

DAISEY API Supplement

All Services	3
API User	3
External Id	3
Mandatory Questions and Required Fields	4
Complete and Partially Complete	4
Assumptions.....	4
Request Queue	5
Failed Requests	5
Forms Service.....	5
GET	5
Request	5
Request Examples	5
Response	6
Response Examples.....	6
Form Data Service.....	9
All Form Data Endpoints	9
Form Data Sequences	9
POST	10
Request	10
Question Response-Option Examples.....	10
Branching Logic Question Examples	11
POST Request Validations - <i>Processing validations after being picked up from the requests queue</i> .	11
Request Examples	12
Response	13
Response Examples.....	13
PUT.....	13
Request	13
PUT Request Validations - <i>Processing validations after being picked up from the requests queue</i> ...	14
Request Examples	14
Response	15
Response Examples.....	15

DELETE..... 15

 Request 15

 DELETE Request Validations - *Processing validations after being picked up from the requests queue*
 16

 Request Examples 16

 Response 17

 Response Examples..... 17

Version History

Date	Version
2019/02/13	1.0
2019/02/26	1.1
2019/03/08	1.2
2019/03/27	1.3
2019/04/03	1.4
2019/04/09	1.5
2019/04/17	1.6
2019/05/16	1.7

All Services

- Only one status code (200, 400, etc.) will be returned for each request in the response header.
- The API will only allow access to forms that the organization has access to.
- Each organization will have unique API User credentials.
- Organization access will be determined by the API User details.
- All message bodies in requests and responses will be in JSON format.

API User

- The API User account is used to interact with the DAISEY API, in combination with the API Key.
- There must be one API User for every organization that wants to interact with the DAISEY API.

External Id

- The externalId is a unique, identifying key for each profile or activity.
- For profile forms, the DAISEY Id and system Id are expected to be left blank and any value provided will be ignored.
- Activities will require a unique external id and will need to be associated with a profile when being submitted.
- The externalId value is determined by the organization creating the request for the DAISEY API and is expected to be of type Long.
- Once an externalId is associated with a DAISEY record, it may not be reused for a different record.
- Typically, the externalId value may be a primary key of a table designating the profile or activity record in the source system's database.
- This value is provided by the source system sending data to the DAISEY API and mapped to the form data created in the DAISEY system, creating a link between the data saved in DAISEY and the data saved in the source system.
- Each formData in DAISEY needs to have an externalId including child profile, caregiver profile, child activity, caregiver activity, environment profile, and environment activity.
- For profile type forms, there may be more than one externalId: the externalId of the profile being created, and the externalId of any profile being associated.

- For activity type forms, there will be more than one externalId: the externalId of the activity being created or updated, the externalId of the child involved, and/or the externalId of the caregiver involved. The external Id of the activity is the externalId of the formData being submitted to create or update. The child and caregiver externalId will be provided as a response to a question on the form.

Mandatory Questions and Required Fields

- Required fields are the parameters necessary to make a successful request to the DAISEY API.
- Mandatory questions are the questions marked mandatory on a form. A response to all mandatory questions is necessary to submit a fully complete form data.
- Responses to optional questions have no impact on complete formData versus partially complete formData.
- Since the DAISEY API is designed to accept the minimal amount of data in a request, the questions marked mandatory are not required to have responses saved when submitting a request through the API, except for activity type forms that require to be associated with a profile upon creation.

Complete and Partially Complete

- When a formData is submitted with an answer to all mandatory questions, the formData will be marked as complete. If there is not an answer for every mandatory question, the formData will be marked as partially complete.
- All formData marked as complete will be available in the typical DAISEY UI.
- Partially complete records require an additional workflow to get the formData to be marked as complete. The partially complete formData will need to be updated either through the API or through the Search Partial Form Data UI to mark it as complete.

Assumptions

- For forms with alternateld question, any value provided will be saved, and no validation is performed on alternateld.
- If the provided program does not match any of the programs available for an organization, the value will be ignored. The record and the error will be added to the errors object and reported back in the response.
- If a questionId in a request is not on a form being submitted but is included in the request, the response will be ignored. The externalId and error of the formData record will be added to the errors object and reported back in the response. This error will not prevent other records in the request from being processed.
- Confidentiality mode flags a record as being confidential. For KDHE, this is applicable when a client requests that a Family Planning visit is confidential. Confidentiality mode will default to false if a value is not provided or if the key-value pair for confidentiality mode is not included in the request.
- For data-sharing organizations, it is not possible to update a different organization's data with an API User outside of that organization. This is due to the fact that the external Id is not required to be unique across all organizations.

- For environment profiles, there is a question with id 540. Question id 540 has label “Associated Children For Environment.” This question will allow for multiple externalIds of child profiles to be submitted as the answer for this question and each child’s externalId should be separated by a pipe character.

Request Queue

- Incoming POST, PUT, and DELETE requests will be placed in a single-threaded queue.
- The queue will process the requests in First Come, First Served order to ensure that formData in previous requests has been committed to the DAISEY system before the next request is processed.

Failed Requests

- Requests for POST, PUT, and DELETE, which fail processing when selected from the queue, will be listed out in a grid in the DAISEY UI.
- This grid will have one row per request and a column to display the errors involved in each request.
- Failed validation errors when processing formData selected from the queue will also trigger an email to be sent to detail the errors involved.

Forms Service

URL: /forms

GET

Request

- The GET request to /forms provides what forms the organization can access and specific details about individual forms.
- Providing the formId in the URL is optional. If left out, the service will return a list of formIds. If included, the service will return a specific formStructure.
- A message body is not required, and any supplied message body will be ignored.
- The fields required to make a successful request to /forms service are API Key, API User, and formId (if requesting a form structure)

Request Examples

Description	Example
GET Request to Forms Service for all formIds and Names	GET /forms
GET Request to Forms Service for a single formStructure	GET /forms/{formId}

Response

- The formStructure in the response consists of formId, formName, formType, enableBranchingLogic, programAssociated, confidentialityMode, mandatoryQuestions, and optionalQuestions.
- The Question object in the response consists of questionId, questionLabel, formatTypeCode, questionOptions, questionComment, minimumResponse, maximumResponse, dataTypeCode, and branchingLogic.
- If there is a minimumResponse or maximumResponse value for a question, the expected answer value must fall within the range specified by the minimum and/or maximum response value(s).
- enableBranchingLogic is true for a form when a form has branchingLogic type questions on it.
- branchingLogic type questions are when the available question on a form changes depending on the answer value of a previous question.
- If there are questionOptions for a question, the expected value for the answer is determined by the questionOptions.
- When questionOptions are available, each questionOption is separated by a pipe character “|”
- All explanation type questions will be excluded from returned question lists.

Response Examples

Description	Example
Response to GET /forms Request with No Errors	<pre>{ "message": "", "formIds": [...] }</pre>
Response to GET /forms/123 Request with No Errors	<pre>{ "message": "", "formStructure": {...} }</pre>
Response to GET Request with Error	<pre>{ "message": "Could not find requested form" }</pre>
Response to Forms Service with formStructure Object and an Empty Error Message Object (Abbreviated Question Lists)	<pre>{ "message": "", "formStructure": { "formId": 29, "formName": "KDHE Child Profile", "formType": "Child Profile", "enableBranchingLogic ": true, "programAssociated": true, "confidentialityMode": false, "mandatoryQuestions": [{ "questionId": 216, "questionLabel": "First Name", "formatTypeCode": "Alphanumeric", "questionOptions": null, </pre>

	<pre> "questionComment": "", "minimumResponse": null, "maximumResponse": null, "dataTypeCode": "Text", "branchingLogic ": null }, { "questionId": 217, "questionLabel": "Last Name", "formatTypeCode": " Alphanumeric", "questionOptions": "", "questionComment": null, "minimumResponse": null, "maximumResponse": null, "dataTypeCode": "Text", "branchingLogic": null }, { "questionId": 496, "questionLabel": "Primary Caregiver ID", "formatTypeCode": "Text", "questionOptions": "", "questionComment": null, "minimumResponse": null, "maximumResponse": null, "dataTypeCode": "Alphanumeric", "branchingLogic": null }], "optionalQuestions": [{ "questionId": 604, "questionLabel": "Sex – Select one", "formatTypeCode": "Text", "questionOptions": "Female Male", "questionComment": "", "minimumResponse": null, "maximumResponse": null, "dataTypeCode": "Drop-down list (single choice)", "branchingLogic": null }, { "questionId": 1263, "questionLabel": "Were you born in the United States including the Virgin Islands?", </pre>
--	---

	<pre> "formatTypeCode": "Text", "questionOptions": "1,Yes born in the U.S. 2,No not born in the U.S. 3,Don't know 4,Refused", "questionComment": "", "minimumResponse": null, "maximumResponse": null, "dataTypeCode": "Drop-down list (single choice)", "branchingLogic": null }, { "questionId": 1264, "questionLabel": "If no, what country were you born in?", "formatTypeCode": "Text", "questionOptions": "", "questionComment": "Please indicate the country or enter 'Don't know' or 'Refused'.", "minimumResponse": null, "maximumResponse": null, "dataTypeCode": "Text", "branchingLogic": "q1263==2" }] } } </pre>
formStructure Object	<pre> { "formStructure": { "formId": 29, "formName": "KDHE Child Profile", "formType": "Child Profile", "enableBranchingLogic": true, "programAssociated": true, "enableBranchingLogic": true, "confidentialityMode": false, "mandatoryQuestions": [{...}, {...}], "optionalQuestions": [{...}, {...}] } } </pre>
question Object	<pre> { "questionId": 216, "questionLabel": "First Name", </pre>

	<pre> "formatTypeCode": "Alphanumeric", "questionOptions": null, "questionComment": "", "minimumResponse": null, "maximumResponse": null, "dataTypeCode": "Text", "branchingLogic ": null } </pre>
--	--

Form Data Service

URL: /formData

All Form Data Endpoints

- An invalid JSON object, invalid API User credentials, and/or an invalid API Key will prevent the entire request from being accepted and result in a 400 HTTP Status Bad Request.
- For valid JSON with correct API syntax, objects in the array are processed independently. An error in one object will not prevent successfully processed objects from being committed to the DAISEY system.
- An object with an error will not be processed.
- The validation process for all requests will be the same as what is offered through import and UI.
- For all formData that requires to be created in a specific sequence, these formData need to be created in separate requests to allow the prerequisite form data to be committed first to the DAISEY system. ([See Form Data Sequences](#))

Form Data Sequences

- Profiles
 - Profiles and Family Associations are the building unit for DAISEY. A Family Association is created by associating profiles for caregiver and child. Family Associations require a specific sequence when creating, updating, and deleting.
 - To create a family association, the primary caregiver profile must exist first.
 - Secondary caregiver will contain primary information in the corresponding questions on the add caregiver form.
 - Child will contain primary information in the corresponding questions on the add child form.
 - Child profiles can exist in the system without an associated primary caregiver.
 - Attempting to associate a child or caregiver with a deleted profile will result in an error.
- Activities
 - The Family Association of child and/or primary caregiver must exist before associating any related child activities or caregiver activities.
 - An activity form must be submitted with a profile involved or it will be rejected.
 - The child profile or caregiver profile involved in an activity type form must exist in DAISEY before associating the activity with the child or caregiver.
 - An error will occur if the prerequisite child or caregiver is not in DAISEY before making the request.
- Environment

- The environment profile must exist in DAISEY before submitting an environment activity.
- The child profile must exist in DAISEY before associating a child to an environment profile.

POST

Request

- POST requests create new formData. A POST request for an externalId that already exists in DAISEY will result in an error.
- Each request must include a single formId in the message body to specify which form's data is being submitted in the request.
- A request may submit a single record or multiple records, but only for one form per request.
- Each record in a multiple record request must be an individual formData object inside the message body of the request.
- Each formData object in the request requires an externalId to identify the record and the questionId and answer for any formData being submitted.
- When submitting multiple records in a single request, error free records can still be processed in the same request.
- The answer will not be saved if the answer value is outside of the specified minimum/maximum response values for questions with a min/max values and an error will occur when processing.
- Confidentiality mode can be set to true for each form data object. A value of true will mark the record as confidential. The default value is false if not provided.
- Organizations do not need to include the confidentialityMode key-value pair in the request, defaulting to a value of false.
- Organizations with programs available can use the programsAssociated array to provide programs associated for a form data. If an organization does not have any programs associated, this key-value can be left out of the request.
- All question answer pairs submitted require that the questionId is included.
- Answers must be in the format specified by the dataTypeCode and formatTypeCode.
- Answers can have a set of specific question options to choose from. Answers that can have multiple selected options should be in a list with values separated by a pipe.
([Question Response-Option Examples](#))

Question Response-Option Examples

There are several patterns to question options. To select an option between pipe characters to submit for an answer, use one of the following patterns:

1. When only text is present
 - a. Yes|Not yet but I plan to|No I do not plan to
 - b. The expected value is the entire text value.
2. When a number is present along the text separated by a comma between pipe characters.
 - a. 0,No|1,Yes-Client Accepted Services|2,Yes-Client Declined|3,Yes-Client Lost
 - b. The expected value is the numeric value before the comma.
3. When a number is present along the text separated by a colon between pipe characters.
 - a. 0: No reason|1: Wouldn't think of it|2: Yes
 - b. The expected value is the numeric value and the text, including the colon.

4. When only a number is present.
 - a. 1|2|3|4|5|6
 - b. The expected value is the numeric value.

When there are no question options present, enter the response according to the question format type. All dates should be formatted with the ISO 8601 standard format YYYY-MM-DD.

Branching Logic Question Examples

For a given question, branchingLogic determines when that question becomes available on a form. The branchingLogic for a question will be of type String. The equation will resemble: q3150==6

The equation for a given question, question X, should be read as: “For a given question, question X, with a branchingLogic equation of q3150==6, question X will be available to answer if the response to question id 3150 is equal to 6.”

There are equations for greater than, greater than or equal to, less than, less than or equal to, equal to, not equal to, and between. The equations may be combined by an AND or an OR as well. For example:
 ((q3150=6) and (q3151>=45)) or
 ((q3150=12) and (q3151>=48))

POST Request Validations - *Processing validations after being picked up from the requests queue*

POST Form Data Processing	An error will occur if...
Beginning	Program provided in programsAssociated array does not match one of the list of programs for an organization
Child Profile	externalId already exists
Caregiver Profile	<ol style="list-style-type: none"> 1. externalId already exists 2. Primary caregiver has another primary caregiver (E.g. Answer “Yes” to question “Is this the primary caregiver?” while also providing a value for “If No, Select Primary Caregiver”)
Environment Profile	externalId already exists
Environment Activity	<ol style="list-style-type: none"> 1. The provided externalId of activity already exists 2. The externalId of “Profile Involved” does not exist
Caregiver / Child Activity	<ol style="list-style-type: none"> 1. The provided externalId of activity already exists 2. The externalId of “Profile Involved” does not exist 3. The family association does not exist before submitting an activity with provided “Profile Involved”
End	<ol style="list-style-type: none"> 1. The expected response to a question does not match the formatTypeCode and dataTypeCode provided in request 2. Any text value has special (restricted) characters: \ < \ 3. A question’s answer is out of range of the specified min and max response for a question

	4. A question's answer is not one of the specified questionOptions when there are questionOptions available
--	---

Request Examples

Description	Example
POST Request URL	POST /formData
Complete Form Data Request formId formData Objects	<pre>{ "formId":29, "formData": [{ "externalId":123, "confidentialityMode ": false, "responses": { "216":"John", "217":"Doe", "496":"", "604":"Male", "1263":2, "1264":"France" }, "programsAssociated": [] }, { "externalId":321, "confidentialityMode ": false, "responses": { "216":"Jane", "217":"Doe", "604":"Female", "1263":1 } }] }</pre>
Form Id and Form Data Object of Request	<pre>{ "formId": 29, "formData": [{ "externalId": 123, "confidentialityMode": false, "responses": { "questionId1": "value1",</pre>

	<pre> "questionId2": "value2" }, "programsAssociated ": [] }} }</pre>
--	---

Response

- The formId and JSON syntax are validated before being inserted into the Request Queue. The response object will include an error if the formId or JSON syntax are invalid.
- If the formId and JSON syntax are valid, then the response will be 200 – OK to designate that the request is valid and has entered the processing queue.

Response Examples

Description	Example
Response to Request with No Errors	<pre>{ "message": null }</pre>
Response to Request with Error	<pre>{ "message": "Invalid formId" }</pre>

PUT

Request

- PUT requests must be for form data that already exists in DAISEY.
- Trying to make a PUT request to create a new form data will result in an error. The externalId must already be mapped to an existing form's data to make a PUT request.
- Can make a single request to update a single record or multiple records.
- Each PUT request can only be made for a single formId.
- Each record in a multiple record request should have an individual formData object with a unique externalId to update multiple records for a single formId.
- Each form's data in a request requires an external Id to identify the record and the questionIds for questions being submitted.
- Only the questionIds and the answers that need to be updated are required in the request to update a record.
- To update an answer value in a formData, make a PUT request with the question Id and answer pair that needs to be updated.
- To delete an answer value in a formData, make a PUT request with the questionId and answer pair that needs to be updated and provide an empty string for text and numeric type answers.
- A request with multiple records will not get rejected if one record has an error. When submitting multiple records in a single request, error free records will be processed in the same request.
- Answers that can have multiple selected options should be in a list with individual values separated by a pipe.

PUT Request Validations - *Processing validations after being picked up from the requests queue*

POST Form Data Processing	An error will occur if...
Beginning	<ol style="list-style-type: none"> 1. Program provided in programsAssociated array does not match one of the list of programs for an organization 2. externalId is mapped to a form data with a formId that is different than the form id in the request.
Child Profile	externalId does not exist
Caregiver Profile	<ol style="list-style-type: none"> 1. externalId does not exist 2. Primary caregiver has another primary caregiver (E.g. Answer "Yes" to question "Is this the primary caregiver?" while also providing a value for "If No, Select Primary Caregiver" 3. Secondary caregiver is converted to primary caregiver (E.g. Answer "Yes" to question "Is this the primary caregiver?" while providing an externalId of a Secondary Caregiver) 4. Primary caregiver is converted to secondary caregiver (E.g. Answer "No" to question "Is this the primary caregiver?" while providing an externalId of a Primary Caregiver)
Environment Profile	externalId does not exist
Environment Activity	<ol style="list-style-type: none"> 1. The provided externalId of activity does not exist 2. The externalId of "Profile Involved" does not exist
Caregiver / Child Activity	<ol style="list-style-type: none"> 1. The provided externalId of activity does not exist 2. The externalId of "Profile Involved" does not exist 3. The family association does not exist before updating an activity with provided "Profile Involved"
End	<ol style="list-style-type: none"> 1. The expected response to a question does not match the formatTypeCode and dataTypeCode provided in request 2. Any text value has special (restricted) characters: \ < \ 3. A question's answer is out of range of the specified min and max response for a question 4. A question's answer is not one of the specified questionOptions when there are questionOptions available

Request Examples

Description	Example
Update Request URL	PUT /formData

<p>Complete Form Data Request Object formId formData Objects Updating first and last names and associating caregiver profile with externalId 112233 - (Assuming externalId 112233 exists)</p>	<pre>{ "formId":29, "formData": [{ "externalId":123, "responses": { "216":"Johnny", "217":"Doerty", "496":112233 } }] }</pre>
<p>formId and formData Object of Request</p>	<pre>{ "formId": 29, "formData": [{ "externalId": 123, "confidentialityMode ": false, "responses": { "questionId1": "value1", "questionId2": "value2", "questionId3": "value3" } }] }</pre>

Response

- The formId and JSON syntax are validated before being inserted into the Request Queue. The response will have an error if the formId or JSON syntax are invalid.
- If the formId and JSON syntax are valid, then the response will be 200 – OK to designate that the request is valid and has entered the processing queue.

Response Examples

Description	Example
Response to Request with No Errors	<pre>{ "message": null }</pre>
Response to Request with Error	<pre>{ "message": "Invalid JSON syntax" }</pre>

DELETE

Request

- Can make a single request to Form Data service to delete a single record or multiple records for per externalId.

- The externalIds should be provided in the message body as a JSON with the key “externalIds” and the value as a list of externalIds.
- The externalId is required to delete the record.
- Deleting a record does not remove it completely. Deleting a record will mark the record as not in use/deleted=true in the database.
- Deleting a record will remove the externalId associated with it in the DAISEY system.
- Profiles without an associated activity form can be deleted.
- Profiles with an associated activity form can’t be deleted.
- Any activity form can be deleted.
- Primary caregivers cannot be deleted when there are associated secondary caregivers.
- Secondary caregivers must be deleted before deleting a primary caregiver.
- Deleting a child profile that is associated with an Environment will remove the child’s association to the environment.
- DELETE requests can be for any externalId and is not restricted to delete per form.
- Since requests do not update the database until the entire request has been processed, it is not possible to delete externalIds to remove any associations in one request. For example, it is not possible to delete a child activity’s formData and then the associated child profile’s formData in the same request.

DELETE Request Validations - *Processing validations after being picked up from the requests queue*

DELETE Form Data Processing	An error will occur if...
Child Profile	The child profile is associated with an existing activity
Caregiver Profile	<ol style="list-style-type: none"> 1. The caregiver profile is associated with an existing activity 2. The caregiver profile is associated with a secondary caregiver profile
Environment Profile	The environment profile is associated with an existing activity
Environment Activity	N/A – Any environment activity form data can be deleted
Caregiver / Child Activity	N/A – Any caregiver/child activity form data can be deleted

Request Examples

Description	Example
Delete Request URL	DELETE /formData
Delete Message Body	<pre>{ "externalIds":[1883920, 2817411] }</pre>

Response

- If the formData does not exist/external Id can't be mapped to an existing formData, the response will be status code 200 OK.
- JSON syntax is validated before being inserted into the Request Queue. The response will have an error if the JSON syntax is invalid.

Response Examples

Description	Example
Response to Request with No Errors	<pre>{ "message": null }</pre>
Response to Request	<pre>{ "message": "Invalid JSON syntax" }</pre>