]

"parameters": [{
   "name": "submitterId",
   "in": "path",
   "required": true,
   "schema": {
      "type": "string"
   }
}]


"name": "transactionId",
"in": "path",
"required": true,
"schema": {
   "type": "string"
}
}
"
"components": {
   "schemas": {
      "visit": {
         "example": {
            "transactionId": "8ec389ea-dcf6-4584-85ef-8a06a8653ef2",
            "memberId": "123456AB",
            "dateOfBirth": "2015-05-23",
            "providerName": "Upstate Hostpital LLC.",
            "nationalProviderId": "1234567890",
            "providerId": "12345678",
            "taxPayerId": "123456789",
            "providerAddress": {
               "address1": "Uptown St1",
               "address2": "Uptown St2",
               "city": "Albany",
               "state": "NY",
               "zip": "12204"
            },
            "providerRateCode": "1234",
            "procedureCode": "A1234",
            "procedureModCode": ["AB", "CD", "EF", "GH"],
            "serviceStartDateTime": "2015-07-04T08:00:01",
            "serviceEndDateTime": "2015-07-04T09:00:00",
            "serviceStartLocation": "Home",
          }
        }
    }
}
"serviceEndLocation": "Community",
"serviceProviderFirstName": "George",
"serviceProviderLastName": "Washington",
"serviceProviderPhoneNumber": "7879849883",
"caregiverId": "123456789"
},
"type": "object",
"properties": {
   "transactionId": {
      "minLength": 1,
      "maxLength": 150,
      "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 Complaint ID if available.",
      "type": "string"
   },
   "memberId": {
      "minLength": 8,
      "maxLength": 8,
      "description": "Medicaid Id for the recipient receiving the service.",
      "type": "string"
   },
   "dateOfBirth": {
      "description": "Date of Birth of the recipient receiving the service",
      "type": "string",
      "format": "date"
   },
   "providerName": {
      "minLength": 1,
      "maxLength": 35,
      "description": "Provider Name is the name of a provider of Medicaid services as used on official State records",
      "type": "string"
"nationalProviderId": {
  "minLength": 10,
  "maxLength": 10,
  "description": "National Provider Identifier (NPI) is the nationally recognized provider identifier assigned by the Center for Medicare & Medicaid Services (CMS).",
  "type": "string"
},
"providerId": {
  "minLength": 8,
  "maxLength": 8,
  "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."
},
"taxPayerId": {
  "description": "Tax Identifier of the provider or Federal Employer Identification Number (FEIN). format:#######",
  "type": "string"
},
"providerAddress": {
  "description": "Providers most current address",
  "$ref": "#/components/schemas/address"
},
"providerRateCode": {
  "minLength": 4,
  "maxLength": 4,
  "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."
},
"type": "string"
"procedureCode": {
  "minLength": 5,
  "maxLength": 5,
  "description": "Procedure Code for the service rendered to the recipient by the provider",
  "type": "string"
},

"procedureModCode": {
  "description": "Array of Two character string modifying the procedure code for the service rendered to the recipient by the provider",
  "type": "array",
  "items": {
    "type": "string"
  }
},

"serviceStartDateTime": {
  "description": "Begin date timestamp of the recipient receiving the service.",
  "type": "string"
},

"serviceEndDateTime": {
  "description": "End date timestamp of the recipient receiving the service.",
  "type": "string"
},

"serviceStartLocation": {
  "minLength": 4,
  "maxLength": 9,
  "description": "Service start location describes the place where the visit began at clock-in (Home or Community).",
  "enum": ["Home", "Community"],
  "type": "string"
},

"serviceEndLocation": {
  "minLength": 4,
"maxLength": 9,
"description": "Service end location describes the place where the visit concluded at clock-out (Home or Community).",
"enum": ["Home", "Community"],
"type": "string"
},
"serviceProviderFirstName": {
    "minLength": 1,
    "maxLength": 35,
    "description": "First name of the servicing worker",
    "type": "string"
},
"serviceProviderLastName": {
    "minLength": 1,
    "maxLength": 60,
    "description": "Last name of the servicing worker",
    "type": "string"
},
"serviceProviderPhoneNumber": {
    "description": "Phone Number of the service worker. format:##########",
    "type": "string"
},
"caregiverId": {
    "minLength": 1,
    "maxLength": 128,
    "description": "The Caregiver ID is the ID used to uniquely identify the person providing the service within the Provider’s EVV System and/or solution. format:##########",
    "type": "string"
}
},
"required": ["transactionId", "memberId", "dateOfBirth", "taxPayerId", "serviceStartDateDateTime", "serviceEndDateDateTime", "serviceStartLocation", "serviceEndLocation", "serviceProviderFirstName", "serviceProviderLastName"]
Appendix G

---

ICD Version 1.4

---
"type": "string",
"description": "Error Message",
"example": "Unable to process the request"
"required": ["address1", "city", "state", "zip"]
}
,"responses": {},
"examples": {},
"requestBodies": {},
"headers": {},
"securitySchemes": {
   "api_key": {
      "type": "apiKey",
      "name": "api_key",
      "in": "header"
   }
},
"links": {},
"callbacks": {}
}
**Sample Transactions**

**POST Request – Batch**

Use Case – I want to submit a batch EVV Entry

NOTE: There are only two sample records in this batch; both have a unique Transaction ID

https://api.emednytest.io/emedny/claims/submitter/23456789/evv

<table>
<thead>
<tr>
<th>Base URI</th>
<th>Submitter ID</th>
<th>Submitted Transaction ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test: <a href="https://api.emednytest.io">https://api.emednytest.io</a></td>
<td>23456789</td>
<td>8ec389ea-dcf6-4584-85ef-8a06a8653ef2</td>
</tr>
<tr>
<td>Production: <a href="https://api.emedny.io">https://api.emedny.io</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
{
  "visits": [
    {
      "transactionId": "8ec389ea-dcf6-4584-85ef-8a06a8653ef2",
      "memberId": "123456AB",
      "dateOfBirth": "2015-05-23",
      "providerName": "Upstate Hospital LLC.",
      "nationalProviderId": "1234567890",
      "providerId": "12345678",
      "taxPayerId": "123456789",
      "providerAddress": {
        "address1": "Uptown St",
        "address2": "Suite 100",
        "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",
      "serviceStartLocation": "Home",
      "serviceEndLocation": "Community",
      "serviceProviderFirstName": "George",
    }
  ]
}
```
"serviceProviderLastName": "Washington",
"serviceProviderPhoneNumber": "7879849883",
"caregiverId": "123456789"
},

{
"transactionId": "8ec389ea-dcf6-4584-85ef-9a06a8653gh3",
"memberId": "123456CD",
"dateOfBirth": "2010-06-23",
"providerName": "Upstate Hospital LLC.",
"nationalProviderId": "1234567890",
"providerId": "12345678",
"taxPayerId": "123456789",
"providerAddress": {
  "address1": "Uptown St",
  "address2": "Suite 100",
  "city": "Albany",
  "state": "NY",
  "zip": "12204"
},
"providerRateCode": "1234",
"procedureCode": "A2345",
"procedureModCode": [
  "AB",
  "CD",
  "EF",
  "GH"
],
"serviceStartDateTime": "2015-08-04T12:00:00",
"serviceEndDateTime": "2015-08-04T13:00:00",
"serviceStartLocation": "Home",
"serviceEndLocation": "Community",
"serviceProviderFirstName": "George",
"serviceProviderLastName": "Washington",
"serviceProviderPhoneNumber": "7879849883",
"caregiverId": "123456789"
} ]}
PUT Request – New Entry

Use Case – I want to submit a single EVV Entry

https://api.emednytest.io/emedny/claims/submitter/23456789/evv/8ec389ea-dcf6-4584-85ef-8a06a8655yz2

<table>
<thead>
<tr>
<th>Base URI</th>
<th>Submitter ID</th>
<th>Submitted Transaction ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test: <a href="https://api.emednytest.io">https://api.emednytest.io</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production: <a href="https://api.emedny.io">https://api.emedny.io</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

{
    "transactionId": "8ec389ea-dcf6-4584-85ef-8a06a8655yz2",
    "memberId": "123456GH",
    "dateOfBirth": "2001-04-23",
    "providerName": "Upstate Hostpital LLC.",
    "nationalProviderId": "1234567890",
    "providerId": "12345678",
    "taxPayerId": "123456789",
    "providerAddress": {
        "address1": "Uptown St",
        "address2": "Suite 100",
        "city": "Albany",
        "state": "NY",
        "zip": "12204"
    },
    "providerRateCode": "1234",
    "procedureCode": "A1234",
    "procedureModCode": [
        "AB",
        "CD",
        "EF",
        "GH"
    ],
    "serviceStartDateTime": "2015-07-04T08:00:01",
    "serviceEndDateTime": "2015-07-04T09:00:00",
    "serviceStartLocation": "Home",
    "serviceEndLocation": "Community",
    "serviceProviderFirstName": "George",
    "serviceProviderLastName": "Washington",
    "serviceProviderPhoneNumber": "7879849883",
    "caregiverId": "123456789"
}
PUT Request – Update Existing Entry

Use Case – I want to submit an update to a single EVV Entry

NOTE: The previous PUT example was used, and the changed data is highlighted

https://api.emednytest.io/emedny/claims/submitter/23456789/evv/8ec389ea-dcf6-4584-85ef-8a06a8655yz2

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

```
{
  "transactionId": "8ec389ea-dcf6-4584-85ef-8a06a8655yz2",
  "memberId": "123456GH",
  "dateOfBirth": "2001-04-23",
  "providerName": "Upstate Hospital LLC.",
  "nationalProviderId": "1234567890",
  "providerId": "12345678",
  "taxPayerId": "123456789",
  "providerAddress": {
    "address1": "Uptown St",
    "address2": "Suite 100",
    "city": "Albany",
    "state": "NY",
    "zip": "12204"
  },
  "providerRateCode": "5678",
  "procedureCode": "A1234",
  "procedureModCode": [
    "AB",
    "CD",
    "EF",
    "GH"
  ],
  "serviceStartDateTime": "2015-07-04T08:00:01",
  "serviceEndDateTime": "2015-07-04T09:00:00",
  "serviceStartLocation": "Home",
  "serviceEndLocation": "Community",
  "serviceProviderFirstName": "George",
  "serviceProviderLastName": "Washington",
  "serviceProviderPhoneNumber": "7897898883"
}
```
"caregiverId": "123456789"
}

**GET Request – Query**

Use Case – I want to retrieve an EVV record from eMedNY

NOTE: The submitter ID and the transaction ID must match the record in eMedNY for this request.

https://api.emednytest.io/emedny/claims/submitter/23456789/evv/8ec389ea-dcf6-4584-85ef-8a06a8655yz2

<table>
<thead>
<tr>
<th>Base URI</th>
<th>Submitter ID</th>
<th>Submitted Transaction ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test: <a href="https://api.emednytest.io">https://api.emednytest.io</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production: <a href="https://api.emedny.io">https://api.emedny.io</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DELETE Request**

Use Case – I want to DELETE an EVV record from eMedNY

NOTE: The submitter ID and the transaction ID must match the record in eMedNY for this request.

https://api.emednytest.io/emedny/claims/submitter/23456789/evv/8ec389ea-dcf6-4584-85ef-8a06a8655yz2

<table>
<thead>
<tr>
<th>Base URI</th>
<th>Submitter ID</th>
<th>Submitted Transaction ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test: <a href="https://api.emednytest.io">https://api.emednytest.io</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production: <a href="https://api.emedny.io">https://api.emedny.io</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>