Sandata Specification User Guide

EVV Vendor REST API v1.0

August 18, 2021
Sandata Technologies, LLC
sandata.com

Clella Newcomb, Director Implementation Services
cnewcomb@sandata.com
516.484.4400 X4324

Table of Contents

Revision History

Date	Version #	Author	Description of the Changes
8/18/2021	1.0	Clella Newcomb	Document Creation
11/15/2021	1.1	Clella Newcomb	Update Sample Visit transmission JSON for Task child segment

Specification User Guide – REST API

Overview

This specification user guide is intended to document the requirements for using the Sandata Real Time Interface for receiving information into the Sandata Aggregator.

A program specification is created for each customer implementation to specify agreed upon frequencies, expected values for fields and those fields which will be omitted or left to the sender's discretion.

This document contains technical information and examples for each entity type.

Intended Audience

The intended audience of this document is:

- Project Management and Technical teams at the state, EVV vendors or agencies who will be implementing this interface.

Real-Time Transactions

Data may be sent via a real-time RESTful API for processing. Sandata will take each request as it is received, process the data, and return a response.

Sandata will provide real-time RESTful API endpoints for the state, EVV vendors or agencies in a UAT environment for user acceptance testing as well as production. This document contains the technical details for utilizing this API. The API is designed to be a service-oriented architecture (SOA). All transactions will utilize the JSON format which is the JAVA equivalent to XML. JSON, like XML is self-describing. A WADL (equivalent to the WSDL) will be provided using the API documentation provided by Swagger.

State, EVV vendors or agencies must be able to send and consume data and responses in a JSON format. JSON allows multiple 'child' entities for a parent *(See JSON Example: Request Payload Example in Appendix)*

NOTE: For testing purposes, generic, de-identified files will be provided, or data for testing will be identified by the state, EVV vendors or agencies based on available or constructed data. For payers, testing these files will be part of the overall system testing process per mutually agreed upon dates determined for joint testing and included in the overall project plan.

Representational State Transfer (REST) Interface

Sandata has developed a RESTful interface that allows for a client to send data as real time transactions with appropriate responses rather than in batches of text files for periodic processing.

In a Sandata RESTful web service, requests made to a resource's URL will elicit a response with a payload formatted in JSON. The response can confirm that some alteration has been made to the stored resource, and the response will provide any errors that may have occurred. When HTTP is used for processing members, you will only need to execute a POST HTTPS request method.

HTTPS (TLS 1.2)

Sandata's RESTful interfaces support TLS v1.2 (a successor to SSL) which provides a layer of security and reliability by exchanging PHI information as encrypted data packets between Sandata and other state, EVV vendors or agencies systems.

Real-Time Processing Interfaces

The real-time processing interfaces refer to Sandata's RESTful HTTPS endpoints for receiving data. Sandata will provide the state, EVV vendors or agencies with URL endpoints for UAT and Production. Sandata will also provide the state, EVV vendors or agencies with a username and password that will use Basic Authentication to validate the request. The state, EVV vendors or agencies will receive a 401 HTTP error code if the username and/or password does not match.

Transmission Method

Sandata supports an SOA architecture. Sandata will provide an API for state, EVV vendors or agency's internal IT organizations to utilize. Sandata will provide sample JSON or XML format information (Java equivalent to XML), as well as the WADL (JSON equivalent of the WSDL) to those parties developing the interface. This specification will include the rest endpoints needed to request status on record acceptance /rejection.

See Appendix for sample transmissions.

Transmission Frequency

For optimal system performance, it is recommended that specification data be sent in near real time. It is expected that information is sent as it is added/changed/deleted in the corresponding EVV Vendor system. Rejection responses will be delivered in a separate API call that is initiated by the EVV Vendor–in near real time.

Transmission Limits

A single transaction may contain from a minimum of 1 to a maximum of 5,000 records. A single record set would include all associated elements.

If the group size exceeds the maximum limit for the group, the entire transaction will be rejected.

During peak loads, records received may be queued and processed as resources permit. Other transactions received for the specification will be queued and processed in order.

Data Type Format Details

The user will send information in JSON or XML format. JSON and XML allow multiple ”child” entities for a single parent.

The format of the information sent must match exactly the format defined in the specification and be sent via web service using JSON or XML. Ultimately, we support only three data types during transmission: string, number, and Boolean. The specification references additional data types to ensure that data is received in the expected formats and appropriate record level editing can be incorporated. Except where numeric, the assumed JSON and XML format should be string. The data type provided in the specification is based on the following field definitions.

See Appendix for samples transmissions.

Note that the format is case sensitive. All field names must be provided in EXACTLY the casing used in the definitions below. Sandata recommends using RESTful services with JSON formatting.

Data Type	Description	Example
DateTime	The date and time is represented as a string with the following format: YYYY-MM-DDTHH:MM:SSZ All times will be provided in UTC. If time is not material, provide as expected.	2016-12-20T16:10:28Z

Date (only Date)	The data is represented as a string with the following format: YYYY-MM-DD Date only will be sent in UTC format.	2016-12-20
Timezone	All time for tracking visits will be in UTC. The Time zone name expected in each transaction is the actual Time zone where the event took place. (i.e. US/Eastern)	A complete list of time zones can be found in the appendix of this document.
String	A string is a row of zero or more characters which can include letters, numbers, or other types of characters as a unit, not an array of single characters. (i.e. plain text).	"This is a string" (See Wikipedia String)
Integer	An integer is a numeric value without a decimal. Integers are whole numbers and can be positive or negative.	52110 (positive) -87721 (negative) (See Wikipedia Integer)
Decimal	A floating point number is referred to as a decimal. Can be positive or negative.	8221.231 (positive) -71.214 (negative) (See Wikipedia Decimal)
Boolean	A logic predicate indicator that can be either true or false.	True False See Wikipedia Boolean

RESTful API Response

The response is contained as part of the "data" entity which is part of the standard Sandata HttpResponse entity. This response may be augmented over time to contain additional information. Consumers of the API should be able to handle responses that contain additional data elements.

- **id** – This field is a RESTful service transaction globally unique ID (UUID) which is generated by Sandata. Please log this UUID as it will help Sandata Tier3 support and troubleshoot any issues.
- **status** – This status has two possible values:
    - SUCCESS: Indicates that the request was received and processed successfully by the Sandata backend.
    - FAILED: Indicates that there was some error detected by the Sandata backend. E.g. 500 Server Error
    - NOTE: Both of these states are returned with an HTTP 200 response code.
- **messageSummary** – This field This field will contain either null for status=SUCCESS or "Parameter Error" for status=FAILED. This would typically occur for a "POST" without BODY.
- **messageDetail** – This field will contain either null for status=SUCCESS or a detailed service error message for status=FAILED. E.g. "Database Unavailable"
- **failedCount** – the number of items in the request that resulted in some error

- **succeededCount** – the number of items in the request that ended in a successful result
- **data** – This entity will contain details of the JSON response. Examples can be provided upon request.

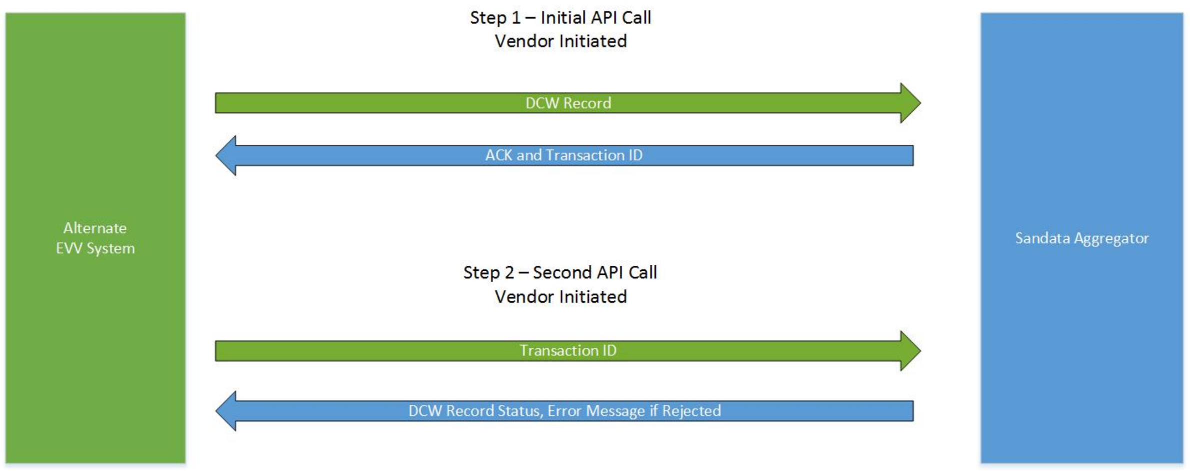
Record Processing

New Record and Updates

New records and updates for previously sent data should be provided using the interface specification. If a set of records is sent (for example: client, employee, or visit), all associated applicable elements should be sent. Partial updates will be rejected. An update that deletes a record will not actually remove information since Sandata will not physically delete information. The deleted record/s will no longer be visible on the application. However, the record history will maintain the original data received.

Rejected Record Process

When records are received, Sandata will return against each group a transaction ID and an ACK (acknowledgment of receipt). This transaction ID can be queried by the caller for status of the records in the transaction. This process will allow the provider/vendor to get status on any of the records that may have been rejected. This process does not apply to Claims Validation. The message flow example below is for an employee record.



Specification Reading Tips

- Segments are denoted as required or optional in the applicable specification and are a named field for transmission.
  - Example EVV Vendor SegmentName: Provider Identification
- The fields listed in the specification are the fields available for transmission to Sandata EVV.
- All data elements/fields in the specification will indicate if NULL is an acceptable value in the Validation Rule column. If the element does not state that NULL is an acceptable value, that element is required.
- Required fields must have data, otherwise the system will reject the record.
- The file will be rejected if a header column name is unknown to the Sandata system.
- If a field is not required, it does not need to be included.
- The Field Information element will be required as part of the header information provided for all EVV Vendor transmissions. This information will be compared to the connection being used within the interface to ensure that the transmission is appropriate. If this match cannot be validated, the transmission will be rejected. Field Information will be *JSON Employee* | *JSON Client* | *JSON Visit*.
- As part of the implementation process, required fields may be adjusted and the available fields may be reduced based on the program specifics.
- All JSON element format is case sensitive. All field names must be provided in EXACTLY the casing used in the definitions below. Specifically, this includes reference values (String match) denoted in "double quotes".

See Appendix for sample transmissions.

Detail Specification Record Logic

Standard date/time Format

All dates and times provided must be sent in UTC (Coordinated Universal Time) format in GMT.

EVV Vendor

EVV Vendor Data Transmission Rules

The following rules apply to information received through this interface. For all rules that result in a rejection, it is expected that the issue will be resolved in the EVV Vendor system and the information subsequently retransmitted.

- There is one set of Interfaces per Sandata Provider Agency ID.
- There will be 3 independent types of data provided:
  - Clients
  - Employees (Field Staff)
  - Visit Information
- Each will be sent individually but can be delivered through the same single connection.
- Visit Information can have up to 2 Calls segments per transmission. Visit Information that have more than 2 Calls segments will be rejected.

Alternate Data Collection System Responsibility

- Visit transmittals
  - Visits should be transmitted near real time. Actual payer frequency requirements may vary.
  - Note that rejection responses will be delivered as separate API calls initiated by the EVV Vendor.
  - Information should be sent for only those records that are added, changed, or deleted. This is considered to be an incremental transmission.
  - Records which have not changed should not be resent.
- Complete transmissions
  - When sending a client, all applicable elements and sub elements must be sent during each transmission.
  - When sending an employee, all applicable elements and sub elements must be sent during each transmission.
  - When sending a visit, all applicable elements and sub elements must be sent during each transmission.
- Call matching
  - Calls received--regardless of the collection method used by the Alternate Data Collection System--are received together into a complete visit by the Aggregator, per the specification.
  - Sandata will not attempt to match or rematch the visits received.
- Data quality
  - All data will be accepted from EVV Vendors "as is," including any calculated fields.

- Latitude and Longitude (CallType = Mobile)
  - Alternate EVV Data Collection Systems are responsible for providing latitude and longitude on all client addresses provided.
  - Latitude and longitude must be provided for both the visit start and visit end time, assuming it is collected via a GPS-enabled device.
- Assigning sequence numbers
  - For each of the 3 types of records (client, employee, visit), the Alternate Data Collection System will be responsible for assigning sequence numbers for each interface to ensure that updates are applied in the appropriate sequence.
  - If a record is rejected, an incremented sequence is expected on the next transmission of that record set.
  - Sequence numbers are per unique record (client, employee, visit) and record set (modifications to the same client, employee, visit).
    - For example, the first time a particular client is sent, the sequence would be set to 1. The second time that same client is sent, the sequence would be set to 2, etc.
- Having the ability to correct defined exceptions
  - Exceptions must be corrected using the standard set of reason codes provided by Payer/State.
  - Some of the defined reason codes require additional text to provide additional information; this information must also be sent as part of this interface.
- Change log transmission
  - Changes made to all visit information must be fully logged, and the log information must be transmitted as part of the visit record, as applicable.
- Using standard date/time format
  - All dates and times provided must be sent in UTC (Coordinated Universal Time) format in GMT.

EVV Vendor Data Processing Rules

- If a record is received and any required data is missing, malformed, or incomplete as defined in the specification, the record will be rejected or set to default values in accordance with the detailed specifications.
- If an optional field is provided with an invalid value (one not listed in this specification), the field will be set to the default value, null and/or rejected, unless otherwise specified in this specification.
- If text (string) field length is longer (>/greater than) than the maximum allowed for that field value, unless otherwise noted, the field will be truncated to the maximum length specified for that field.
- Any record without a sequence number will be rejected. Sequence numbers are per unique record (client, employee, visit). For example, the first time a particular

client is sent, the sequence would be set to 1. The second time the same client is sent, the sequence would be set to 2, etc.

- Records will be processed in the order received using the assigned sequence number.
- If a record that has been received has a sequential number that is less than the one already processed, it WILL BE PROCESSED, but will be logged as “received” and inserted into history. It will not be considered to be the current record.
- Header information as determined for the payer and program must be included in each transmission for each record (client, employee, visit), otherwise the entire collection of records will be rejected.

EVV Vendor Client Data Rules

- The following represents a subset of the requirements for client information. Please see the program specification for all applicable element expectations.
  - If the client does not include at least 1 complete address (address line 1, city, state, zip code) the client will be rejected.
  - If the client does not include the defined unique identifier, the client will be rejected.
  - If the client does not include a Client Other ID (external ID) and Sequence ID, the client will be rejected.
  - If the client does not include first name, last name and time zone, the client will be rejected.

EVV Vendor Employee Data Rules

- The following represents a subset of the requirements for employee information. Please see the program specification for all applicable element expectations.
  - If Staff Other ID (External ID), Sequence ID and Staff ID are not provided, the employee will be rejected.
  - If employee first name and last name are not provided, the employee will be rejected.

EVV Vendor Visit Data Rules

- Clients and Employees should be sent before visits, to ensure they exist in the Sandata system at the time of visit receipt.
  - No Client Provided - To allow the Aggregator to determine if the visit is for a Payer/State client, the visit must include a client. If a visit does not include a client, the complete visit will be rejected.

    - Invalid/Unknown Client Provided - To allow the Aggregator to determine if the visit is for a Payer/State Client, the visit must include a valid client associated with the payer. If a visit includes a client that is unknown to Sandata (has not been received and accepted), the complete visit record will be rejected.
    - No Employee Provided / Invalid or Unknown Employee Provided - If a visit does not include an employee (visit record sent without an employee associated), the visit will be accepted and the 'Unknown Employee' exception will be calculated and applied. This record is accepted but raises an exception.
- The Alternate EVV system is expected to be able to handle a visit that crosses calendar days.

- A visit can only be cancelled if it does not have any calls associated with it or any adjusted times. If a visit has calls but is being cancelled in the source EVV system, the "Bill Visit" indicator should be set to False to indicate that the visit should be disregarded for billing purposes. The visit status will be set to Omit by the Aggregator.

- The following rules apply to the dates and times provided for the visit:

Where Date and Times are provided for the following:

Call In	Call Out	Adjusted In	Adjusted Out	Rule
x	x			**Call Out must be > Call In Otherwise record rejected.**
Superseded by Adj. In	Superseded by Adj. Out	x	x	**Adj. Out must be > Adj. In Otherwise record rejected.**
x	Superseded by Adj. Out		x	**Adj. Out must be > Call In Otherwise record rejected.**
Superseded by Adj. In	x	x		**Call Out must be > Adj. In Otherwise record rejected.**

- Upon receipt, Sandata will calculate all configured Payer/Program exceptions and apply those exceptions as applicable. For those exceptions that may be recalculated over the life of the visit, these exceptions will be calculated as appropriate.

- It is assumed that there are some exceptions that cannot be "fixed" in the Alternate Data Collection System by their nature. They are configured for the Payer/State program as requiring acknowledgement by the system user. One of the included visit elements provides the ability for the user to send their acknowledgement. These exceptions require attestation that the exception has been reviewed/acknowledged in the system along with the appropriate reason code and attestation that appropriate documentation exists. Exceptions are specific to a given Payer/Program and will be noted in the specification.
- Upon receipt, Sandata will calculate and apply visit status as defined for the Payer/Program.
- The Alternate Data Collection System will be expected to send a reason code and optionally the defined resolution code if it applies to the payer. Based on the definitions of the reason codes, some reason codes require additional information explaining the change. If additional information is required, the alternate data collection system must collect the information and include it when transmitting the visit to Sandata.

EVV Vendor Sequencing

The SequenceID on all three types of records (clients, employees, visits) should be independent per record and should be incremented each time any record is sent. The Sequence ID will be used to ensure that a record is processed only once and that the most current information is used for reporting and claims processing. In the event a visit update is not accepted (rejected), the SequenceID on that transmission should not be reused. The next update should increment to the next number in the sequence. Failure to do so will cause the new record to be rejected as a duplicate.

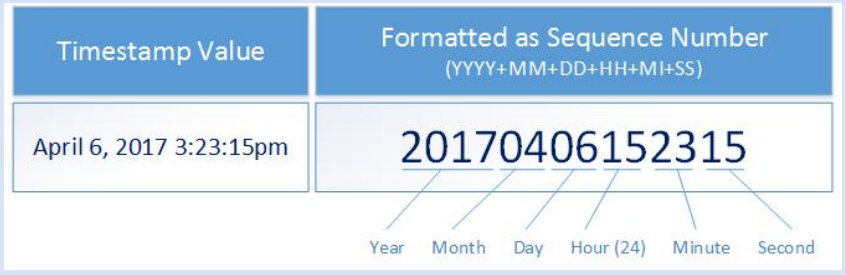
EVV Vendor Sequence Rules:

- If the latest SequenceID is greater than the highest value previously received, the record set will not be rejected. i.e. latest SequenceID = 5, previous SequenceID = 4 ➔ Record accepted and latest record is displayed.
- If the latest SequenceID is less than the value previously received, and the record has not yet been processed, it will be accepted and recorded as historical information. i.e. latest SequenceID = 8, previous SequenceID = 10 ➔ Record accepted and latest record is still SequenceID = 10.
- If the Sequence ID is equal to a value previously received, it will be rejected. i.e. latest SequenceID = 15, previous SequenceID = 15 ➔ Record rejected.
- Gaps in sequence will be allowed.

Please Note:

For those agencies that wish to use the Alternate EVV interface, and would prefer to use timestamps as the sequence number in their deliveries, the Sandata system can accept the timestamp value as the sequence number, under two conditions:

1. *The timestamp value provided must contain only numbers, and no other symbols (i.e. "/", "-", and ":" characters removed)*
2. *The timestamp value provided must be formatted as YYYYMMDDHHMMSS. For example:*



Timestamp Value	Formatted as Sequence Number (YYYY+MM+DD+HH+MI+SS)
April 6, 2017 3:23:15pm	20170406152315

Year
Month
Day
Hour (24)
Minute
Second

EVV Vendor Message Acknowledgement (ACK) and Transaction ID

Index	Column Name	Description	Max Length	Type
	AgencyIdentifier	Unique identifier for the agency.	10	String
	ProviderID	Unique identifier for the agency.	64	String
	TransactionID	Unique identifier for the request generated by the payer.	50	String
	Reason	Default and only value provided: "Transaction Received"	250	String

EVV Vendor Response for Record Status

Index	Column Name	Description	Max Length	Type
	AgencyIdentifier	Unique identifier for the agency.	10	String
	ProviderID	Unique identifier for the agency.	64	String
	RecordType	Type of record that was rejected	10	String

		Values: Client, Employee, Visit		
	RecordOtherID	Value of the record identifier	50	String
	Reason	Default and only value provided: “Transaction Received”	250	String

Technical Companion and Examples

API Location

The RESTful APIs can be reached at the following locations:

- EVV Vendor:
    - The endpoints accept JSON data and support the HTTP POST method.
    - Production:
        - https://api.sandata.com/interfaces/intake/clients/rest/api/v1.1
        - https://api.sandata.com/interfaces/intake/employees/rest/api/v1.1
        - https://api.sandata.com/interfaces/intake/visits/rest/api/v1.1
    - UAT:
        - https://uat-api.sandata.com/interfaces/intake/clients/rest/api/v1.1
        - https://uat-api.sandata.com/interfaces/intake/employees/rest/api/v1.1
        - https://uat-api.sandata.com/interfaces/intake/visits/rest/api/v1.1
    - In addition, WADL documents describing the APIs can be found at the following locations:
        - Production:
            - https://api.sandata.com/interfaces/intake/clients/rest/api/v1.1/wadl
            - https://api.sandata.com/interfaces/intake/employees/rest/api/v1.1/wadl
            - https://api.sandata.com/interfaces/intake/visits/rest/api/v1.1/wadl
        - UAT:
            - https://uat-api.sandata.com/interfaces/intake/clients/rest/api/v1.1/wadl
            - https://uat-api.sandata.com/interfaces/intake/employees/rest/api/v1.1/wadl
            - https://uat-api.sandata.com/interfaces/intake/visits/rest/api/v1.1/wadl

The endpoints accept JSON data and support the HTTP POST method.

A Sandata implementation manager will denote which API is appropriate to use given the requirements of the program.

Authentication Header

The API endpoints utilize Basic Authentication. Therefore, a valid "Authorization" header must be sent with each request. This header is simply a Base 64 encoded representation of the username and password in the format "username:password".

The credentials are determined and distributed during implementation.

An example header for “user@example.com” with password “secret” would be:

Authorization: Basic dXNlckBleGFtcGxlLmNvbTpzZWNyZXQ=

Account Header

In addition to the data endpoint header, a header denoting the callers EVV “Account” must be sent. The credentials provided are specific to an account, and all data sent must also correspond to that account, or the request will be rejected.

An example of this header would be:

Account: 12345

Alternatively, for MCO customers and other vendors sending data on behalf of multiple EVV accounts, the “EntityGuid” header is used. This ID will be provided by Sandata during implementation.

An example of this header would be:

EntityGuid: 12345

Content-Type Header

As with all RESTful API requests, the “Content-Type” header should also be included:

Content-Type: application/json

Suggested Workflow

Interacting with the RESTful API is a two-step process:

- Step 1 – Send a POST request with the data to the API
- Step 2 – Utilize the GET “Status” API to check that processing completed successfully

If the call for Status check results in a messageSummary of "The result for the input UUID is not ready yet. Please try again.", then the sender process must "sleep" and recheck Status until the Status API call returns a messageSummary of either "All records updated successfully." Or …"Records uploaded, please check errors/warnings and try again."

Details are as follows:

The first step is to POST the data being sent to the URL mentioned above in the "API Location" section. When data is sent, the Sandata system will validate the input meets the business requirements, process the data, and return a response.

The response sends back some key pieces of information. This includes any errors that may have been flagged, as well as a UUID, generated by Sandata, which uniquely identifies the request. See example responses below in the "Sample Responses" section.

After this response is sent, the Sandata system begins processing the data into the system. Since the initial POST has already received a response, callers must use a second endpoint to check on the status of their request.

To this end, the API is accompanied by an additional endpoint for checking status. This endpoint is reached simply by appending "/status" to the URL in the "API Location" section above. Calls to this endpoint must utilize the HTTP GET method and send in the UUID that is returned in the response to the POST call.

An example GET request for status would be sent as follows:

> https://api.sandata.com/interfaces/intake/clients/rest/api/v1.1/status?uuid=8d7c31f7-4a09-41a9-8edd-f9819def58f1

In summary, the caller would POST data to the API, receive a response with a UUID, then utilize the "status" endpoint via GET in order to determine if processing was completed and successful.

An example EVV Vendor workflow when sending employees, clients, and visits would be:

1. Send POST request with employee data; receive UUID.
2. Utilize UUID to query employee "Status" API; if still processing, sleep and recheck.
3. Once "Status" API for employees indicates processing is finished, send POST request with client data; receive UUID.
4. Utilize UUID to query client "Status" API; if still processing, sleep and recheck.
5. Once "Status" API for client indicates processing is finished, send POST request with visit data; receive UUID.
6. Utilize UUID to query visit "Status" API; if still processing, sleep and recheck.
7. Once "Status" API for visits indicates processing is finished, all data has been transmitted.

Appendix

JSON examples:

Request Payload Example

Below find sample POST bodies for each entity, as well as sample responses in both successful and unsuccessful situations. Note that, based on implementation, not all fields are required to be present. In addition, certain implementations may include custom fields that are not represented in the samples. Please refer to the specification for a full set of fields and their details.

JSON Employee

```
[{
        "ProviderIdentification": {
                "ProviderQualifier": "MedicaidID",
                "ProviderID": "123456"
        },
        "EmployeeQualifier": "EmployeeCustomID",
        "EmployeeIdentifier": "999999999",
        "EmployeeOtherID": "999999999",
        "SequenceID": 99811930002,
        "EmployeeLastName": "Employee",
        "EmployeeFirstName": "Test",
        "EmployeeEmail": "dummy@sandata.com",
}]
```

JSON Client

```
[{
        "ProviderIdentification": {
                "ProviderQualifier": "MedicaidID",
                "ProviderID": "123456"
        },
        "ClientID": "96641",
        "ClientFirstName": "Test",
        "ClientMiddleInitial": "T",
        "ClientLastName": "Client",
        "ClientQualifier": "ClientMedicaidID",
        "ClientMedicaidID": "999999999",
        "ClientIdentifier": "999999999",
        "MissingMedicaidID": "False",
        "SequenceID": 99811930002,
        "ClientCustomID": "111111111",
        "ClientOtherID": "2222",
        "ClientTimezone": "US/Eastern",
        "Coordinator": "123",
        "ProviderAssentContPlan": false,
        "ClientPayerInformation": [{
                "PayerID": "57",
                "PayerProgram": "123",
                "ProcedureCode": "123",
                "ClientPayerID": "987654321",
                "ClientEligibilityDateBegin": "2019-01-01",
                "ClientEligibilityDateEnd": "2020-01-01",
                "ClientStatus": "02",
                "EffectiveStartDate": "2019-01-01",
                "EffectiveEndDate": "2020-01-01"
        }],
        "ClientAddress": [{
                "ClientAddressType": "Home",
                "ClientAddressIsPrimary": true,
                "ClientAddressLine1": "36 West 5th Street",
                "ClientAddressLine2": "10th Floor",
                "ClientCounty": "Kings",
                "ClientCity": "Manhattan",
                "ClientState": "NY",
                "ClientZip": "10017",
                "ClientAddressLongitude": -73.4228741,
                "ClientAddressLatitude": 40.7431032
        }],
        "ClientPhone": [{
                "ClientPhoneType": "Home",
                "ClientPhone": "1234567890"
        }],
        "ClientDesignee": [{
                "ClientDesigneeFirstName": "",
                "ClientDesigneeLastName": "",
                "ClientDesigneeEmail": "",
                "ClientDesigneeStatus": "",
                "ClientDesigneeStartDate": "",
                "ClientDesigneeEndDate": "",
                "ClientDesigneeRelationship": ""
        }],
        "ClientResponsibleParty": [{
                "ClientContactType": "Other",
                "ClientContactFirstName": "Test",
                "ClientContactLastName": "Respparty",
                "ClientContactPhoneType": "Mobile",
                "ClientContactPhone": "3478788467",
                "ClientContactEmailAddress": "dummy@sandata.com",
                "ClientContactAddressLine1": "2727 East 29th Street",
                "ClientContactAddressLine2": "Apt 8I",
                "ClientContactCity": "Brooklyn",
                "ClientContactState": "NY",
                "ClientContactZip": "11229"
        }]
}]
```

JSON Visit

```
[{
        "ProviderIdentification": {
                "ProviderID": "123456",
                "ProviderQualifier": "MedicaidID"
        },
        "VisitOtherID": "123456789",
        "SequenceID": 111,
        "EmployeeQualifier": "EmployeeCustomID",
        "EmployeeOtherID": "999999999",
        "EmployeeIdentifier": "999999999",
        "GroupCode": null,
        "ClientIDQualifier": "ClientMedicaidID",
        "ClientID": "111111111",
        "ClientOtherID": "111111111",
        "VisitCancelledIndicator": false,
        "PayerID": "999",
        "PayerProgram": "PRG",
        "ProcedureCode": "T1000",
        "Modifier1": null,
        "Modifier2": null,
        "Modifier3": null,
        "Modifier4": null,
        "VisitTimeZone": "US/Eastern",
        "ScheduleStartTime": "2019-07-28T16:02:26Z",
        "ScheduleEndTime": "2019-07-28T20:02:26Z",
        "ContingencyPlan": "CP01",
        "Reschedule": false,
        "AdjInDateTime": "2019-07-28T15:02:26Z",
        "AdjOutDateTime": "2019-07-28T19:02:26Z",
        "BillVisit": true,
        "HoursToBill": 10,
        "HoursToPay": 10,
        "Memo": "This is a memo!",
        "ClientVerifiedTimes": true,
        "ClientVerifiedTasks": true,
        "ClientVerifiedService": true,
        "ClientSignatureAvailable": true,
        "ClientVoiceRecording": true,
        "Calls": [{
                "CallExternalID": "123456789",
                "CallDateTime": "2019-07-28T16:02:26Z",
                "CallAssignment": "Time In",
                "GroupCode": null,
                "CallType": "Other",
                "ProcedureCode": "T1000",
                "ClientIdentifierOnCall": "111111111",
                "MobileLogin": null,
                "CallLatitude": 40.34455,
                "CallLongitude": -21.99383,
                "Location": "123",
                "TelephonyPIN": 999999999,
                "OriginatingPhoneNumber": "9997779999"
        }],
        "VisitExceptionAcknowledgement": [{
                "ExceptionID": "15",
                "ExceptionAcknowledged": false
        }],
        "VisitChanges": [{
                "SequenceID": "110",
                "ChangeMadeBy": "dummy@sandata.com",
                "ChangeDateTime": "2019-07-25T18:45:00Z",
                "GroupCode": null,
                "ReasonCode": "7227",
                "ChangeReasonMemo": "Change Reason Memo 999",
                "ResolutionCode": "A"
        }],
        "Tasks": [{
                "TaskID": "321",
                "TaskReading": "98.6",
                "TaskRefused": false
```

```
          }]
  }]
```

Sample Responses

See some sample responses below. Note that the samples are provided for employee, but the same pattern is followed for both client and visit.

Employee POST (Successful)

```
{
  "id": "7f6dcd1a-ec5e-4efd-a2d4-1049756016a5",
  "status": "SUCCESS",
  "messageSummary": "The result for the input UUID is not ready yet. Please try 
again.",
  "data": {
    "uuid": "7f6dcd1a-ec5e-4efd-a2d4-1049756016a5",
    "account": "12345",
    "message": "The result for the input UUID is not ready yet. Please try again.",
    "reason": "Transaction Received."
  }
}
```

Employee POST (Validation Error)

```
{
  "id": "ea76e9a1-9b29-4f3d-af1c-6b573eb29b76",
  "status": "FAILED",
  "messageSummary": "[1] Records uploaded, please check errors/warnings and try 
again.",
  "data": [
    {
      "ProviderIdentification": {
        "ProviderID": "123456",
        "ProviderQualifier": "MedicaidID",
        "ErrorCode": null,
        "ErrorMessage": null
      },
      "EmployeeIdentifier": "999999999",
      "EmployeeOtherID": "2222",
      "SequenceID": 99811930002,
      "EmployeeQualifier": "EmployeeCustomID",
      "EmployeeOtherID": "999999999",
      "EmployeeLastName": "Employee",
      "EmployeeFirstName": "Test",
      "EmployeeEmail": "dummy@sandata.com",
      "EmployeeManagerEmail": "dummymanager@sandata.com",
      "EmployeeAPI": "111111111",
      "EmployeePosition": "AKN",
      "ErrorCode": null,
      "ErrorMessage": "ERROR: The EmployeePosition expected format is not correct. The 
record should satisfy this regular expression ['HHA|HCA|RN|LPN|PCA']. Invalid 
Value='AKN'. The record is being rejected."
    }
  ]
}
```

Employee GET (Status)

A sample response to a status GET request that has finished processing is:

```
{
	"id": "73b7a9d7-a79a-45cc-9def-cb789c111f4b",
	"status": "SUCCESS",
	"messageSummary": "All records updated successfully.",
	"data": {
		"uuid": "73b7a9d7-a79a-45cc-9def-cb789c111f4b",
		"account": null,
		"message": "All records updated successfully.",
		"reason": "Transaction Received."
	}
}
```

If the request is not yet finished being processed, the “messageSummary” will be “The result for the input UUID is not ready yet. Please try again.”

```
{
	"id": "873a1d97-0681-402e-8268-b6cad8f2b4b7",
	"status": "SUCCESS",
	"messageSummary": "The result for the input UUID is not ready yet. Please try 
again.",
	"data": {
		"uuid": "873a1d97-0681-402e-8268-b6cad8f2b4b7",
		"account": "12345",
		"message": "The result for the input UUID is not ready yet. Please try 
again.",
		"reason": "Transaction Received."
	}
}
```

If the request was processed but failed business rules, an example status would be:

```
{
  "id": "e5de964b-9803-4051-b89b-8a89926e4983",
  "status": "SUCCESS",
  "messageSummary": "[2] Records uploaded, please check errors/warnings and try 
again.",
  "data": [
    {
      "ProviderIdentification": {
        "ProviderID": "123456",
        "ProviderQualifier": "SandataID",
        "ErrorCode": null,
        "ErrorMessage": null
      },
      "EmployeeIdentifier": "999999999",
      "EmployeeOtherID": "2222",
      "SequenceID": 99811930002,
      "EmployeeQualifier": "EmployeeCustomID",
      "EmployeeOtherID": "999999999",
      "EmployeeLastName": "Employee",
      "EmployeeFirstName": "Test",
      "EmployeeEmail": "dummy@sandata.com",
      "EmployeeManagerEmail": "dummymanager@sandata.com",
      "EmployeeAPI": "111111111",
      "EmployeePosition": "RN",
      "ErrorCode": "-709",
      "ErrorMessage": "Version number is duplicated or older than current"
    }
  ]
}
```

Terminology

Legend

Field Name	Other Possible Naming
Client	Individual
	Member
	Patient
	Recipient
Employee	Caregiver
	Consumer Directed Worker
	Home Health Aide
	Staff
	Worker
Provider	Agency
	Third Party Admin (TPA)
Payer	Admission
	Contract
	Insurance Company
	Managed Care Organization (MCO)
	State
Contract	Program
	Program Code
HCPCS	Bill Code
	Procedure Code
	Service

Acronymns

Term	Definition
AKA	**Also Known As**
API	**Application Programming Interface**
GMT	**Greenwich Mean Time**
HTTP	**Hypertext Transfer Protocol**
JSON	**JavaScript Object Notation**
SOAP	**Simple Object Access Protocol**
SRS	**System Requirement Specifications**
TBD	**To Be Determined**
UTC	**Universal Time Coordinated**
XML	Extensible Markup Language