1 Overview 4
1.1 Purpose 4
1.2 Reference URLs 4
1.3 CDMS Hierarchy 5
1.4 CDMS API Summary 8
1.5 Postman Collection 9
1.6 Rate Limits / Caveats 11
1.7 API Roles / Access 12
1.8 Vault Permission Sets / Roles 16
2 Authentication / API Login 19
3 Studies 22
3.1 Retrieve Studies 22
4 Countries 25
4.1 Retrieve Countries 25
4.2 Retrieve Study Countries 27
5 Sites 29
5.1 Retrieve Sites 29
6 Subjects / Casebooks 32
6.1 Retrieve Subjects 32
6.2 Create Subjects (Casebooks) 37
6.3 Set Subject Status 40
6.4 Unset Subject Status 44
7 Event Groups 47
7.1 Create Event Groups 47
7.2 Upsert Event Groups 51
8 Events 57
8.1 Retrieve Events / Forms 57
8.2 Create Events 62
8.3 Set Event Date 64
8.4 Set Event as Did Not Occur 71
9 Forms 74
9.1 Retrieve Forms / Item Data 74
9.2 Create Forms 79
9.3 Upsert Forms 84

9.4 Create Item Groups 87
9.5 Upsert Item Groups 92
9.6 Set/Upsert Form Data (Items) 96
9.7 Submit Forms 114
9.8 Reopen Submitted Forms 118
10 Queries 122
10.1 Retrieve Queries - By Filters 122
10.2 Retrieve Queries - By ID(s) 128
10.3 Retrieve Queries - By Medical Coding Definition 132
10.4 Open Queries - By Study Context 133
10.5 Open Queries - By Item ID 139
10.6 Open Queries - By Event ID 141
10.7 Answer Queries - By Study Context 143
10.8 Answer Queries - By Query ID 145
10.9 Close Queries - By Study Context 147
10.10 Close Queries - By Query ID 149
10.11 Reopen Queries - By Study Context 150
10.12 Reopen Queries - By Query ID 152
11 Medical Coding 154
11.1 Retrieve Coding Requests 154
11.2 Retrieve Coding Queries 162
11.3 Open Coding Queries 166
12 Jobs 168
12.1 Jobs / Data Extracts Overview 168
12.2 Working with Vault FTP Folders Using Vault API 173
12.3 Retrieve Recently Finished Jobs 174
12.4 Start Study Job - SDE 175
12.5 Start Study Job - Subject Progress Listing 179
12.6 Start Study Job - Event Progress Listing 180
12.7 Start Study Job - Form Progress Listing 181
12.8 Start Study Job - Query Detail Listing 182
12.9 Start Study Job - Core Listings 183
12.10 Start Study Job - Data and Definitions Export 185
12.11 Retrieve Job Status 186
12.12 Cancel Job 188

12.13 Retrieve Job Output Log 189
12.14 Retrieve Job Output File 191
13 Users 192
13.1 Retrieve Users 192
13.2 Upload Users 198
13.3 Inactivate User – Vault or Domain Level 201
13.4 Activate User – Domain Level 203
13.5 Activate User – Vault Level 204
14 Study Design 205
14.1 Design Overview 205
14.2 Retrieve Casebook Versions 206
14.3 Retrieve Coding Definitions 207
15 Best Practices / Tips 209

# 1 Overview

## 1.1 Purpose

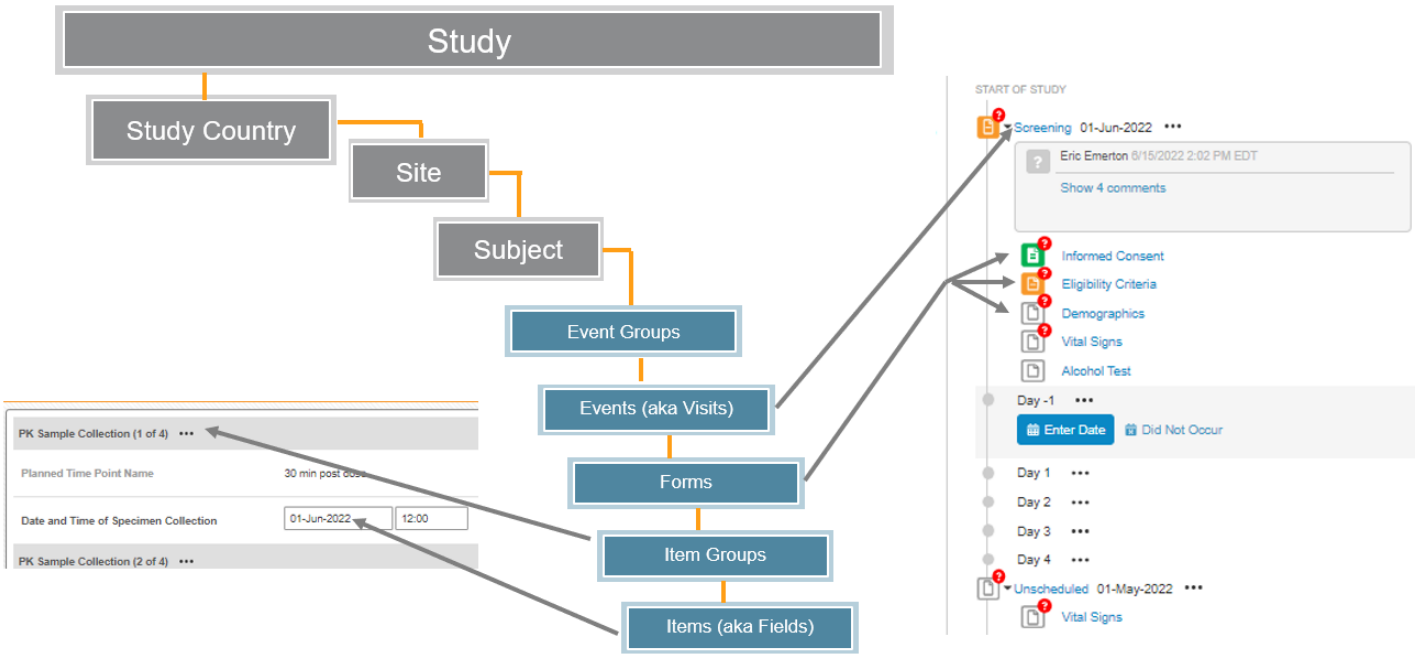
This document serves as an extended guide, supplemental to the API developer website for Veeva Vault CDMS API. Best practices, cause/effect examples are given here, whereas the online help is quicker reference with typically few/one example(s). Refer to the **Best Practices** chapter at the end of this document, and **TIP** entries are also given inline with the examples.

## 1.2 Reference URLs

| | |
|---|---|
| https://developer-cdms.veevavault.com/api/22.2/ | As of Veeva Vault release 22.2, this is the main CDMS API reference (subjects, events, forms, etc.)<br><br>**NOTE**: As new major Vault CDMS releases occur (April, August, December each year), the documentation link for newer API version is the same, but with new XX.Y ( year.release number) |
| https://developer.veevavault.com/ | General home page for Veeva Vault developers |
| https://developer.veevavault.com/docs/#rest-api | Quick start guide for general Veeva Vault API |
| https://developer.veevavault.com/api/22.2/ | Platform Vault API reference as of this release. General Vault concepts used by any vault application (logs, FTP, documents, etc.) are outlined |

## 1.3 CDMS Hierarchy

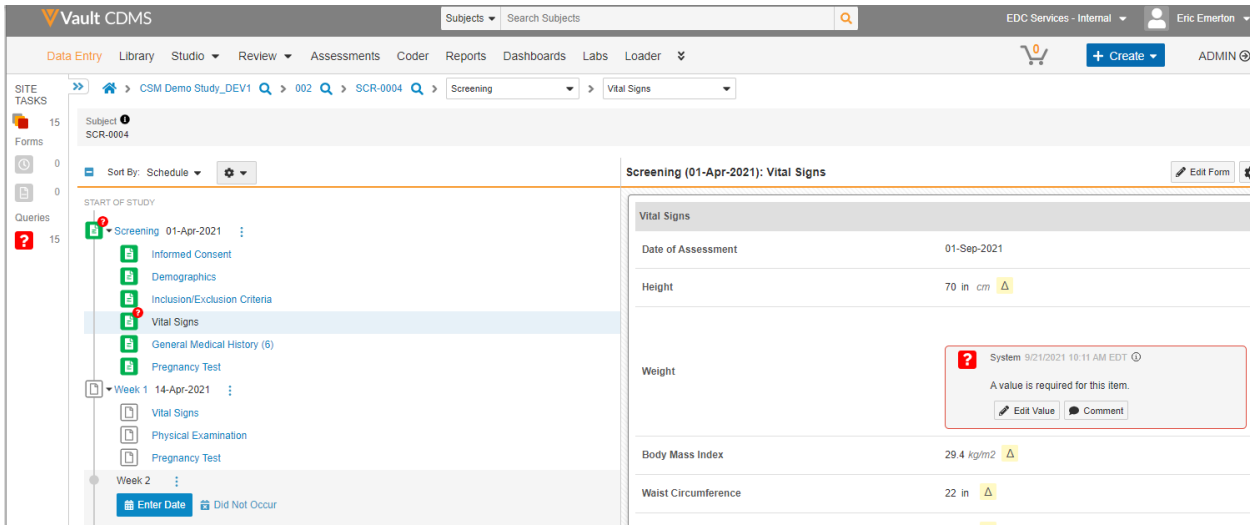
The CDMS data hierarchy / terminology is summarized by:




- Access within a study follows the **Study Country -> Site** hierarchy, i.e. which **sites** a user can see. A Data Manager might have access to all, a regional CRA some countries but not all, and site users just their specific site
- Once in a subject, the components of data follow largely an ODM CDISC model (Visits/Forms/Item Groups/Items), where 'Event' = 'Visit'. There is an additional layer for Event Groups to cover repeating cycles of events/visits as one unit.

Referring to the below UI example…

- Study = **CSM Demo Study_DEV1**
- Study Country = (not pictured, **United Status**, a relationship to the site)
- Site = **002**
- Subject = **SCR-0004**
- Event Group = (not pictured). A group of events (aka visits) are organized into an event group to better facilitate repeating cycles, unscheduled events.
- Event = **Screening**
- Form = **Vital Signs** (details, right side of screen)
- Item Group = **Vital Signs** (group of fields on a CRF, used for repeating when necessary)
- Items = **Date of Assessment** (date field), **Weight** (unit codelist field), etc.




- **WARNING**: The pictured elements in the EDC UI are **LABELS**, not the parameters necessary to interact through the API. The CDMS API operates with design **NAMES** for push of data in, and has a few (albeit) rare cases where labels are returned.

- Depending on the API endpoint these components are usually required to identify the subject, then the location in the casebook / CRFs of the subject. Some endpoints allow a 'by ID' update, the internal Vault ID, instead of a Study Country / Site / Subject / Event Group / Event… combination. (but not all)

- How does one get the proper parameters for these design elements? The team that built the study should relay this information vai **Study Design Specifications (SDS)** document generated from the Vault CDMS **Studio** module. At later API releases, there will be a design level API to get this information of names <-> labels, plus supporting properties at all design levels, and for each casebook definition.

- Continuing with the example pictured above, the **SDS (workbook) -> Schedule – Tree** tab of the SDS:

| A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|
| Caseboo | Event Group | Event Group Name | Event | Event Name | Form | Form Name | Restricte | Repea |
| 1 | | | | | | | | |
| 1 | Screening | eg_SCREENING | | | | | | No |
| 1 | Screening | eg_SCREENING | Screening | ev_SCREENING | | | | No |
| 1 | Screening | eg_SCREENING | Screening | ev_SCREENING | Informed Consent | IC | No | No |
| 1 | Screening | eg_SCREENING | Screening | ev_SCREENING | Demographics | DM | No | No |
| 1 | Screening | eg_SCREENING | Screening | ev_SCREENING | Inclusion/Exclusion Criteria | IE | No | No |
| 1 | Screening | eg_SCREENING | Screening | ev_SCREENING | Vital Signs | VS | No | No |
| 1 | Screening | eg_SCREENING | Screening | ev_SCREENING | General Medical History | MH | No | Yes |
| 1 | Screening | eg_SCREENING | Screening | ev_SCREENING | Pregnancy Test | LB_PREG | No | No |
| 1 | Treatment | eg_TREATMENT | | | | | | No |
| 1 | Treatment | eg_TREATMENT | Week 1 | ev_WEEK1 | | | | No |
| 1 | Treatment | eg_TREATMENT | Week 1 | ev_WEEK1 | Vital Signs | VS2 | No | No |
| 1 | Treatment | eg_TREATMENT | Week 1 | ev_WEEK1 | Physical Examination | PE | No | No |

…tells us the Event Group, Event, Form design names for the example given.

- Event Group = **eg_SCREENING**
- Event = **ev_SCREENING**
- Form = **VS**

- Item group design names, item design names are then found on the Form Definitions tab:

| F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Item Group | Item Group Label | IG Re | Header Visible | Visual | RM | IG | Item Name | External ID | Label | Data Type | Requi |
| | | | | | | | | VS | | | |
| VS | Vital Signs | No | Yes | Yes | | | | VS | | | |
| VS | Vital Signs | | | | | | VSDAT_2 | VSDAT | Date of Assessment | Date | Yes |
| VS | Vital Signs | | | | | | HEIGHT | HEIGHT | Height | Unit | Yes |
| VS | Vital Signs | | | | | | WEIGHT | PL_WEIGHT | Weight | Unit | Yes |
| VS | Vital Signs | | | | | | BMI | BMI | Body Mass Index | Number | Yes |
| VS | Vital Signs | | | | | | WSTCIR | WSTCIR | Waist Circumference | Unit | Yes |
| VS | Vital Signs | | | | | | VSPOS | VSPOS | Position | Codelist | Yes |
| VS | Vital Signs | | | | | | SYSBP | SYSBP | Systolic Blood Pressure | Number | Yes |
| VS | Vital Signs | | | | | | DIABP | DIABP | Diastolic Blood Pressure | Number | Yes |
| VS | Vital Signs | | | | | | HR | HR | Heart Rate | Number | Yes |
| VS | Vital Signs | | | | | | RESP | RESP | Respiratory Rate | Number | Yes |
| VS | Vital Signs | | | | | | TEMP | TEMP | Body Temperature | Unit | Yes |
| | | | | | | | | MH | | | |
| MH | General Medical History | No | Yes | Yes | | | | MH | | | |

…which indicates:

- Item Group = **VS**
- Items = **VSDAT_2, HEIGHT, WEIGHT, BMI…**

## 1.4 CDMS API Summary

- The Vault API is REST based. The vast majority of the API endpoints use URL parameters or JSON request bodies, return JSON response bodies (unless a file or log download)
- General vault API includes a JSON return structure like:

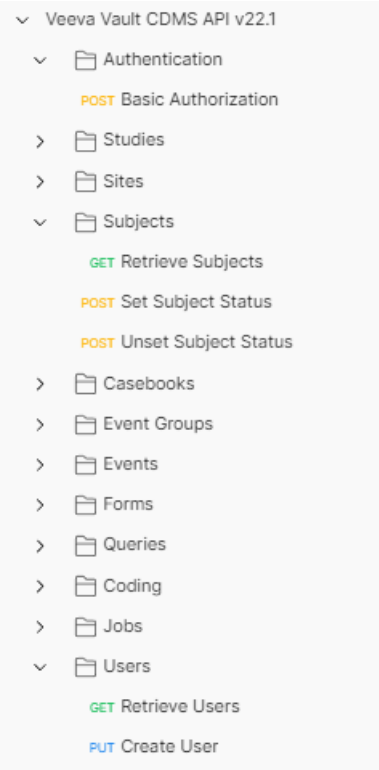
```
{
    "responseStatus": "SUCCESS",
    "responseDetails": {
        "pagesize": 1000,
        "pageoffset": 0,
        "size": 131,
        "total": 131
    },
    "data": [
```

.. with **responseStatus** returning the overall request status. Most Vault APIshave paging, with tota, limit, offset, and URLs to the 'next' / 'previous' page of data (of a query/request).

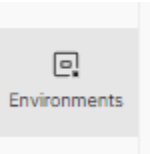
- CDMS API endpoint are very 1 to 1 with the actions / buttons seen in the UI. As example, if the intent is to add a subject, plus data to some forms in a screening visit, then APIs must be called for each of (potentially) - (a) Add subject (b) Set subject status (c) Set event date (first event for subject) (d) Set of data on form (with add item groups on the form if/when necessary (e ), Finally the submit of the form. The sequencing of these calls is vitally important. Currently, many actions are not rolled into fewer API calls, via parameters. (e.g., set data on a form, plus submit it, as one call). Consolidated API calls to accomplish multiple actions are in consideration for future releases all the time.
- CDMS specific errors – most examples are here: https://developer-cdms.veevavault.com/api/22.2/#errors-cdms

## 1.5 Postman Collection

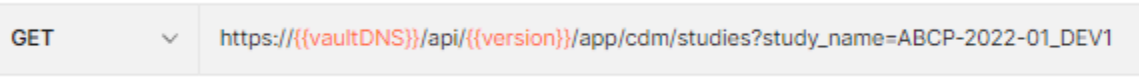
Any REST endpoint tool can be used for experimenting as part of API development. This guide will show examples using Postman. Each API release includes a Postman collection, downloadable from the developer API website. As example of a postman collection:




The use of a postman collection requires setting of an environment profil:

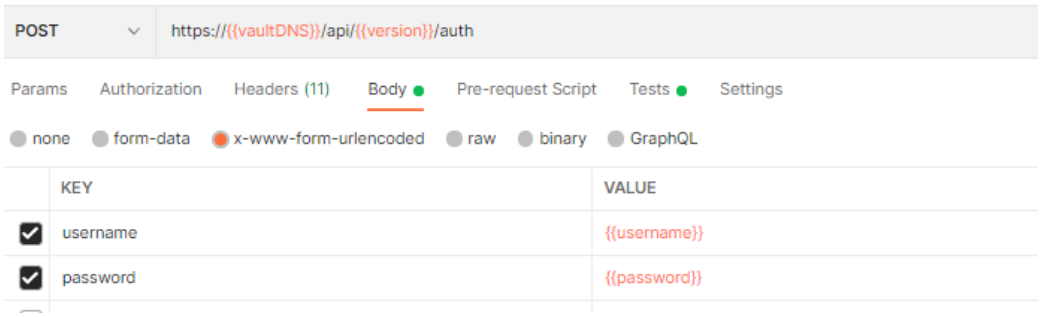



... to change between variables with in the collection, e.g. {{vaultDNS}} below (the URL for your vault), {{version}} (e.g. **v22.2** for this release):

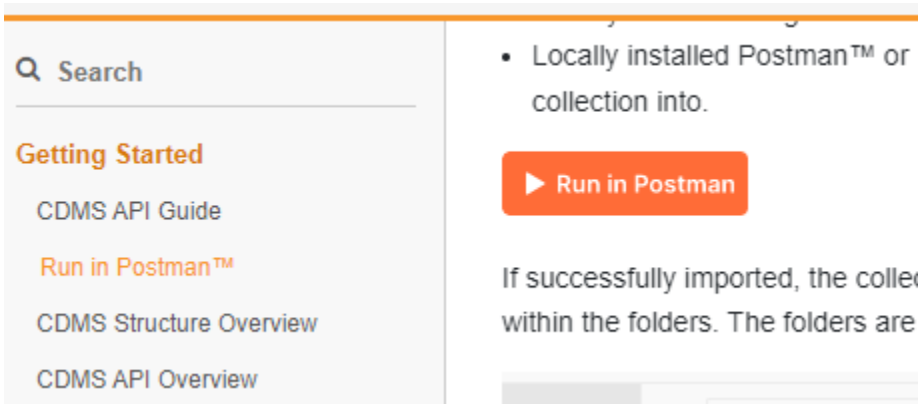



If your vault URL is, as example, **https://edc-services-internal.veevavault.com/...** , then the portion **edc-services-internal.veevavault.com** is what is used for the vaultDNS variable above.

Further, your username / password to the vault are environment variables used in the Authentication endpoint:




The postman collection can be imported from the help website, **Run in Postman** section:



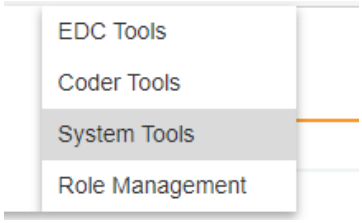

The **Run in Postman** button will lead you through getting the collection imported to your local Postman workspace, or Postman for Web workspace (if you have one).

## 1.6 Rate Limits / Caveats

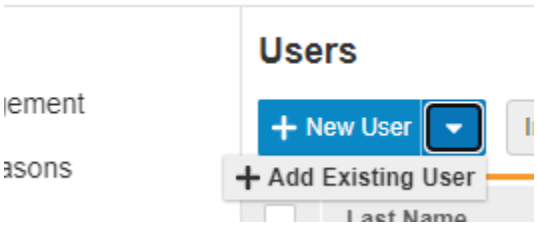
- **WARNING:** There are API limits vault wide, as well as by user, to protect the vault. These are general vault platform specific, not necessarily applicable because of Vault CDMSc. Earlier releases (< 21R1 ) have hard limits for each 5-minute window, plus a 24-hour daily limit for all users of the vault. For later versions, vault will slow down response times as the rate limits are hit. There is **also** a limit on the number of API logins in any single calendar minute (20). Do not re-authenticate with each API call. More information at https://developer-cdms.veevavault.com/api/22.2/#api-rate-limits
- **WARNING:** Many API returns will include a '**SUCCESS'** at the top level, that is the to/from the server was OK. But the body of the return must be parsed / inspected to see if there are any errors on the attempt. Do not assume top level success for pushing data to CDMS means it was fully successful! Most API actions into the system (update/insert data) are done in batch API requests, so each section should be inspected for its success/failure (not just the overall request status)
- **WARNING:** The CDMS API is built on top of the vault API, which allows raw VQL'ing to any object, fields that the user has rights to read. It is strongly **discouraged** to use raw object retrievals for reverse engineering of the data. Doing so will likely result in a job / code being tripped up at some later release when that data model changes. All attempts should be made to use the formal endpoints for data retrieval / updates into a study.

## 1.7 API Roles / Access

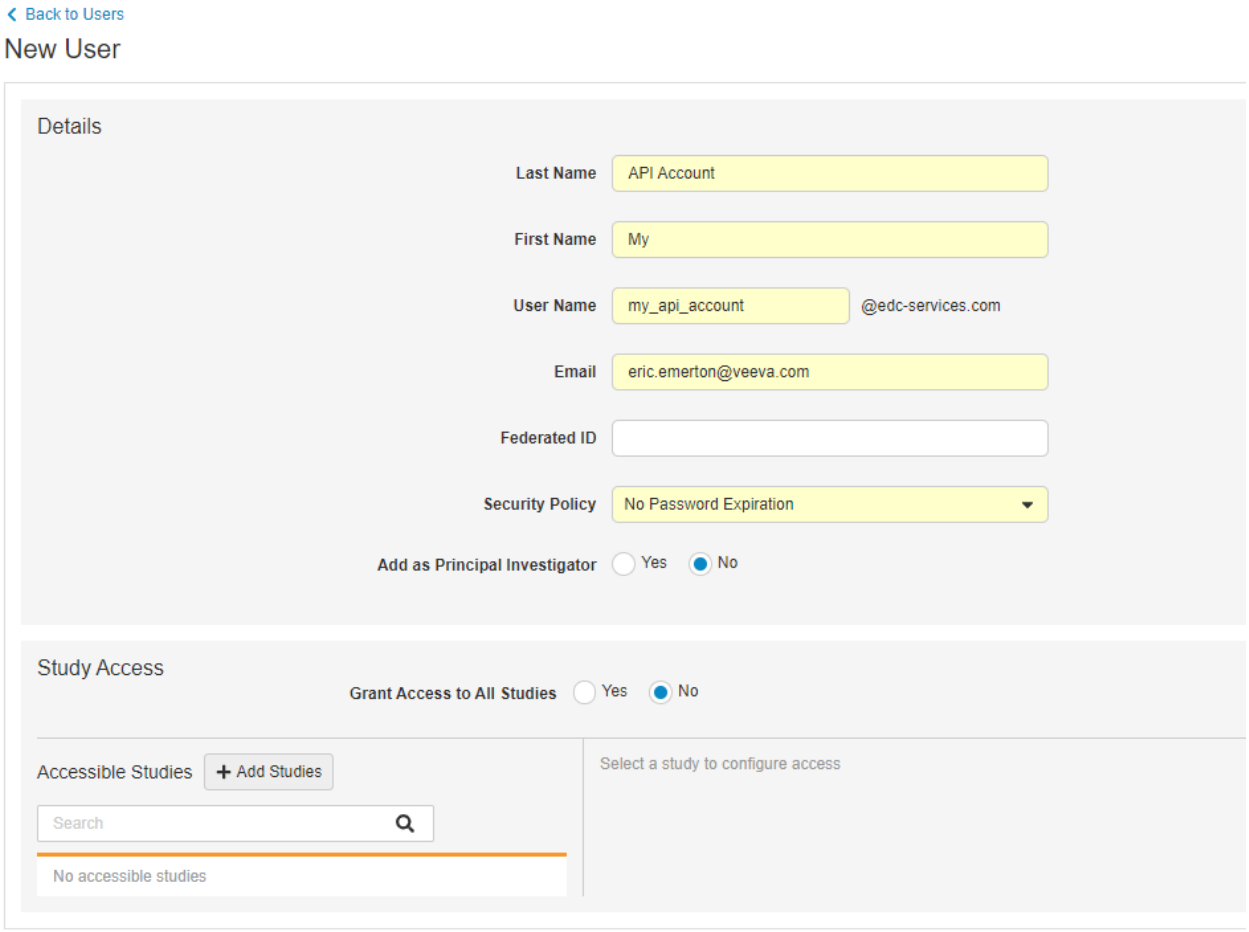
A user with Security Profile **Vault Owner** can use the API, the entire system in fact, in all ways. Other users are given access just like any other study user, with a specific role in that study. Users with rights to administer study users navigate to the **System Tools** area:



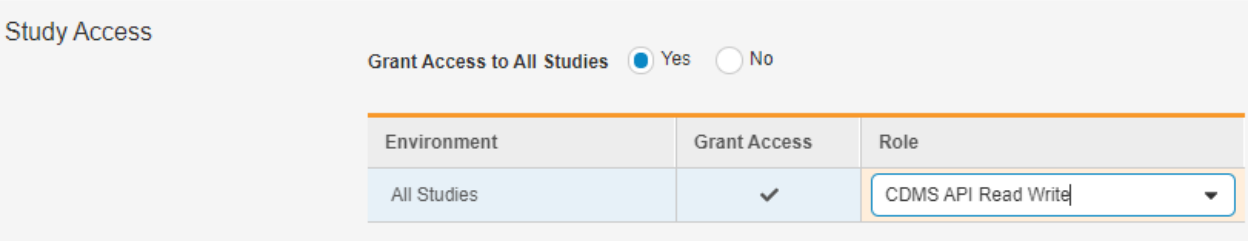

.. from which, one then creates a user (or add/apply an existing user on the Vault domain):



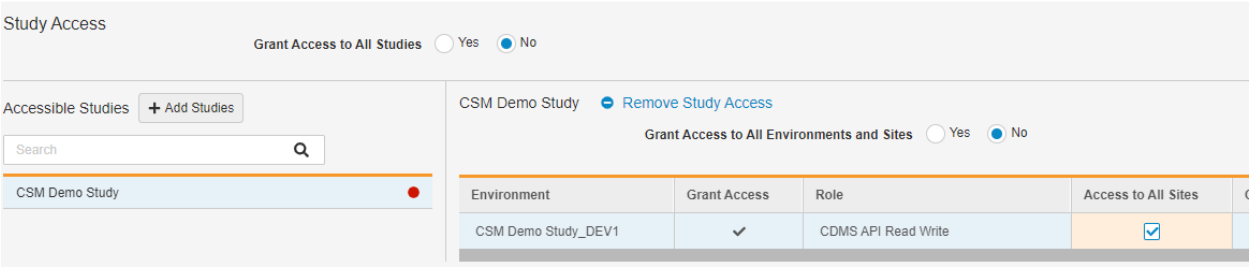

The access granted for a study on new user / apply existing user has multiple options for **Study Access** (2nd part of pictured):


Back to Users
New User
Details
Last Name
API Account
First Name
My
User Name
my_api_account
@edc-services.com
Email
eric.emerton@veeva.com
Federated ID
Security Policy
No Password Expiration
Add as Principal Investigator
Yes
No
Study Access
Grant Access to All Studies
Yes
No
Accessible Studies
+ Add Studies
Search
No accessible studies
Select a study to configure access

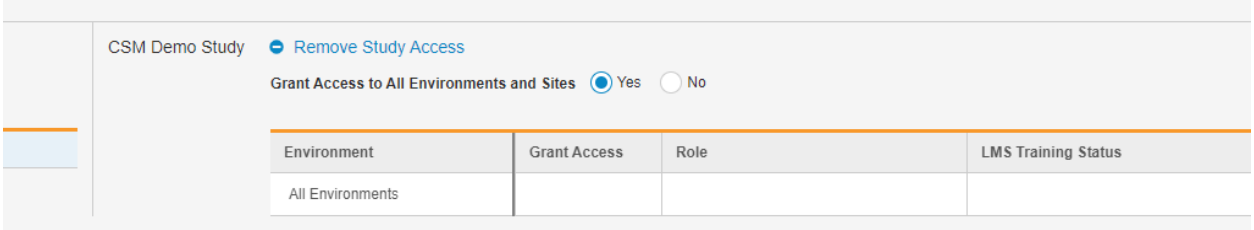
**Option A**: Access to all studies now, **AND** in the future: (less common)




**Option B**: Specific study / environment (note the **Access to All Sites = checked**, otherwise specific sites/countries):




**Option C**: Use **Grant Access to All Environments and Sites** = Yes. This is equivalent to **Access to All Sites**, but for all instances in the vault for the study. This is most applicable in a UAT vault where there are UAT1, UAT2, UAT3, etc. under one study umbrella. In a production or dev vault it has the same effect as Option B.



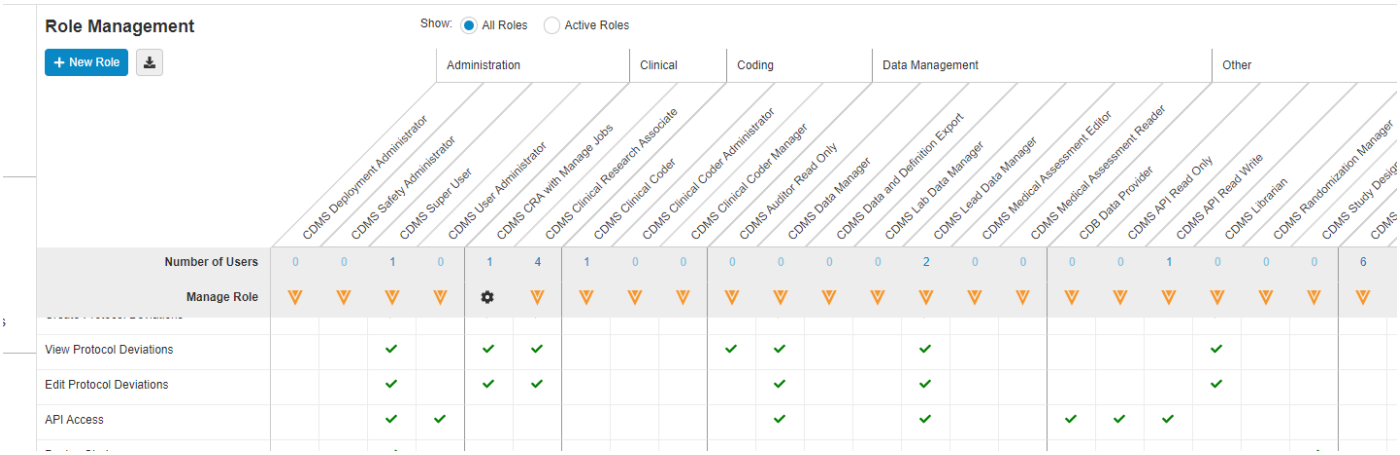

**TIP**: Consider only ever granting an API user the access / studies it needs, when needed for Production. In a UAT / Dev vault, access to all studies, or all instances of a study might be appropriate, unless the API user is for a partner / vendor company (e.g., IRT developing integrations)

**WARNING**: An API user is no different from any other study user. Typically, users must go through / pass a training before their study access is truly enabled. For an API / system user, ensure to check the Ignore LMS Status, Assume Trained attribute, otherwise the API user won't find the study (or studies) intended

| | LMS Training Status | Ignore LMS Status, Assume Trained |
|---|---|---|
| | | ☑ |

## 1.8 Vault Permission Sets / Roles

Vault CDMS resides on the Vault Platform and uses Vault Application Roles and Permission Sets for the basis of what a user can / cannot do. Should a custom role be setup (example: to also access a custom object / area), then the role needs to be setup with the API access permission, usually attainable with Vault CDMS's **Tools -> System Tools -> Role Management** grid, simply ensuring the API Access line checked for the new custom role:

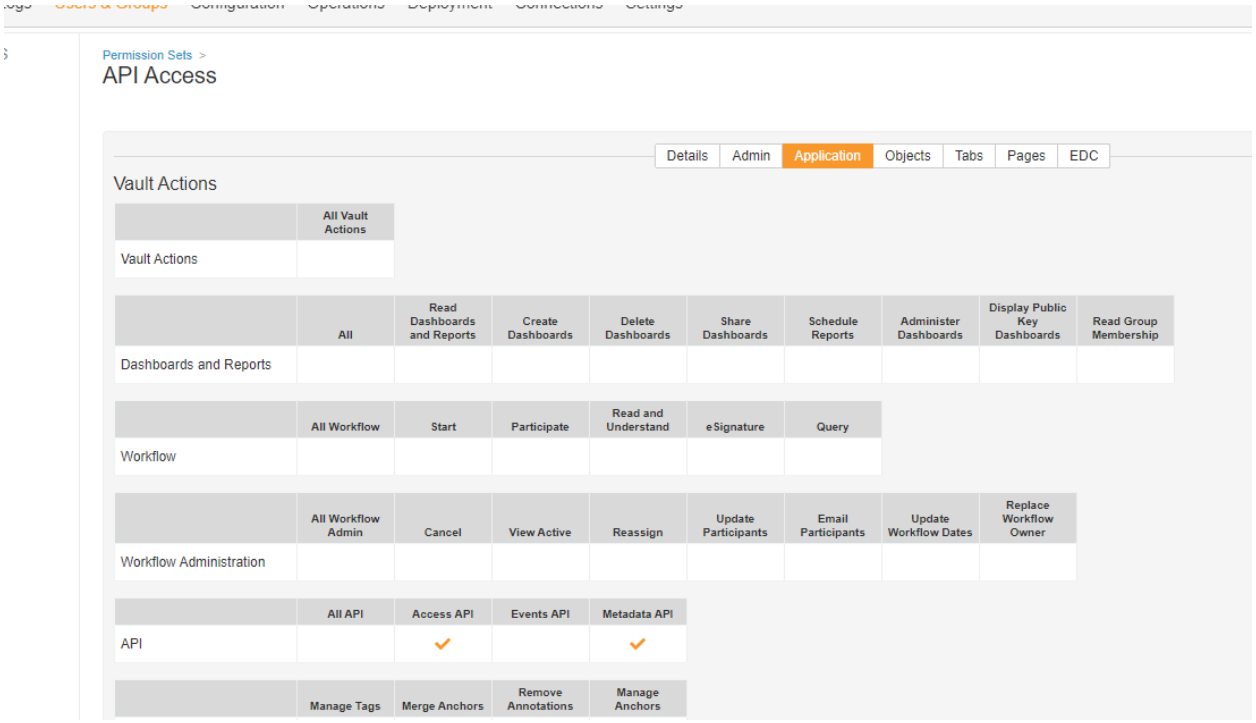



What follows in the rest of this section is the details on Vault level roles/permission sets. The key to a user being able to use the API on a study (or studies) is a permission set in their study role:

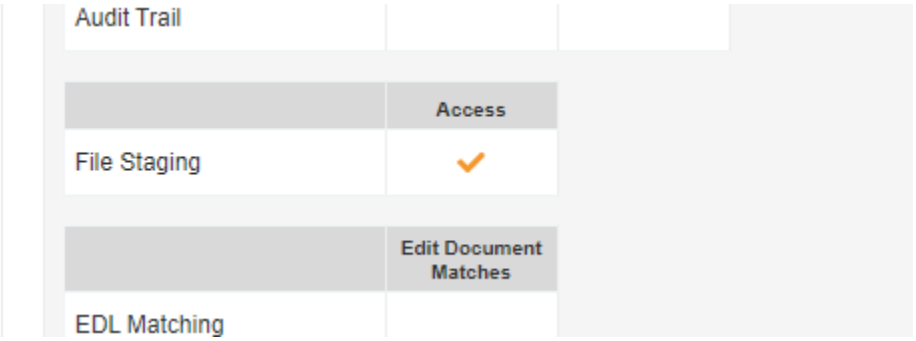
| Name | Status | Source | Description |
|---|---|---|---|
| API Access | Active | Standard | CDMS Permission Set for API access |

.. is necessary for using the API. This permission set is standard to CDMS and will either be included in a standard CDMS role, or custom CDMS role. The API user then is granted the role, with a study (or studies) as necessary

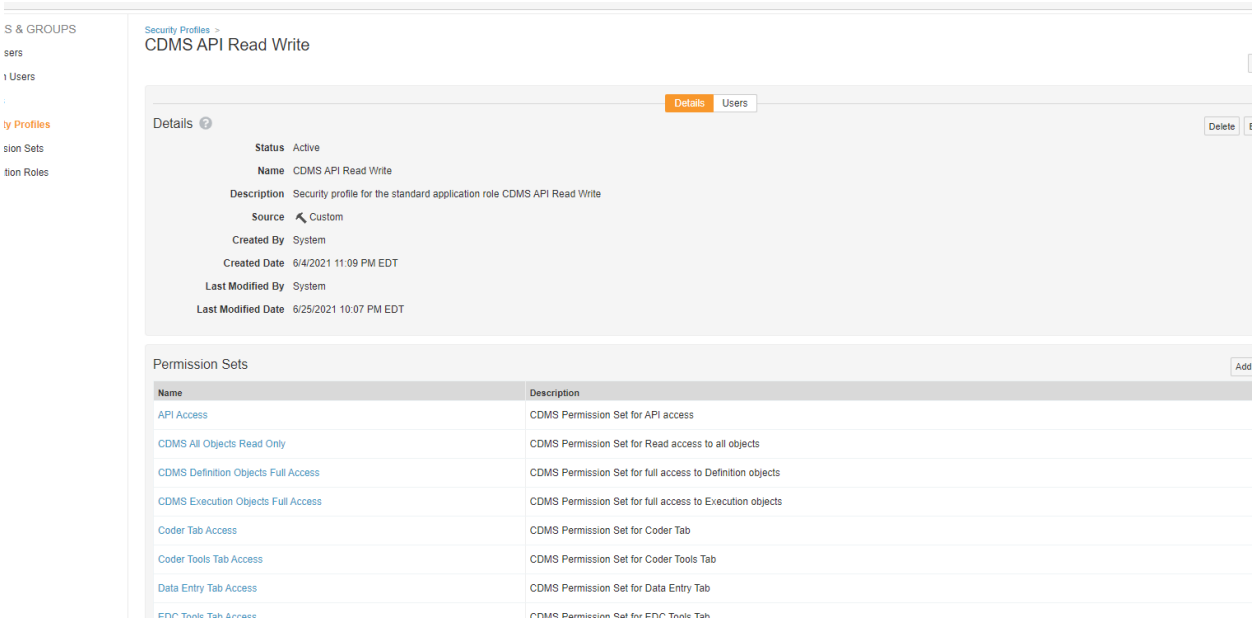
Should custom permission sets be setup in a vault, the key for API'ing is / are these settings:




As well as:

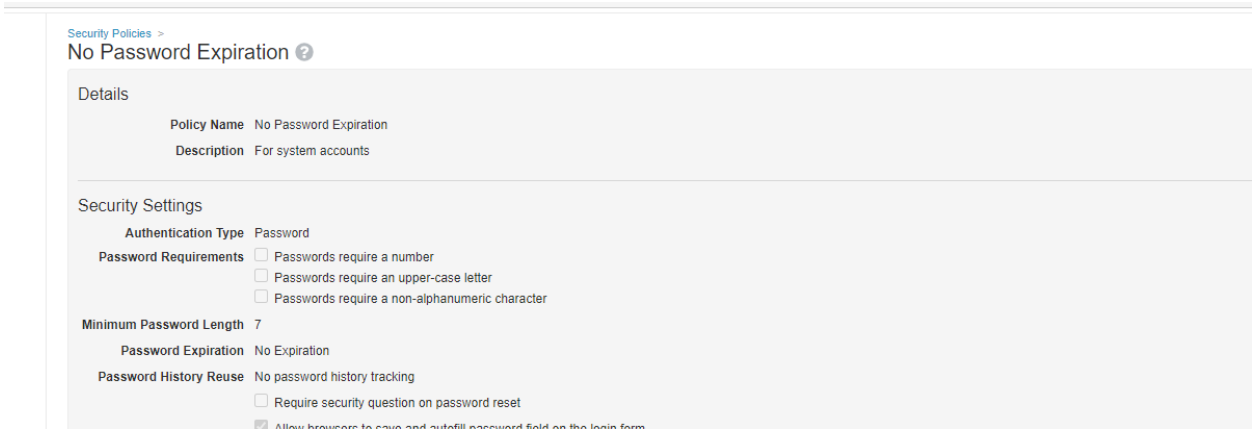



Together with base Vault settings for using its API (CDMS or any other Vault application), the other permission sets are CDMS specific. As example, a role **CDMS API Read Write** (custom), has the **API Access** permission set, plus relevant CDMS role




**TIP**: Consider using the built in API roles (**Standard**) for API users. Custom roles are usually made as clones of a standard role, with additive permissions to other areas, usually custom objects / tabs.

**TIP**: Consider using a Security Policy in the vault with no password expiration, so that API accounts do not suddenly run into password expiration during their use/runs. Below is a security policy named 'No Password Expiration', with the specific '**No Expiration**' property set.



## 2 Authentication / API Login

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#authentication |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/studies |
| Parameters | • username<br>• password<br><br>(these parameters are part of form-data post style) |
| Notes | • For login to the vault, get an API session.<br>• For Basic authorization (e.g. username / password passed)<br>• Other style login, using non-Basic authentication, see: https://developer.veevavault.com/api/22.2/#oauth-2-0-openid-connect<br>• Some older vault versions allowed a GET style with username / password. This has been disallowed in later versions (must use a **POST**). Even with older versions, it's strongly recommended **NOT** to use a GET style.<br>• An account to Veeva Vault is on a **domain**, which is one or more **vaults**. The above example, the user is on the **eke.com** domain with the Veeva Vault infrastructure and has access to two vaults on that domain.<br>• Above, the vaults the user has access to are returned, with parameters for id, name, URL of those vault locations. A **vault** is one to one with a URL – each of the two vaults above will have their own studies. (multiple studies reside in one vault)<br>• Note the **vaultId** return of the JSON at **top** level, this refers to the active vault on login. (vault ID = 31762).<br>• Accounts to vault(s) follow:<br>  ○ Non-production **domain**: UAT, Training, Dev vaults.  one account<br>  ○ Production domain where production study vaults reside.  2nd, different account |

## Examples

| Description / Request | Response |
|---|---|
| Successful Login | |
| …/api/v22.2/auth<br><br>(with post parameters for username / password, in this case user name: eric.emerton@eke.com) | {<br>"responseStatus": "SUCCESS",<br>"sessionId": "32735BDA5964E3F43C4F73F6109EFB1E2DCA8A2C1AAEB6B56C44822ADF36EE5C037A5DB751ED8AE924BC4A38EDE92CCB652F136A4AD59C7BB8490F170E514A44",<br>"userId": 92677,<br>"vaultIds": [<br>{<br>"id": 1004329,<br>"name": "EDC-QA-FAST-EKE",<br>"url": "https://eke-vvqa1-10.vaultdev.com/api"<br>},<br>{<br>"id": 1004330,<br>"name": "EDC-QA-SLOW-EKE",<br>"url": "https://eke-vvqa1-9.vaultdev.com/api"<br>}<br>],<br>"vaultId": 1004330<br>} |
| Bad User and/or password | |
| …/api/v22.2/auth | {<br>"responseStatus": "FAILURE",<br>"responseMessage": "Authentication failed for user [eric.emerton@eke.com]",<br>"errors": [<br>{<br>"type": "USERNAME_OR_PASSWORD_INCORRECT",<br>"message": "Authentication failed for user: eric.emerton@eke.com."<br>}<br>],<br>"errorType": "AUTHENTICATION_FAILED"<br>} |
| Session timed out | |
| | {<br>"responseStatus": "FAILURE",<br>"errors": [<br>{<br>"type": "INVALID_SESSION_ID", |

| Description / Request | Response |
|---|---|
| | "message": "Invalid or expired session ID."<br>}<br>]<br>} |

- The key portion of a successful authentication return is the **sessionId** parameter:

"responseStatus": "SUCCESS",
"sessionId":
"32735BDA5964E3F43C4F73F6109EFB1E2DCA8A2C1AAEB6B56C44822ADF36EE5C037A5DB751ED8AE924BC4A38EDE92CCB652F136A4AD59C7BB8490F170E514A44",
"userId": 92677,

.. as it must be passed in the header of all subsequent requests of the API session.

# 3 Studies

## 3.1 Retrieve Studies

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#retrieve-studies">https://developer-cdms.veevavault.com/api/22.2/#retrieve-studies</a></td></tr>
<tr><td>Style</td><td>GET</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/studies</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>limit</td><td>Optional</td><td>Pagination API - different limit from the default 1000 can be used</td></tr>
<tr><td>offset</td><td>Optional</td><td>Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Used to retrieve the list of studies in the vault</li>
<li>The casebook versions JSON array within each study entry enumerates all current casebook versions of the study. The casebook versions are assigned at add subject/casebook, per an active version at the subject’s site. Later, during amendments, a subject / site / entire study can transition from one version to the next.</li>
<li>ONLY the studies the user has rights to are returned</li>
<li>WARNING: If a user requires training (their role) in the vault, the study will not exist in API returns. Ensure API roles are configured to not be blocked by LMS restrictions</li>
<li>WARNING: The study_name value is what is used for all other requests into / out of CDMS. On rare occasions, the value can be out of sync with the visual study label seen on screen in the UI. For example, in a UAT vault the 6th UAT might be StudyName_UAT6 for study_name, but visually on screen StudyName_CandidateForLive.</li>
</ul>
</td></tr>
</table>

## Examples

| Description / Request (URL) | Response |
|---|---|
| Get All Studies | |
| …/api/v22.2/app/cdm/studies<br><br>(no parameters on the URL) | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 2,<br>"total": 2<br>},<br>"studies": [<br>{<br>"study_name": "Labrinone_DEV1",<br>"external_id": "Labrinone",<br>"study_phase": "Phase III",<br>"study_status": "Execution",<br>"casebook_versions": [<br>{<br>"study_name": "Labrinone_DEV1",<br>"casebook_version": 1,<br>"version_name": "Initial Version",<br>"external_id": "1",<br>"previous_version_name": null,<br>"description": null,<br>"change_reason": null,<br>"casebook_status": "in_progress__v",<br>"created_by": "Eric Emerton",<br>"created_date": "2018-02-12T22:32:21Z",<br>"last_modified_date": "2020-03-04T14:16:20Z"<br>}<br>]<br>},<br>{<br>"study_name": "ABCP-2022-01_DEV1",<br>"external_id": "ABCP-2022-01",<br>"study_phase": "Phase II",<br>"study_status": "Locked",<br>"casebook_versions": [<br>{<br>"study_name": "ABCP-2022-01_DEV1",<br>"casebook_version": 2,<br>"version_name": "2",<br>"external_id": "2",<br>"previous_version_name": null,<br>"description": null,<br>"change_reason": "For amendment to study", |

| Description / Request (URL) | Response |
| --- | --- |
| | "casebook_status": "published__v",<br>"created_by": "Eric Emerton",<br>"created_date": "2022-02-21T22:32:21Z",<br>"last_modified_date": "2022-02-21T14:16:20Z"<br>},<br>{<br>"study_name": "ABCP-2022-01_DEV1",<br>"casebook_version": 1,<br>"version_name": "Initial Version",<br>"external_id": "1",<br>"previous_version_name": null,<br>"description": null,<br>"change_reason": null,<br>"casebook_status": "published__v",<br>"created_by": "Eric Emerton",<br>"created_date": "2022-01-13T22:32:21Z",<br>"last_modified_date": "2022-01-13T14:16:20Z"<br>}<br>]<br>}<br>]<br>} |

# 4 Countries

## 4.1 Retrieve Countries

| | |
|---|---|
| Help Main | **WARNING: No CDMS formal endpoint for this yet, instead use of base Vault VQL is shown.** |
| Style | **GET** |
| Endpoint | <your Vault URL>/api/v22.2/query?q=SELECT id,abbreviation__v,abbreviation_2__v,name__v from country__v order by name__v |
| Parameters | • q (the query, as shown above) |
| Notes | • **Informational only**, do not necessarily actively use. The CDMS concept of 'Study Country' is a reference / copy of a base Vault country. This VQL gets that master country list. Study Country is more relevant, being countries 'in use' for a study. This query would only be needed if there is a need to detect what other countries are available to the vault, but not activated into a specific study.<br>• The 3-character and 2-character ISO codes are returned |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Get All Countries (Vault Level) | |
| …/api/v22.2/query?q=SELECT id,abbreviation__v,abbreviation_2__v,name__v from country__v order by name__v | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"pagesize": 1000,<br>"pageoffset": 0,<br>"size": 131,<br>"total": 131<br>},<br>"data": [<br>{<br>"id": "OOV000000000401",<br>"abbreviation__v": "AFG",<br>"abbreviation_2__v": "AF",<br>"name__v": "Afghanistan"<br>},<br>{<br>"id": "OOV000000000402",<br>"abbreviation__v": "ALB",<br>"abbreviation_2__v": "AL",<br>"name__v": "Albania"<br>}, |

| Description / Request (URL) | Response |
| --- | --- |
| | {<br>"id": "OOV000000000403",<br>"abbreviation__v": "DZA",<br>"abbreviation_2__v": "DZ",<br>"name__v": "Algeria"<br>},<br>{<br>"id": "OOV000000000404",<br>"abbreviation__v": "ARG",<br>"abbreviation_2__v": "AR",<br>"name__v": "Argentina"<br>}<br>:<br>:<br>(etc)<br>:<br>]<br>} |

## 4.2 Retrieve Study Countries

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#retrieve-study-countries |
| Style | **GET** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/studycountries |
| Parameters | |

| Name | Req/Opt | Notes |
|---|---|---|
| study_name | Required | Name of the study |
| limit | Optional | Pagination API - different limit from the default 1000 can be used |
| offset | Optional | Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd |

**Notes**

- Used to get the countries of a study. These are activated from the vault's main countries list (previous section) and are then available for selection on a new site.
- The 3-character country code is returned. The 2 character ISO code will be available in a later release of the API. The workaround is to get 2 character country code from the main Vault countries list - see previous section.
- It is not necessary to run this in most sessions as the get of sites / subjects will include study countries

### Examples

| Description / Request (URL) | Response |
|---|---|
| Get All Study Countries | |
| …/api/v22.2/app/cdm/studycountries?study_name=ABCP-2022-01_DEV1 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 3,<br>"total": 3<br>},<br>"study_countries": [<br>{<br>"study_country": "United States",<br>"country": "United States",<br>"country_abbreviation": "USA",<br>"created_by": "Eric Emerton",<br>"created_date": "2022-05-23T17:17:24Z",<br>"last_modified_date": "2022-05-23T17:17:24Z"<br>}, |

| Description / Request (URL) | Response |
| --- | --- |
| | {<br>"study_country": "Austria",<br>"country": "Austria",<br>"country_abbreviation": "AUT",<br>"created_by": "Eric Emerton",<br>"created_date": "2022-05-23T17:22:45Z",<br>"last_modified_date": "2022-05-23T17:22:45Z"<br>},<br>{<br>"study_country": "Canada",<br>"country": "Canada",<br>"country_abbreviation": "CAN",<br>"created_by": "Eric Emerton",<br>"created_date": "2022-05-23T17:22:53Z",<br>"last_modified_date": "2022-05-23T17:22:53Z"<br>}<br>]<br>} |

# 5 Sites

## 5.1 Retrieve Sites

| | |
|---|---|
| Reference | https://developer-cdms.veevavault.com/api/22.2/#retrieve-sites |
| Style | **GET** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/sites |
| Parameters (URL) | **Name** / **Req/Opt** / **Notes**<br>study_name / Required / Name of the study<br>study_country / Optional / Name of the study country, i.e. just the sites of that country<br>limit / Optional / Pagination API - different limit from the default 1000 can be used<br>offset / Optional / Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd |
| Notes | • Used to get the sites of a study.<br>• Optionally, get the sites only of one country<br>• Casebook version (i.e. what new subjects added to the site will receive) is returned with other attributes |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Get All Sites of a Study | |
| …/api/v22.2/app/cdm/sites?study_name=ABCP-2022-01_DEV1 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 3,<br>"total": 3<br>},<br>"sites": [<br>{<br>"site": "101",<br>"site_name": "Cary Hospital",<br>"site_status": "active__v",<br>"site_closeout_status": "",<br>"study_country": "United States",<br>"principal_investigator": "Linda Johnson", |

| Description / Request (URL) | Response |
|---|---|
| | "casebook_version": 1,<br>"timezone": "(GMT-05:00) Eastern Standard Time (America/New_York)"<br>},<br>{<br>"site": "102",<br>"site_name": "Raleigh Hospital",<br>"site_status": "active__v",<br>"site_closeout_status": "",<br>"study_country": "United States",<br>"principal_investigator": "Mary Peters",<br>"casebook_version": 1,<br>"timezone": "(GMT-05:00) Eastern Standard Time (America/New_York)"<br>},<br>{<br>"site": "201",<br>"site_name": "Montreal General",<br>"site_status": "active__v",<br>"site_closeout_status": "",<br>"study_country": "Canada",<br>"principal_investigator": "Jim Jones",<br>"casebook_version": 1,<br>"timezone": "(GMT-05:00) Eastern Standard Time (America/New_York)"<br>}<br>]<br>} |
| Get All Sites of a Country | |
| …/api/v22.2/app/cdm/sites?study_name=ABCP-2022-01_DEV1&study_country=Canada | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 1,<br>"total": 1<br>},<br>"sites": [<br>{<br>"site": "201",<br>"site_name": "Montreal General",<br>"site_status": "active__v",<br>"site_closeout_status": "",<br>"study_country": "Canada",<br>"principal_investigator": "Jim Jones",<br>"casebook_version": 1,<br>"timezone": "(GMT-05:00) Eastern Standard Time |

| Description / Request (URL) | Response |
| --- | --- |
| | (America/New_York)"<br>}<br>]<br>} |

# 6 Subjects / Casebooks

## 6.1 Retrieve Subjects

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#retrieve-subjects">https://developer-cdms.veevavault.com/api/22.2/#retrieve-subjects</a></td></tr>
<tr><td>Style</td><td>GET</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/subjects</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>Optional</td><td>Name of the study country, i.e. just the sites of that country</td></tr>
<tr><td>site</td><td>Optional</td><td>Name/number of the site, i.e. just subjects of that site</td></tr>
<tr><td>subject</td><td>Optional</td><td>Name/Number of the specific subject to filter to.</td></tr>
<tr><td>last_modified_date</td><td>Optional</td><td>Filter to subject’s modified since a certain date. Format: yyyy-MM-ddTHH:mm:ssZ</td></tr>
<tr><td>id</td><td>Optional</td><td>The specific vault ID of the subject, or IDs of several (comma separated)<br><br>NOTE: One can use the ID filter, but with no other filters (site, study_country, last_modified_date).</td></tr>
<tr><td>limit</td><td>Optional</td><td>Pagination API - different limit from the default 1000 can be used</td></tr>
<tr><td>offset</td><td>Optional</td><td>Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Used to get the subjects of a study</li>
<li>The response returned includes additional dates for milestones for the subject. These only appear when there is a date value, omitted from response if not set. Milestone dates are set by either study design configuration, or specific set of values by the API (Set Subject Status, later section). The Help Main link above details the names of those potential return values.</li>
<li>See the table below for filtering styles.</li>
</ul>
</td></tr>
</table>

## Subject Filtering Styles / Examples

| Style | Example | Result / Notes |
|---|---|---|
| Study Context | `.../subjects?study_name={..}` | All subjects |
| Study Context | `.../subjects?study_name={..}&study_country=Canada&site=101` | Site 101 subjects |
| Study Context | `.../subjects?study_name={..}&site=101` | Site 101 subjects (with/without country, same) |
| Study Context | `../cdm/subjects?study_name={..}&site=101,102` | Site 101 and Site 102 subjects |
| Study Context | `../cdm/subjects?study_name={..}&site=101,102,501` | Assuming 501 does not exist, Site 101 / 102, same |
| Study Context | `../cdm/subjects?study_name={..}&site=501,502` | **ERROR** - If neither exists, error return (not empty array), that is, no valid sites |
| Study Context | `.../subjects?study_name={..}&study_country=Canada&site=101,102` | **ERROR** - Multi site search is not allowed when including a study country, even if both reside at that country |
| Study Context | `../cdm/subjects?study_name={..}&last_modified_date=2022-07-01T12:00:00Z` | Subjects modified since July 1st, 12:00 UTC |
| By ID | `../cdm/subjects?study_name={..}&id=OPP00000001V002` | Just the subject with that matching ID, assuming valid Vault ID |
| By ID | `../cdm/subjects?study_name={..}&id=OPP00000001V002,OPP00000001V003` | The two subjects, assuming both are valid Vault IDs |
| By ID | `../cdm/subjects?study_name={..}&id=OPP00000001V077,OPP00000001V088` | Assuming neither exists, the return is empty JSON array (not an error), e.g.<br><br>:<br>subjects [ ]<br>:<br>: |

## Examples

| Description / Request (URL) | Response |
|---|---|
| Get All Subjects of a Study | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCP-2022-01_DEV1 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 12,<br>"total": 12<br>},<br>"subjects": [<br>{<br>"id": "OPP00000001P001",<br>"study_name": "ABCP-2022-01_DEV1",<br>"study_country": "United States",<br>"site": "101",<br>"site_name": "Cary General",<br>"subject": "SCR-0009",<br>"status": "pre_screen__v",<br>"casebook_version": 1,<br>"created_date": "2022-06-20T20:28:37Z",<br>"created_by": "Eric Emerton",<br>"last_modified_date": "2022-06-20T20:28:39Z"<br>},<br>{<br>"id": "OPP00000001O001",<br>"study_name": "ABCP-2022-01_DEV1",<br>"study_country": "United States",<br>"site": "101",<br>"site_name": "Cary General",<br>"subject": "SCR-0008",<br>"status": "pre_screen__v",<br>"casebook_version": 1,<br>"created_date": "2022-06-20T20:14:34Z",<br>"created_by": "Eric Emerton",<br>"last_modified_date": "2022-06-20T20:14:36Z"<br>},<br>:<br>etc..<br>:<br>]<br>} |

| Description / Request (URL) | Response |
|---|---|
| Get All Subjects of a Country (context filter) | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCP-2022-01_DEV1&study_country=Canada | (as above.. but to filter of Canada sites only) |
| Get Subjects of a Site (context filter), where milestone dates are set/present (all shown) | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCP-2022-01_DEV1&study_country=United States&site=101 | : : }, "subjects": [ { "id": "OPP00000001H001", "study_name": "ABCP-2022-01_DEV1", "site": "101", "study_country": "United States", "site_name": "Cary General", "subject": "101-001", "status": "complete__v", "casebook_version": 1, "created_date": "2022-06-22T20:53:42Z", "created_by": "Eric Emerton", "initial_consent_date": "2022-06-01", "screened_date": "2022-06-01", "screen_failed_date": "2022-06-02", "enrolled_date": "2022-06-04", "randomized_date": "2022-06-04", "start_treatment_date": "2022-06-21", "end_treatment_date": "2022-06-28", "withdrawn_date": "2022-07-01", "started_follow_up_date": "2022-07-02", "lost_to_follow_up_date": "2022-08-01", "end_study_date": "2022-08-15", "last_modified_date": "2022-09-23T17:18:15Z" }, : : |
| Get Subjects of a Site (context filter, with country included) | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCP-2022-01_DEV1&study_country=United States&site=101 | (as above.. but to filter of site 101 only) |

| Description / Request (URL) | Response |
|---|---|
| | |
| Get Subjects of a Site (context filter, no country included, works either way) | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCP-2022-01_DEV1&site=101 | (as above.. but to filter of site 101 only) |
| Get Subjects of Several Sites (context filter, no country needed) | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCP-2022-01_DEV1site=101,102 | (as above.. but to filter of site 101 and site 102 only) |
| Get Subjects of Several Sites (context filter, no country needed) | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCP-2022-01_DEV1site=101,102 | (as above.. but to filter of site 101 and site 102 only) |
| Get Specific Subject (ID style filter) | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCPharma_DEV1&id=OPP000000000501 | (as above.. but the one subject returned only) |
| Get Two Specific Subjects (ID style filter) | |
| …/api/v22.2/app/cdm/subjects?study_name=ABCPharma_DEV1&id=OPP000000000501,OPP000000000502 | (as above.. but the two subjects returned only) |
| Get Subjects of Several Sites, WITH country included (**ERROR returned**, multi site search is only with **just** site parameter) | |
| …/app/cdm/subjects?study_name=ABCP-2022-01_DEV1&study_country=United States&site=101,102 | {<br>"responseStatus": "FAILURE",<br>"errorMessage": "Search of multiple sites is not allowed when a country is provided"<br>} |

| Description / Request (URL) | Response |
|---|---|
| | |

## 6.2 Create Subjects (Casebooks)

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#create-subjects |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/casebooks |
| Parameters | |

| Name | Array/Section | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| study_country | subjects | Required | Name of the study country (for the new subject/casebook) |
| site | subjects | Required | Name/number of the site (for the new subject/casebook) |
| subject | subjects | Optional | The specific subject number to set. When setting, this must be unique study wide. Auto generated numbers - depending on study configuration - can repeat what are pre screening numbers at different sites, but the formal subject number / value must be unique study wide. |

Notes

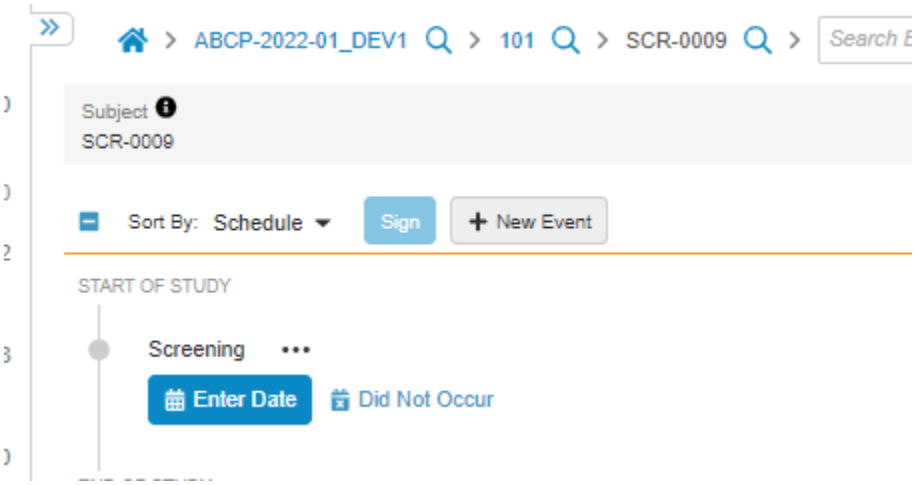
- Used to add a subject/casebook into the study
- For the purposes of add, subject = casebook (synonyms in this context)
- The **subject** parameter is optional. When omitted, the next auto screening number is assigned. Those follow the pattern SCR-0001, SCR-0002, SCR-0003.. within each site, although that is configurable in the target study.
- Multiple subjects/casebooks can be added with one API call. (entry in JSON array per attempt)

### Examples

| Description / Request (URL) | Response |
|---|---|
| Add Subject - setting a specific subject number | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101002"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"subjects": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101002",<br>}<br>]<br>} |
| Adding Many Subjects | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101002"<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101003"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"subjects": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101002",<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101003",<br>}<br>]<br>} |
| Add Subject using 'Next' Style Pre Screen Numbering | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"study_country": "United States",<br>"site": "101"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"subjects": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0008",<br>}<br>]<br>} |

| Description / Request (URL) | Response |
|---|---|
| Add Many Subjects - Some work, some fail | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"study_country": "United States",<br>"site": "101"<br>},<br>{<br>"study_country": "Germany",<br>"site": "201"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"subjects": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0012",<br>},<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "[Study Country] with name [Germany] cannot be found",<br>"study_country": "Germany",<br>"site": "201",<br>}<br>]<br>} |

**IMPORTANT:** The end result of the add subject/casebook is the same as the UI Create Casebook button/action. The subject is left with only the first event leading, with no date yet. Further actions are performed with additional endpoint calls. (e.g., set event date, then setting of form data, set of subject status)



## 6.3 Set Subject Status

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#set-subject-status">https://developer-cdms.veevavault.com/api/22.2/#set-subject-status</a></td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/subjects/actions/setstatus</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array/Section</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>subjects</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>subjects</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>subjects</td><td>Required</td><td>Subject to set status</td></tr>
<tr><td>subject_status</td><td>subjects</td><td>Required</td><td>The status to set, using the API name</td></tr>
<tr><td>date</td><td>subjects</td><td>Required</td><td>Date to set for milestone value of that status. Format: yy-MM-dd</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Used to set the status of a subject</li>
<li>Often done as a 2nd step, right after adding of a new subject (e.g., advance past Pre-Screening default status to Screening, Enrolled, or other)</li>
<li>That subject_status value must be the API name, not the label (e.g. enrolled__v, including the Veeva suffix). See the table below for out of the box statuses, although there can be additional statuses in the vault (custom)</li>
<li>The set of a milestone date, i.e., the date the subject took on that status is required. ‘Now’ is not specifically used, as this might/could be a date in the past being set. This is the equivalent of a studio design rule, which when some condition is true from submit of form data, a date is set + status changed for the subject.</li>
<li>Refer to the table below / next page for order of precedence of subject statuses, when setting or unset.</li>
<li>A future release of the API will decouple these concepts, so the move of status and/or the set of milestone dates can be done separately, i.e. not using milestone dates, or correcting status dates without having to unset / redo status calls.</li>
<li>WARNING: Currently, the set of multiple subjects, or multiple statuses for one subject (in one API call) is not allowed. The format of request is still with the format of a subjects array, for future state of this endpoint.</li>
</ul>
</td></tr>
</table>

### Status Order / Precedence

Subject statuses follow an order of precedence, with general rule of thumb that to rollback a status, one must call the **Unset Subject Status** (next section) in reverse order of the precedence:

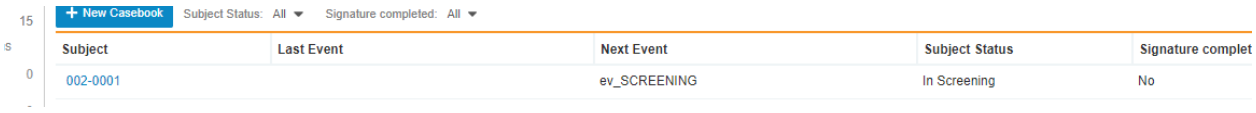
| Order | Subject Status | API Value to Set | Field Return in Retrieve Subjects |
|---|---|---|---|
| 1 | Pre Screen | pre_screen__v | *N/A, not a milestone date* |
| 2 | Consented | consented__v | initial_consent_date |
| 3 | In Screening | in_screening__v | screened_date |
| 4 | Screen Failure | screen_failure__v | screen_failed_date |
| 5 | Enrolled | enrolled__v | enrolled_date |
| 6 | Randomized | randomized__v | randomized_date |
| 7 | Started Treatment | started_treatment__v | started_treatment_date |
| 8 | End Of Treatment | end_of_treatment__v | end_of_treatment_date |
| 9 | Withdrawn | withdrawn__v | withdrawn_date |
| 10 | Started Follow Up | started_follow_up__v | started_follow_up_date |
| 11 | Lost to Follow Up | lost_to_follow_up__v | lost_to_follow_up_date |
| 12 | Complete | complete__v | end_of_study_date |

## Examples

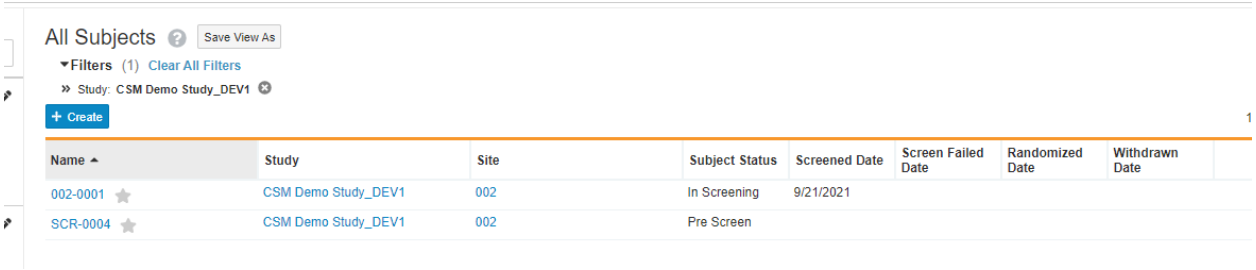
| Description / Request (URL) | Response |
|---|---|
| **Set Enrolled Status to Subject** | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101001",<br>"subject_status": "enrolled__v",<br>"date": "2022-01-15"<br>}<br>]<br>} | {<br>"responseStatus": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101001",<br>"subject_status": "enrolled__v"<br>}<br>]<br>} |
| **Attempt that fails** | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"study_country": "United Statess",<br>"site": "101",<br>"subject": "101002",<br>"subject_status": "enrolled__v",<br>"date": "2022-01-16"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"subjects": [<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "[Study Country] with name [United Statess] not found",<br>"site": "101",<br>"study_country": "United Statess",<br>"subject": "101002",<br>"subject_status": "enrolled__v"<br>}<br>]<br>} |
| **Failing Attempt (Batch)** | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101002",<br>"subject_status": "in_screening__v",<br>"date": "2022-01-15"<br>},<br>{<br>"study_country": "United States", | {<br>"responseStatus": "FAILURE",<br>"errors": [<br>{<br>"type": "SDK_ERROR",<br>"subtype": "INVALID_DATA",<br>"message": "Status can only be set for one subject at a time"<br>}<br>]<br>} |

| Description / Request (URL) | Response |
| --- | --- |
| <pre>"site": "101",<br>"subject": "101003",<br>"subject_status": "enrolled__v",<br>"date": "2022-01-16"<br>}<br>]<br>}</pre> | |

Once set, example of status in the EDC UI:




Milestone dates set are not present in the main entry UI, but available in reports / exports:



## 6.4 Unset Subject Status

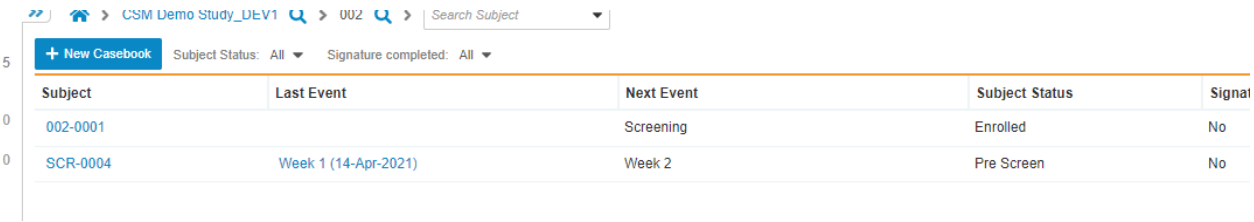
<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#unset-subject-status">https://developer-cdms.veevavault.com/api/22.2/#unset-subject-status</a></td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/subjects/actions/unsetstatus</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array/Section</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>subjects</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>subjects</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>subjects</td><td>Required</td><td>Subject to unset status</td></tr>
<tr><td>subject_status</td><td>subjects</td><td>Required</td><td>The status to unset, using the API name</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Used to unset the status of a subject</li>
<li>See the table of statuses in the previous section for precedence / order. There can be also be additional statuses in the vault (custom)</li>
<li>The unset of the subject status will also rollback /remove the milestone date previously set. There is order of precedence so if a subject has moved In Screening -> Enrolled -> Randomized, and an unset of Enrolled takes place, BOTH the enrolled and randomized dates are removed. See the previous section for an order of precedence of subject statuses</li>
<li>WARNING: Currently, the unset of multiple subjects, or multiple for one subject (in one API call) is not allowed. The format of request is still with format of a subjects array, for future state</li>
</ul>
</td></tr>
</table>

## Examples

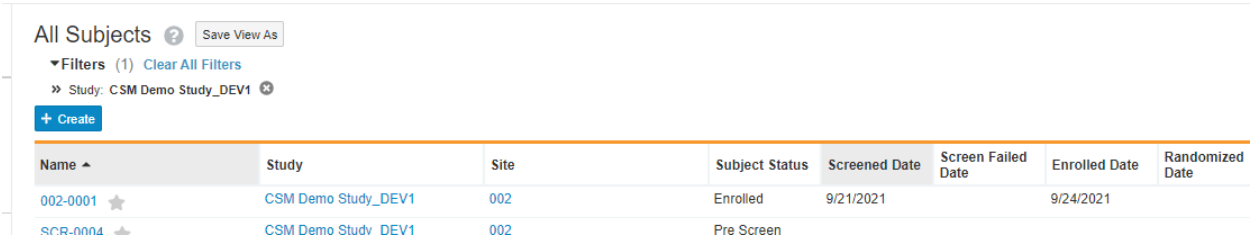
| Description / Request (URL) | Response |
|---|---|
| Unset Enrolled Status | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101001",<br>"subject_status": "enrolled__v"<br>}<br>]<br>} | {<br>"responseStatus": "ABCP-2022-01_DEV1",<br>"subjects": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101001",<br>"subject_status": "enrolled__v"<br>}<br>]<br>} |

## Flow Based Example

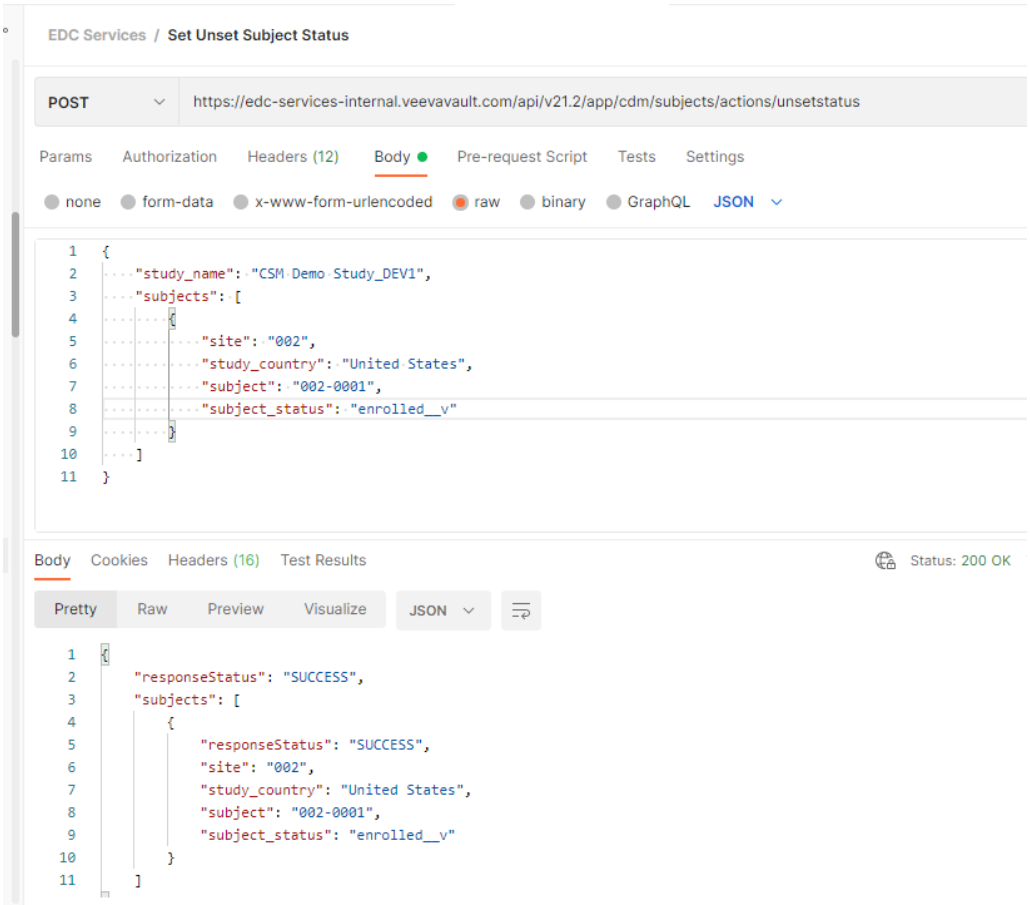
For a UI based cause effect, starting with a subject at Enrolled:



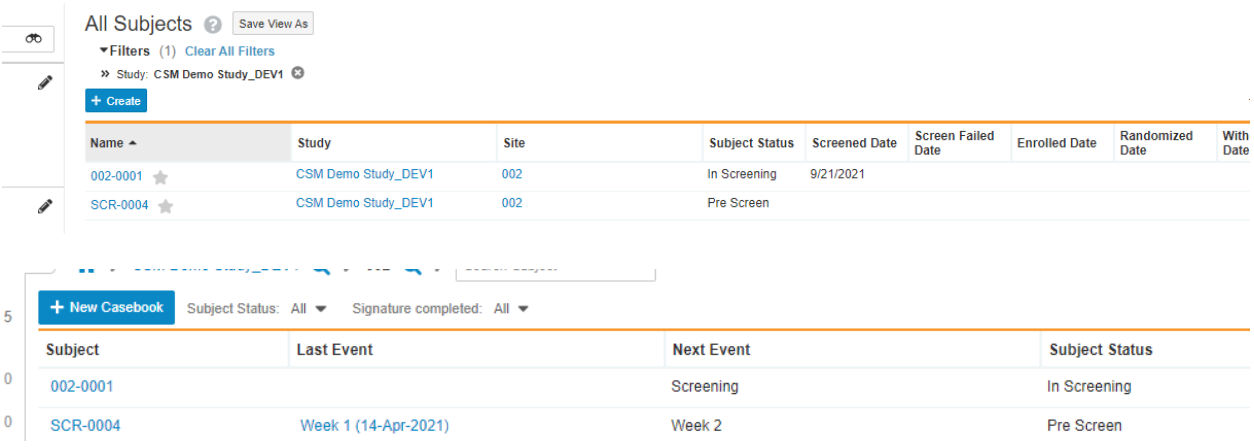

.. and with milestone of Sept 24th for Enrolled:



**Example / Return**: (success, removes the milestone date as part of status rollback):




**Result In EDC UI:** (status listing, enrolled date is removed, then main subject list, back to **In Screening** status)



# 7 Event Groups

## 7.1 Create Event Groups

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#create-event-groups |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/eventgroups |
| Parameters | |

| Name | Array/Section | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| study_country | eventgroups | Required | Name of the study country (of the subject's site) |
| site | eventgroups | Required | Name/number of the site |
| subject | eventgroups | Required | Subject to act on |
| eventgroup_name | eventgroups | Required | Design **name** in the study of the event group to add/upsert |
| date | eventgroups | Optional* | Only applicable to **Unscheduled** type event groups. Format: yyyy-MM-dd |

Notes

- Used to add an Event Group
- This action will add the **'next'** event group, in sequence if the event group repeats. Example: Unscheduled event group currently has 2 existing, the action would add one with event group **sequence** = 3.
- Use the upsert (PUT), next section, for conditional add at a specific sequence. Example: 3 unscheduled event groups exist, and attempt to add sequence = 3 is essentially a skip. (i.e. 'add if not there, otherwise do nothing).
- For the **Unscheduled** type event group, the API call **must** include an event date (format: **yyyy-MM-dd**). Il other event group types, the date is not set in this API call, instead the Set Event Date (see the **Set Event Date** section)

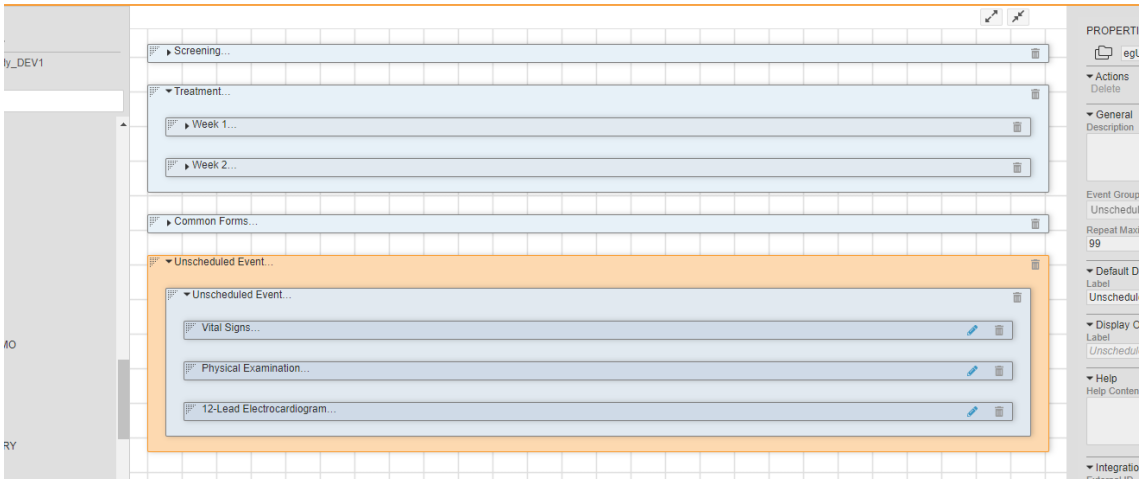
### Examples

| Description / Request (URL) | Response |
|---|---|
| Add New Event Groups (multiple subjects in one call) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"eventgroups": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001", | {<br>"responseStatus": "SUCCESS",<br>"eventgroups": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101", |

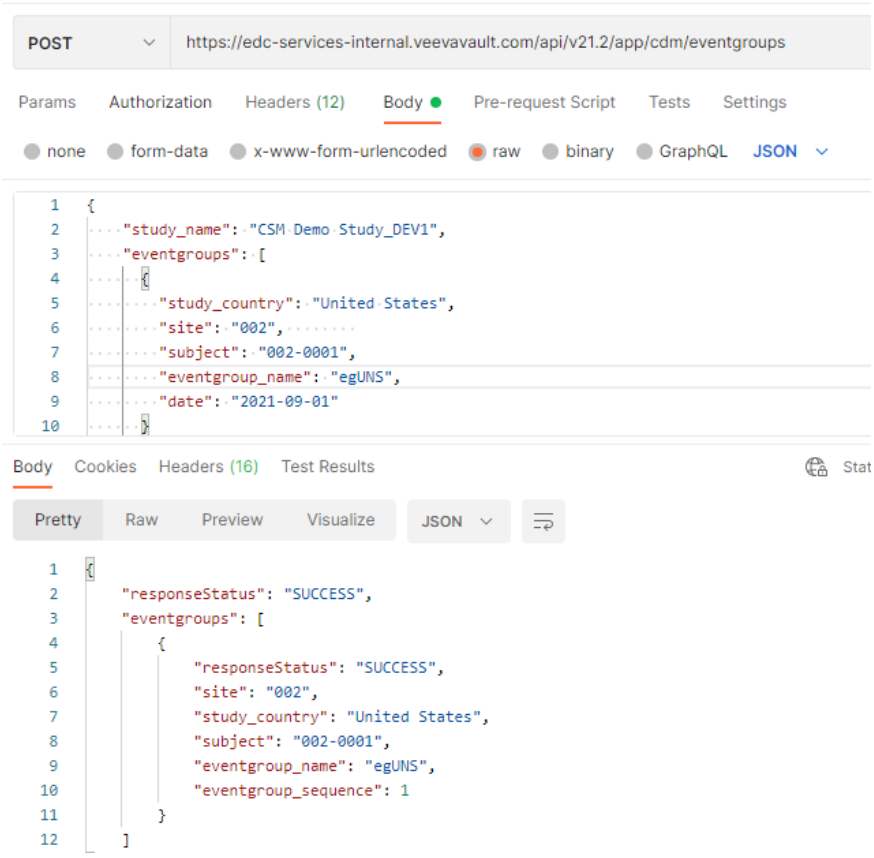
| Description / Request (URL) | Response |
| --- | --- |
| <pre>"eventgroup_name": "egSCREENING"<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-002",<br>"eventgroup_name": "egIRT_INFO"<br>}<br>]<br>}</pre> | <pre>"subject": "101-001",<br>"eventgroup_name": "egSCREENING",<br>"eventgroup_sequence": 1<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egIRT_INFO",<br>"eventgroup_sequence": 1<br>}<br>]<br>}</pre> |

### Flow Based Example

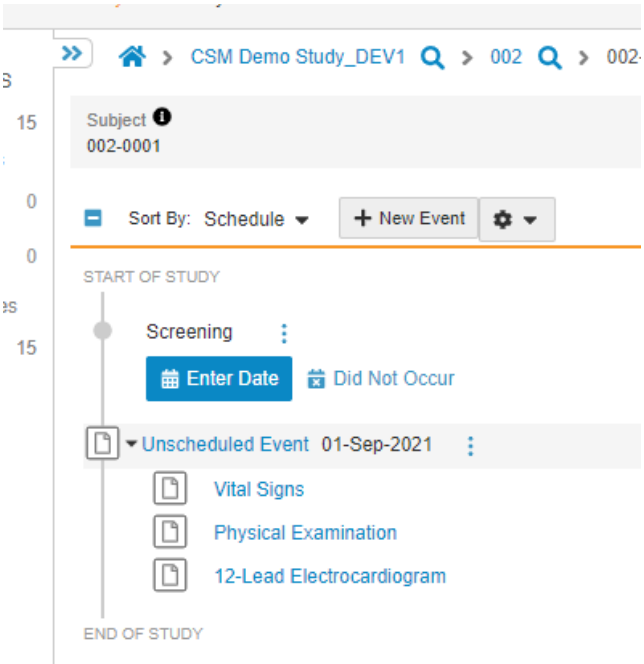
For the study design: (event groups are eg_SCREENING, eg_TREATMENT, eg_COMMONFORMS, eg_UNS, in design name):



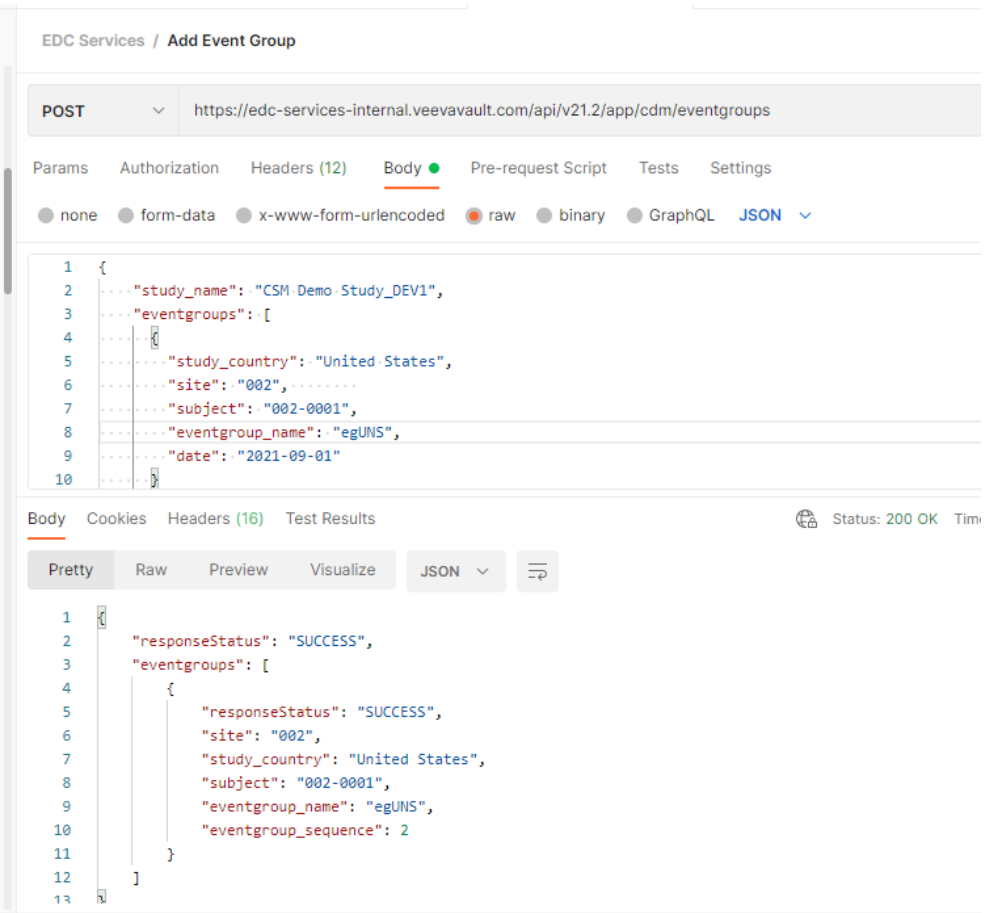

The event group (highlighted above) is of Type = **Unscheduled** and has name **egUNS**. The POST (add) example returns the newly added event group sequence since it repeats:



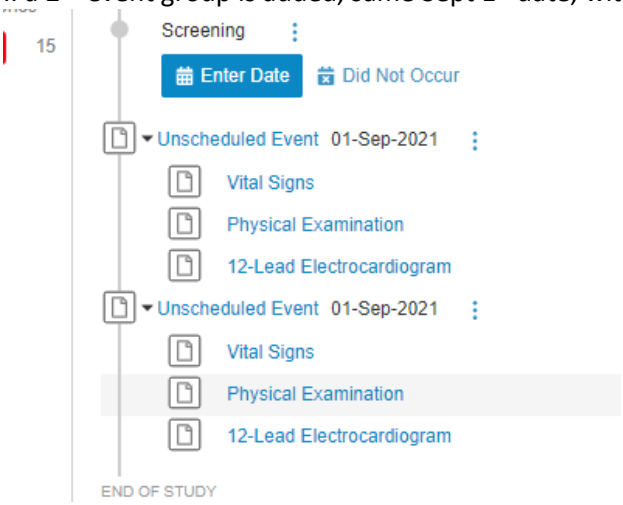

**Result in EDC:**




**WARNING:** The blind call of this endpoint will continue to add more event groups if it repeats, e.g.:




.. a 2nd event group is added, same Sept 1st date, with Event Group Sequence 2 (Not pictured in the UI):




**TIP:** Good practice is to inspect the existing forms / events before attempting the action. Or use the Upsert (**PUT**) for a very specific sequence of the event group, see the next section.

## 7.2 Upsert Event Groups

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#upsert-event-groups |
| Style | **PUT** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/eventgroups |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| study_country | eventgroups | Required | Name of the study country (of the subject's site) |
| site | eventgroups | Required | Name/number of the site |
| subject | eventgroups | Required | Subject to act on |
| eventgroup_name | eventgroups | Required | Design **name** in the study of the event group to add/upsert |
| eventgroup_sequence | eventgroups | Required | |
| date | eventgroups | Optional* | Only applicable to **Unscheduled** type event groups. Format: yyyy-MM-dd |

**Notes**

- Used to add (if doesn't exist yet at that specific sequence), or skip if exists
- For an **Unscheduled** type event group, the API call **must** include an event date (format: **yyyy-MM-dd**) for the event inside the group. All other event group types, the date is not set in this API call, instead using **Set Event Date** (next section)
- **WARNING:** The sequences of event groups should be added in order. Leaving a 'hole' (e.g. Event Group Sequence = 2, where Event Group Sequence = 1 entry does not exist) should not be attempted, and will be enforced as an error.

### Examples

**Description / Request (URL)**

Multiple upserts at the same time

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "eventgroups": [
        {
            "study_country": "United States",
            "site": "101",
            "subject": "101-001",
            "eventgroup_name": "egSCREENING",
            "eventgroup_sequence": 1
        },
        {
            "study_country": "United States",
            "site": "101",
```

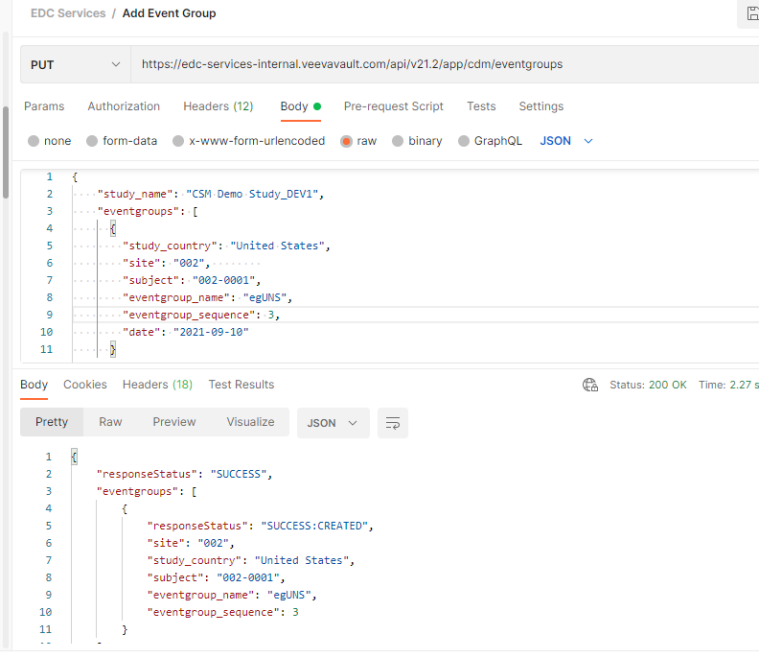
**Response**

```
{
    "responseStatus": "SUCCESS",
    "eventgroups": [
        {
            "responseStatus": "SUCCESS:CREATED",
            "study_country": "United States",
            "site": "101",
            "subject": "101-001",
            "eventgroup_name": "egSCREENING",
            "eventgroup_sequence": 1
        },
        {
            "responseStatus": "SUCCESS:CREATED",
```

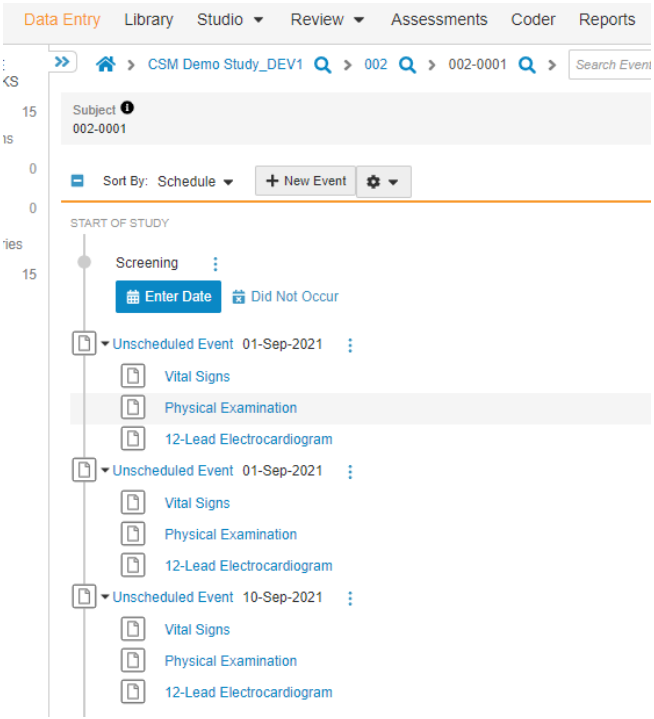
| Description / Request (URL) | Response |
|---|---|
| <pre>"subject": "101-002",<br>"eventgroup_name": "egMAIN_VISITS",<br>"eventgroup_sequence": 1<br>}<br>]<br>}</pre> | <pre>"study_country": "United States",<br>"site": "101",<br>"subject": "101-002",<br>"eventgroup_name": "egMAIN_VISITS",<br>"eventgroup_sequence": 1<br>}<br>]<br>}</pre> |
| New unscheduled type event group, added | |
| <pre>{<br>"study_name": "ABCP-2022-01_DEV1",<br>"eventgroups": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egUNSCHEDULED",<br>"eventgroup_sequence": 2,<br>"date": "2022-04-01"<br>}<br>]<br>}</pre> | <pre>{<br>"responseStatus": "SUCCESS",<br>"eventgroups": [<br>{<br>"responseStatus": "SUCCESS:CREATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egUNSCHEDULED",<br>"eventgroup_sequence": 2<br>}<br>]</pre> |
| Add of two, first exists (skipped), 2nd added | |
| <pre>{<br>"study_name": "ABCP-2022-01_DEV1",<br>"eventgroups": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egUNSCHEDULED",<br>"eventgroup_sequence": 1,<br>"date": "2022-04-01"<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egUNSCHEDULED",<br>"eventgroup_sequence": 2,<br>"date": "2022-05-01"<br>}<br>]<br>}</pre> | <pre>{<br>"responseStatus": "SUCCESS",<br>"eventgroups": [<br>{<br>"responseStatus": "SUCCESS:UPDATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egUNSCHEDULED",<br>"eventgroup_sequence": 1<br>},<br>{<br>"responseStatus": "SUCCESS:CREATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egUNSCHEDULED",<br>"eventgroup_sequence": 2<br>}<br>]<br>}</pre> |

## Flow Based Examples

With two unscheduled event groups already existing, a specific target of adding the 3rd unscheduled event:



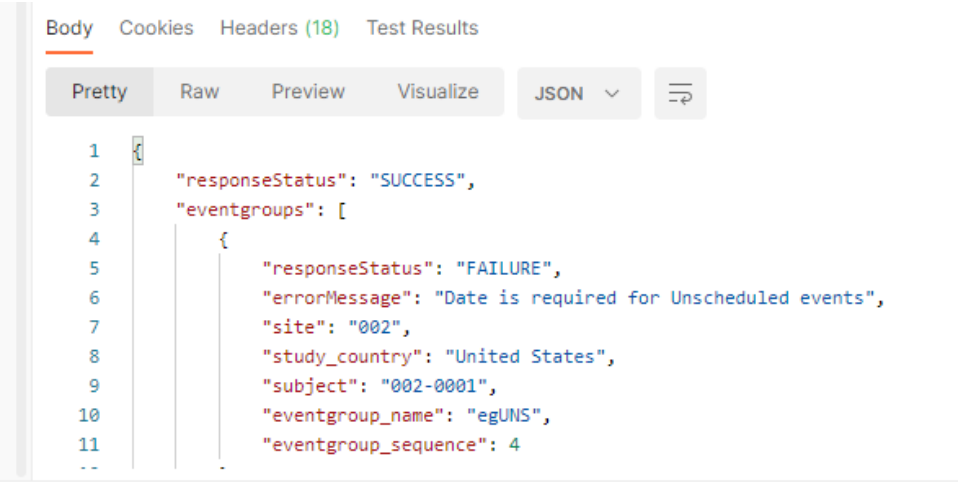

...yields, in EDC:



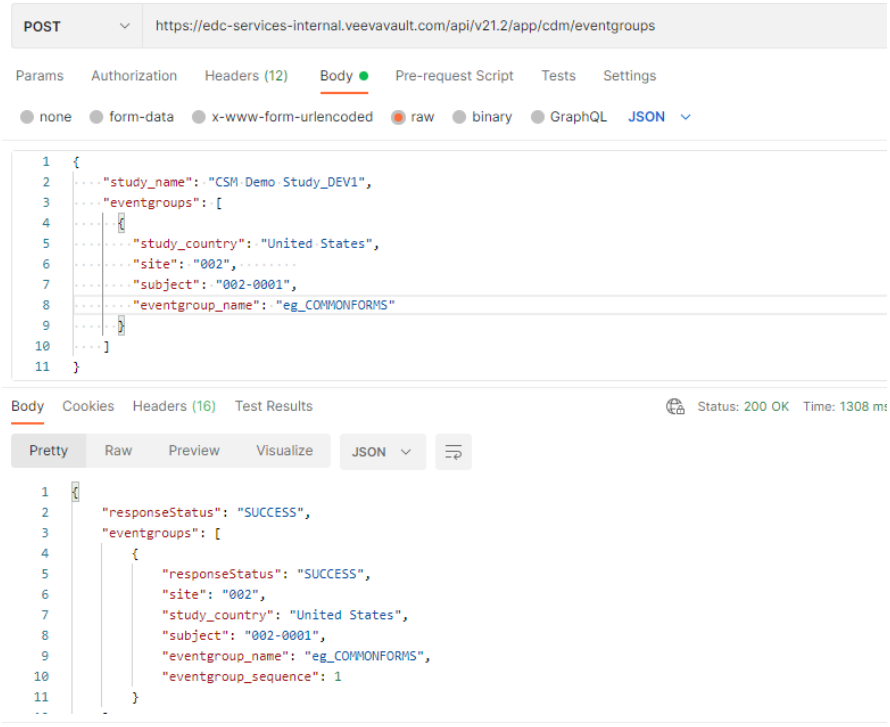

**WARNING:** The **Upsert** of an event group is **not** used to update an existing event group, its first event's date. To update the event date, use / see the Set Event Date API (next section). The use of upsert, when the event group already exists, is merely a 'skip, do nothing' safe call. That is, create/add if not there, otherwise, no changes. This

becomes a safer way to add specific event groups, as **Create Event Group** is always a brand new event group, next in sequence.

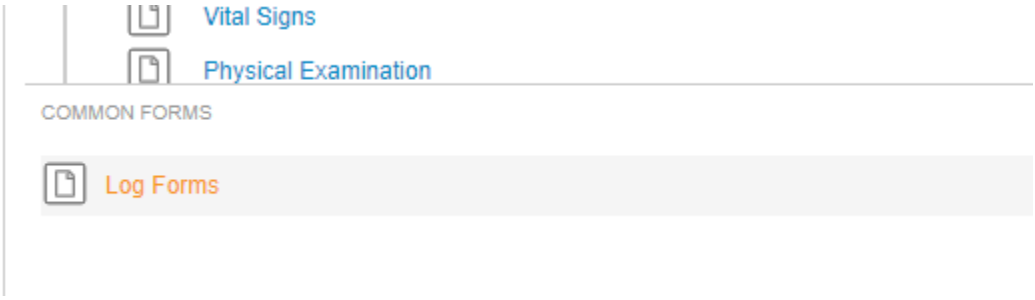
The event date is required for an event group of type = **Unscheduled**, with failed attempt:



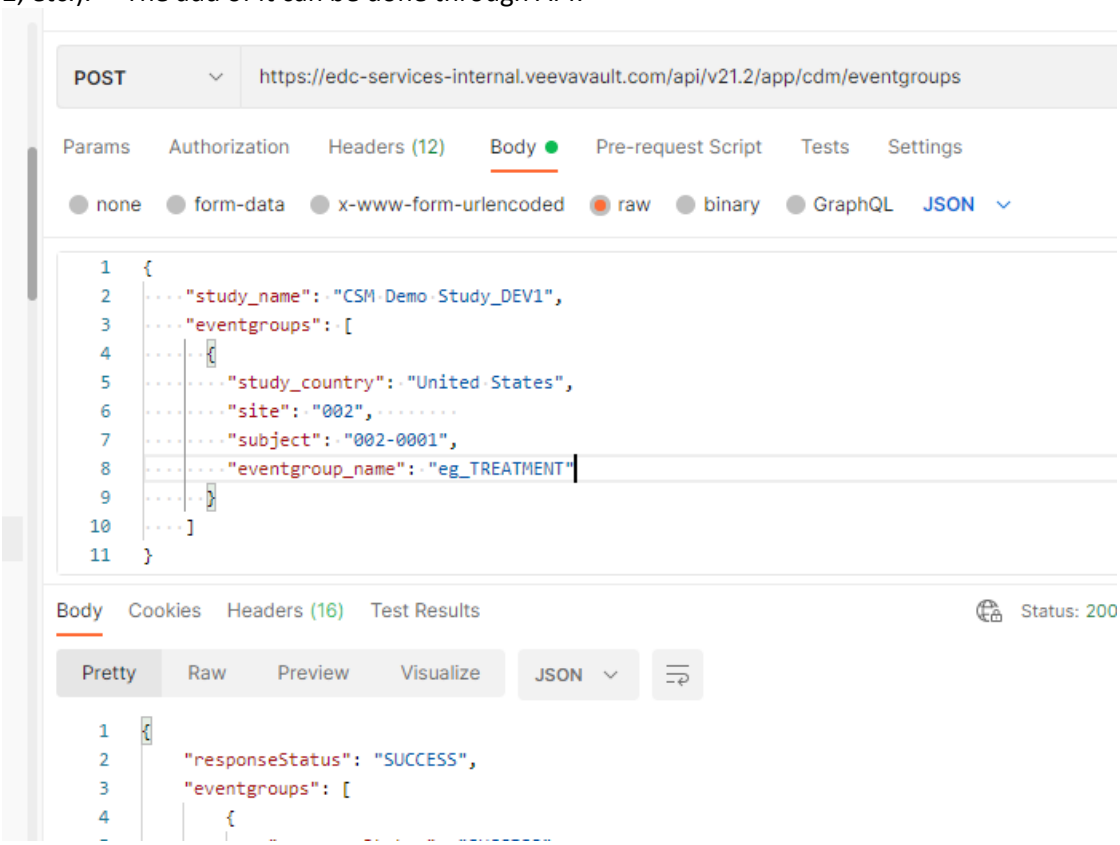
Contrast this to the add of a scheduled event group that only integration (e.g. IRT system) should control / add. For the example study, the **Common Forms** is just such an event group of events/forms. The add is successful:



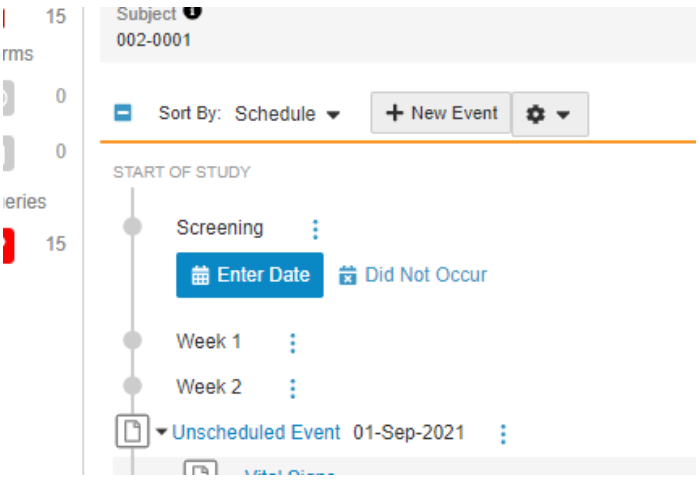

With result an EDC event group with no date set, but awaiting entry into the forms of it:



Lastly, there are occasions to have an API call add an event group, scheduled (like a Common / Logs area) but that will require a date. In the example study the eg_TREATMENT is the main event group of the study (Events Week 1, Week 2, etc.). The add of it can be done through API:




**But**, the date is not set – i.e., for the first event of the event group (like unscheduled types are). To set that, one must follow on with a [Set Event Date] API call (next section)



# 8 Events

## 8.1 Retrieve Events / Forms

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#retrieve-events">https://developer-cdms.veevavault.com/api/22.2/#retrieve-events</a></td></tr>
<tr><td>Style</td><td>GET</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/events</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>Required</td><td>Name of the study country (of the subject's site)</td></tr>
<tr><td>site</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>Optional</td><td>Design name in the study of the event group to filter to</td></tr>
<tr><td>eventgroup_sequence</td><td>Optional</td><td>The specific sequence of an event group</td></tr>
<tr><td>event_name</td><td>Optional</td><td>Design name in the study of the event to filter to</td></tr>
<tr><td>limit</td><td>Optional</td><td>Pagination API - different limit from the default 1000 can be used</td></tr>
<tr><td>offset</td><td>Optional</td><td>Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>• Use this API to retrieve existing events / forms (of the events) for at most, one subject<br>• The largest scope of the retrieval is one subject</td></tr>
</table>

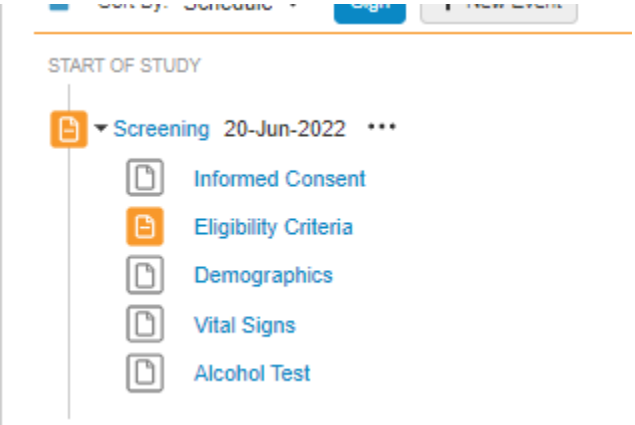
### Examples

<table>
<tr><th>Description / Request (URL)</th><th>Response</th></tr>
<tr><td colspan="2">Get of specific event (1)</td></tr>
<tr><td>../app/cdm/events?study_name=ABCP-2022-01_DEV1&study_country=United States&site=101&subject=SCR-0001&eventgroup_name=egSCR&eventgroup_sequence=1&event_name=evSCR</td><td><pre>{
    "responseStatus": "SUCCESS",
    "responseDetails": {
        "limit": 1000,
        "offset": 0,
        "size": 1,
        "total": 1
    },
    "events": [
        {
            "study_country": "United States",</pre></td></tr>
</table>

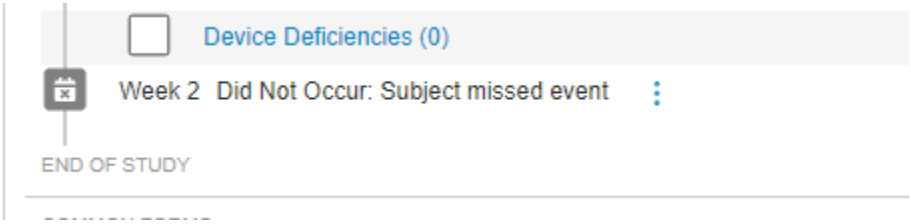
| Description / Request (URL) | Response |
| --- | --- |
| | "site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"locked": false,<br>"frozen": false,<br>"event_date": "2022-06-20",<br>"forms": [<br>{<br>"form_name": "Informed-Consent",<br>"form_sequence": 1,<br>"form_status": "blank__v",<br>"locked": true,<br>"frozen": true,<br>"intentionally_left_blank": false,<br>"intentionally_left_blank_reason": null<br>},<br>{<br>"form_name": "Eligibility-Criteria",<br>"form_sequence": 1,<br>"form_status": "submitted__v",<br>"locked": false,<br>"frozen": true,<br>"intentionally_left_blank": false,<br>"intentionally_left_blank_reason": null<br>},<br>{<br>"form_name": "Demographics",<br>"form_sequence": 1,<br>"form_status": "blank__v",<br>"locked": false,<br>"frozen": true,<br>"intentionally_left_blank": false,<br>"intentionally_left_blank_reason": null<br>},<br>{<br>"form_name": "Vital-Signs",<br>"form_sequence": 1,<br>"form_status": "blank__v",<br>"locked": false,<br>"frozen": true,<br>"intentionally_left_blank": false,<br>"intentionally_left_blank_reason": null<br>}<br>]<br>} |

| Description / Request (URL) | Response |
|---|---|
| | ]<br>} |
| Get of an event that is in planned or did not occur status (no forms yet) | |
| .../app/cdm/events?study_name=ABCP-2022-01_DEV1&study_country=United States&site=101&subject=SCR-0001&eventgroup_name=egMAIN | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 1,<br>"total": 1<br>},<br>"events": [<br>{<br>"study_country": "United States",<br>"site": "102",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egMAIN",<br>"eventgroup_sequence": 1,<br>"event_name": "evVISIT1",<br>"event_sequence": 1,<br>"locked": false,<br>"frozen": false,<br>"event_date": null,<br>"forms": []<br>}<br>]<br>} |

The event date originates from the level above all forms of that event (Jun 20th, 2022 for example):



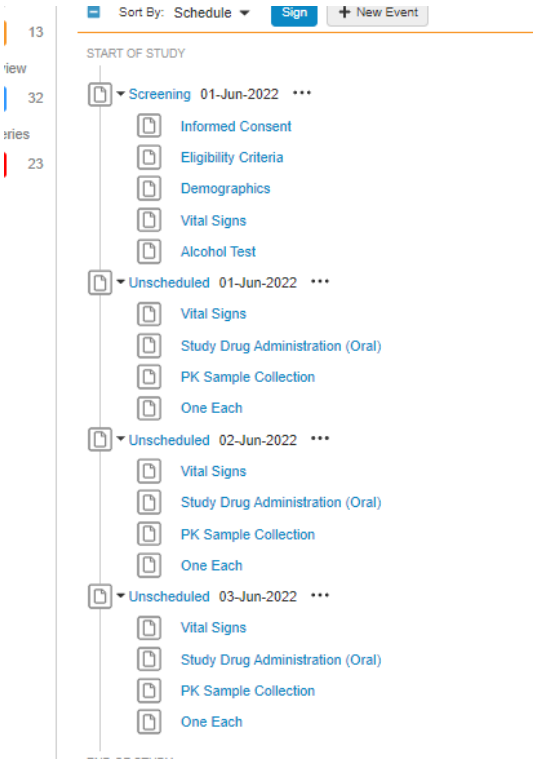

The **Example 2** above includes no forms returned. This can happen when the event status is Planned (no event date yet), or Did Not Occur (specifically marked by site as missed or other reason)




### Form Statuses

| Label | API Return | When / Notes |
|---|---|---|
| Blank | blank__v | The form is not yet visited (no data points / items), or is visited but all fields are their initial blank status. The form can also take on this status after a reset action |
| In Progress | in_progress__v | At least one data point (item) has its first value in the form |
| Submitted | submitted__v | The form is in submitted/completed state. No updates (UI or API) can occur until the form is opened for edit<br><br>**NOTE**: A form can be in submitted status, and be additionally marked Intentionally Left Blank. (i.e. reasons for not having the data) |
| In Progress Post Submit | in_progress_post_submit__v | Like submitted, but the form has been submitted (then reopened) at least once. |

Consider repeating event groups / events, the eventgroup_sequence is key for identification, as well as the event date. Below 3 events – Unscheduled – repeat:




.. with applicable section in the return, noting **eventgroup_sequence: 3** : ( June 3rd, 2022)

Pretty Raw Preview Visualize JSON


```
},
{
    "study_country": "United States",
    "site": "101",
    "subject": "SCR-0005",
    "eventgroup_name": "egUNS",
    "eventgroup_sequence": 3,
    "event_name": "evUNS",
    "event_sequence": 1,
    "locked": false,
    "frozen": false,
    "event_date": "2022-06-03",
    "forms": [
        {
            "form_name": "VS",
            "form_sequence": 1,
            "form_status": "blank__v",
            "locked": false,
            "frozen": false,
            "intentionally_left_blank": false,
            "intentionally_left_blank_reason": null
        },
        {
            "form_name": "EX",
```

## 8.2 Create Events

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#create-events |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/events |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| study_country | events | Required | Name of the study country (of the subject's site) |
| site | events | Required | Name/number of the site |
| subject | events | Required | Subject to act on |
| eventgroup_name | events | Required | Design **name** in the study of the event group where the event resides |
| eventgroup_sequence | events | Optional | The specific sequence of an event group, **1** if omitted |
| event_name | events | Required | Design **name** in the study of the event to add |

Notes

- **Only** used for the addition of an event to an existing event group. It does not also add a non-existent event group. See Add/Upsert Event Group sections for event groups
- Does not set the event date, instead one would use **Set Event Date** (next section)
- The endpoint is used for a dynamic (or 'integration only') event.
- **TIP:** Although the example above is from a non-repeating event group, and **eventgroup_sequence** is not strictly required, it is good practice to always include this value. Repeating event groups, later forms will need specific values.

### Examples

**Description / Request (URL)**

Non-repeating event group where event resides

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "events": [
        {
            "study_country": "United States",
            "site": "101",
            "subject": "101-001",
            "eventgroup_name": "egTREATMENT",
            "event_name": "evV6"
        }
    ]
}
```

**Response**

```
{
    "responseStatus": "SUCCESS",
    "events": [
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
            "subject": "101-001",
            "eventgroup_name": "egTREATMENT",
            "eventgroup_sequence": 1,
            "event_name": "evV6",
            "event_sequence": 1
        }
    ]
}
```

| Description / Request (URL) | Response |
|---|---|
| Add of event inside a repeating event group, the 2nd sequence | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"events": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egCYCLE",<br>"eventgroup_sequence": 2,<br>"event_name": "evDYNAMIC_ASSESS"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"events": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egCYCLE",<br>"eventgroup_sequence": 2,<br>"event_name": "evDYNAMIC_ASSESS",<br>"event_sequence": 1<br>}<br>]<br>} |
| Multiple attempted, one fails due to already existed | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"events": [<br>{<br>"study_country": "United States",<br>"site": "201",<br>"subject": "201-001",<br>"eventgroup_name": "egCYCLE",<br>"eventgroup_sequence": 2,<br>"event_name": "evDYNAMIC_ASSESS"<br>},<br>{<br>"study_country": "United States",<br>"site": "201",<br>"subject": "201-002",<br>"eventgroup_name": "egCYCLE",<br>"eventgroup_sequence": 2,<br>"event_name": "evDYNAMIC_ASSESS"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"events": [<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "Cannot create event because it already exists",<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-001",<br>"eventgroup_name": "Days-6-14",<br>"eventgroup_sequence": 2,<br>"event_name": "evDYNAMIC_ASSESS"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-002",<br>"eventgroup_name": "Days-6-14",<br>"eventgroup_sequence": 2,<br>"event_name": "evDYNAMIC_ASSESS",<br>"event_sequence": 1<br>}<br>]<br>} |

## 8.3 Set Event Date

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#set-event-date">https://developer-cdms.veevavault.com/api/22.2/#set-event-date</a></td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/events/actions/setdate</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>events</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>events</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>events</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>events</td><td>Required</td><td>Design name in the study of the event group where the event resides</td></tr>
<tr><td>eventgroup_sequence</td><td>events</td><td>Optional</td><td>The specific sequence of an event group, 1 if omitted</td></tr>
<tr><td>event_name</td><td>events</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>date</td><td>events</td><td>Required</td><td>The date to set, yyyy-MM-dd format</td></tr>
<tr><td>change_reason</td><td>events</td><td>Depends</td><td>The reason for date change. Only required when the event already has an event date set</td></tr>
<tr><td>allow_planned_override</td><td>events</td><td>Optional</td><td>Default is false. When passing true, a date can be set that is out of the planned window. If false, such a date is returned as an error</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Used to set an event date to an existing event. (does not add the event or event group the event resides, that is done in other API calls first)</li>
<li>Date format is yyyy-MM-dd</li>
<li>The endpoint call (as POST) is used for both setting a date when none exists, plus updating an event that has an existing date value. (PUT is not used for update)</li>
<li>TIP: Although the example above is from a non-repeating event group, and eventgroup_sequence is not strictly required, it is good practice to always include this value. Repeating event groups, later forms will need specific values.</li>
</ul>
</td></tr>
</table>

### Examples

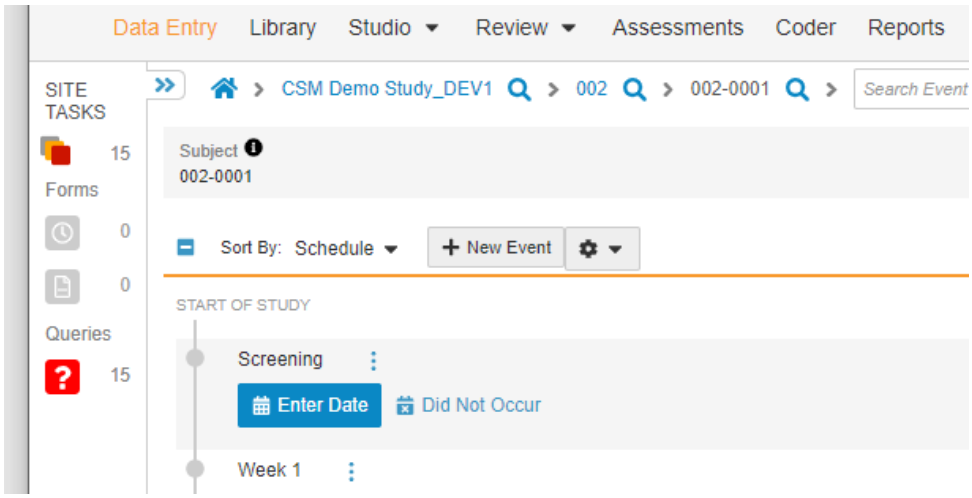
| Description / Request (URL) | Response |
|---|---|
| Non-repeating event group (where event resides) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"events": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egTREATMENT",<br>"event_name": "evV6" ,<br>"date": "2022-06-01"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"events": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egTREATMENT",<br>"eventgroup_sequence": 1,<br>"event_name": "evV6",<br>"event_sequence": 1<br>}<br>]<br>} |
| Two attempted, one succeeds, one fails | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"events": [<br>{<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-001",<br>"eventgroup_name": "egTREATMENT",<br>"event_name": "evDay1" ,<br>"date": "2022-06-01"<br>},<br>{<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-005",<br>"eventgroup_name": "egTREATMENT",<br>"event_name": "evDay1" ,<br>"date": "2022-06-05"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"events": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-001",<br>"eventgroup_name": "egTREATMENT",<br>"eventgroup_sequence": 1,<br>"event_name": "evDay1",<br>"event_sequence": 1,<br>"date": "2022-06-01"<br>},<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "[Subject] with name [201-005] not found",<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-005",<br>"eventgroup_name": "egTREATMENT",<br>"eventgroup_sequence": 1,<br>"event_name": "evDay1",<br>"event_sequence": 1,<br>"date": "2022-06-05"<br>} |

| Description / Request (URL) | Response |
|---|---|
| | ]<br>} |
| Setting a specific sequence for repeating event group (events within) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"events": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egCYCLE",<br>"eventgroup_sequence": 2,<br>"event_name": "evDYNAMIC_ASSESS",<br>"date": "2022-06-01"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"events": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egCYCLE",<br>"eventgroup_sequence": 2,<br>"event_name": "evDYNAMIC_ASSESS",<br>"event_sequence": 1,<br>"date": "2022-06-01"<br>}<br>]<br>} |
| Override where the event date would otherwise be out of window | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"events": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egTREATMENT",<br>"event_name": "evV6" ,<br>"date": "2022-06-01",<br>"allow_planneddate_override": **true**<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"events": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egTREATMENT",<br>"eventgroup_sequence": 1,<br>"event_name": "evV6",<br>"event_sequence": 1<br>}<br>]<br>} |
| Change of date (requires reason) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"events": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001",<br>"eventgroup_name": "egTREATMENT", | {<br>"responseStatus": "SUCCESS",<br>"events": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "101-001", |

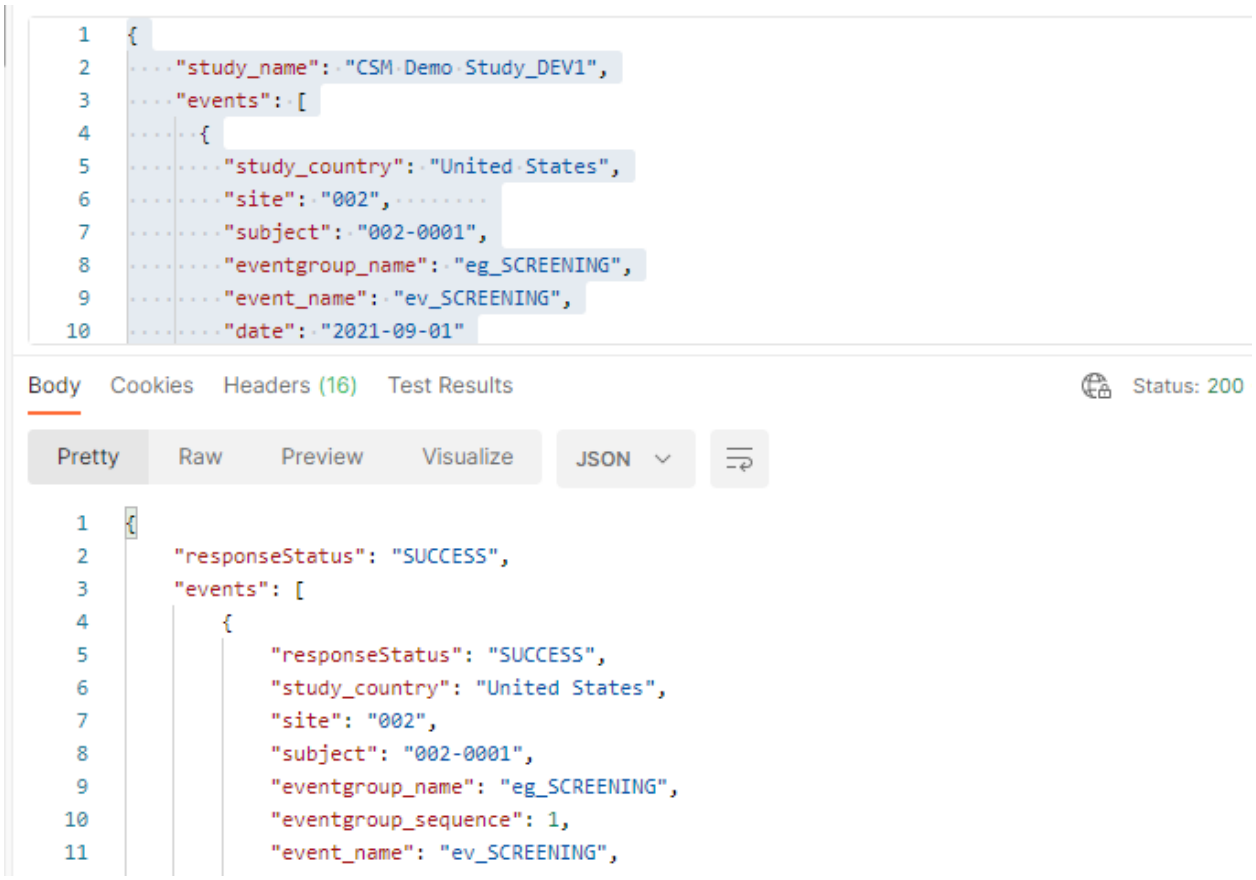
| Description / Request (URL) | Response |
| --- | --- |
| <pre>"event_name": "evV6" ,<br>"date": "2022-06-02",<br>"change_reason": "Change by IRT system"<br>}<br>]<br>}</pre> | <pre>"eventgroup_name": "egTREATMENT",<br>"eventgroup_sequence": 1,<br>"event_name": "evV6",<br>"event_sequence": 1<br>}<br>]<br>}</pre> |

### Flow Based Example

For this subject in the EDC UI, there is no **Screening** date yet:




Performing the API call to set **Sept 1, 2021,** as **Screening** event date:



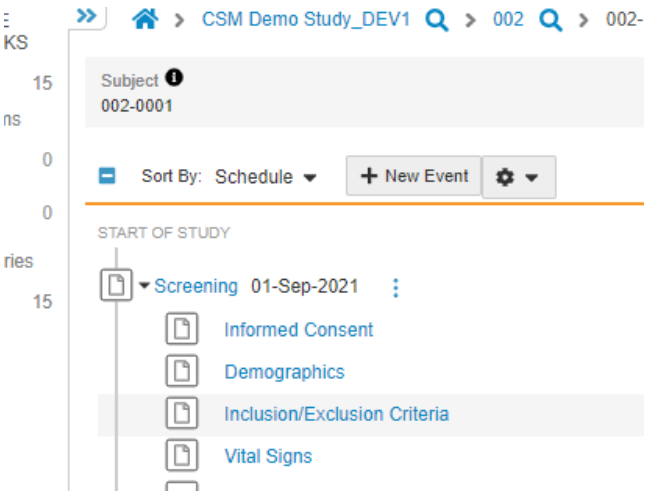
```
{
    "study_name": "CSM Demo Study_DEV1",
    "events": [
        {
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "event_name": "ev_SCREENING",
            "date": "2021-09-01"
```

Body Cookies Headers (16) Test Results Status: 200

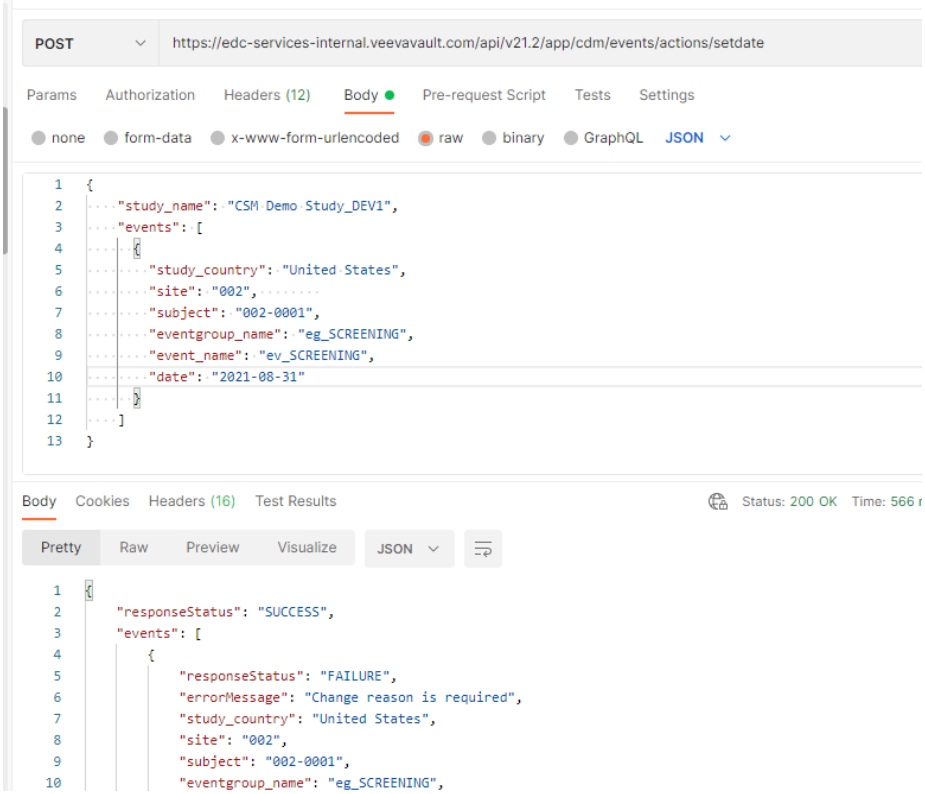
Pretty Raw Preview Visualize JSON

```
{
    "responseStatus": "SUCCESS",
    "events": [
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
```

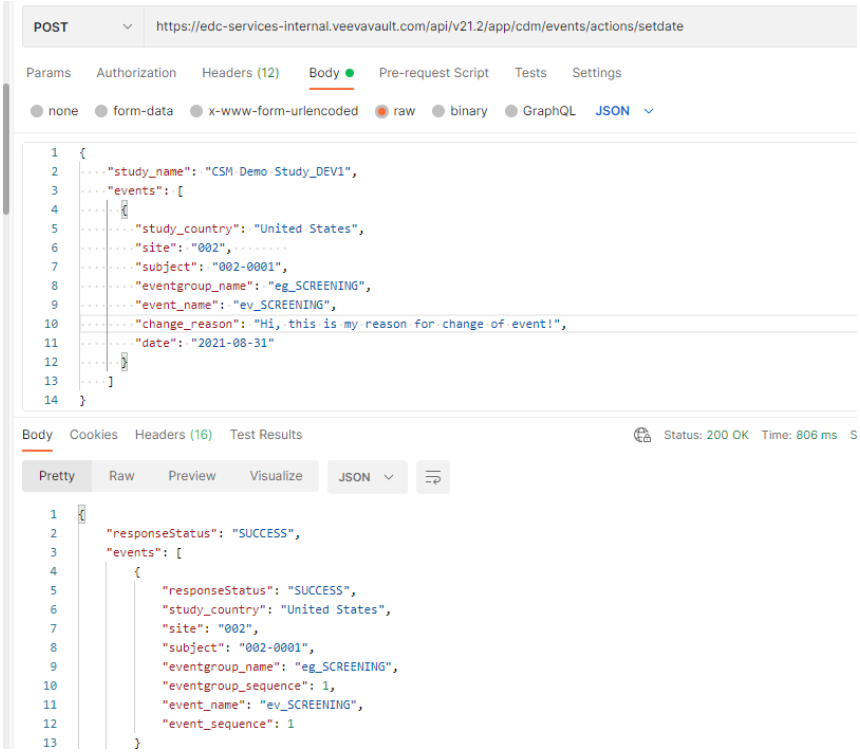
...which yields – in the EDC UI – the event now built out with the forms the study scheduled prescribes:



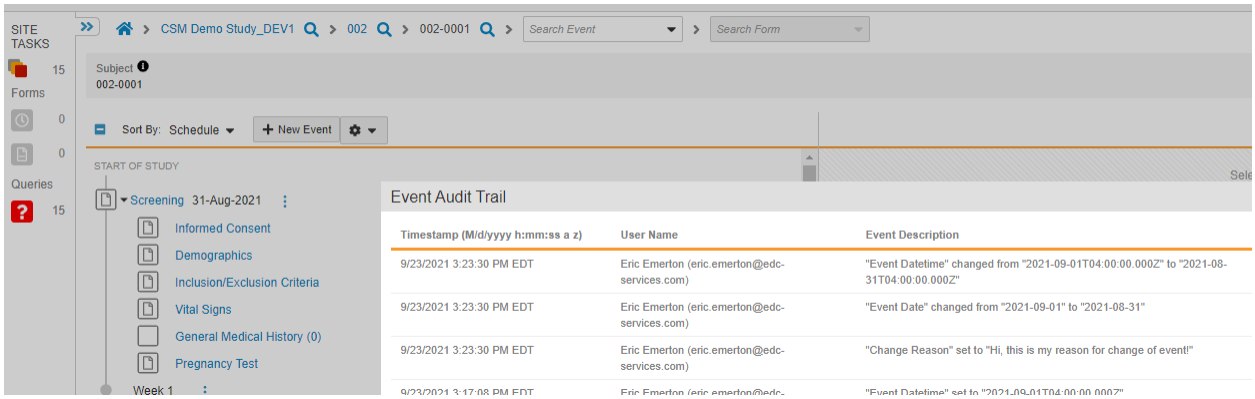

**IMPORTANT:** This step is key to then being able to set data to forms of an event. If the event does not have a date yet, then no forms will exist (for scheduled, non-log events).
For changing the date of an existing event, the same call can be used:




... **note (!!)** the inner **FAILURE**. Data points changed (after initial entry) require change reasons. Adding the parameter **change_reason**, with any custom reason:




.. yielding – in the EDC UI - the change to Aug 31st, with reason passed via API noted in audit trail:


SITE TASKS
15
Forms
0
0
Queries
15
CSM Demo Study_DEV1
002
002-0001
Search Event
Search Form
Subject
002-0001
Sort By: Schedule
New Event
START OF STUDY
Screening 31-Aug-2021
Informed Consent
Demographics
Inclusion/Exclusion Criteria
Vital Signs
General Medical History (0)
Pregnancy Test
Week 1
Event Audit Trail
Timestamp (M/d/yyyy h:mm:ss a z)
User Name
Event Description
9/23/2021 3:23:30 PM EDT
Eric Emerton (eric.emerton@edc-services.com)
"Event Datetime" changed from "2021-09-01T04:00:00.000Z" to "2021-08-31T04:00:00.000Z"
9/23/2021 3:23:30 PM EDT
Eric Emerton (eric.emerton@edc-services.com)
"Event Date" changed from "2021-09-01" to "2021-08-31"
9/23/2021 3:23:30 PM EDT
Eric Emerton (eric.emerton@edc-services.com)
"Change Reason" set to "Hi, this is my reason for change of event!"

## 8.4 Set Event as Did Not Occur

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#set-event-as-did-not-occur |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/events/actions/didnotoccur |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| study_country | events | Required | Name of the study country (of the subject's site) |
| site | events | Required | Name/number of the site |
| subject | events | Required | Subject to act on |
| eventgroup_name | events | Required | Design **name** in the study of the event group where the event resides |
| eventgroup_sequence | events | Optional | The specific sequence of an event group, **1** if omitted |
| event_name | events | Required | Design **name** in the study of the event |
| change_reason | events | Required | The reason for date change. Only **required** when the event already has an event date set |

**Notes**

- Used to mark an entire event with status = **Did Not Occur**, which then allows for visits after to be worked with, dates set.
- Change of reason can be any text, although certainly consider being consistent with the text usage.
- **TIP:** Although the example above is from a non-repeating event group, and **eventgroup_sequence** is not strictly required, it is good practice to always include this value. Repeating event groups, later forms will need specific values.

### Examples

| Description / Request (URL) | Response |
|---|---|
| One and successful | |

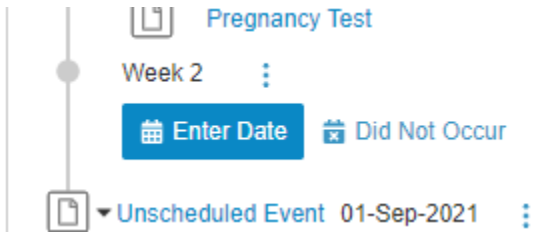
```
{
    "study_name": "ABCP-2022-01_DEV1",
    "events": [
        {
            "study_country": "United States",
            "site": "101",
            "subject": "101-001",
            "eventgroup_name": "egTREATMENT",
            "event_name": "evV6" ,
            "change_reason": "Missed visit"
        }
```

```
{
    "responseStatus": "SUCCESS",
    "events": [
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
            "subject": "101-001",
            "eventgroup_name": "egTREATMENT",
            "eventgroup_sequence": 1,
            "event_name": "evV6",
```

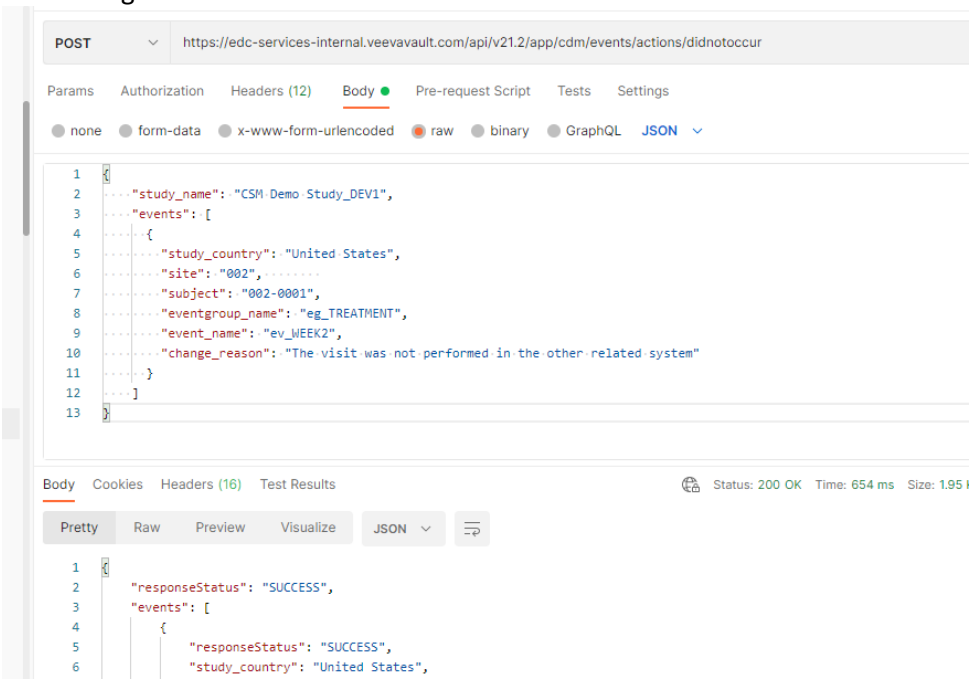
| Description / Request (URL) | Response |
|---|---|
| ]<br>} | "event_sequence": 1<br>}<br>]<br>} |
| Multiple, one succeeds, one fails | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"events": [<br>{<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-001",<br>"eventgroup_name": "egTREATMENT",<br>"event_name": "evV6" ,<br>"change_reason": "Missed visit"<br>},<br>{<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-001",<br>"eventgroup_name": "egTREATMENT",<br>"event_name": "evV6" ,<br>"change_reason": "Missed visit"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"events": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-001",<br>"eventgroup_name": "egTREATMENT",<br>"eventgroup_sequence": 1,<br>"event_name": "evV6",<br>"event_sequence": 1<br>},<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "[Subject] with name [201-005] not found",<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "201-005",<br>"eventgroup_name": "egTREATMENT",<br>"eventgroup_sequence": 1,<br>"event_name": "evV6",<br>"event_sequence": 1<br>}<br>]<br>} |

## Flow Based Examples

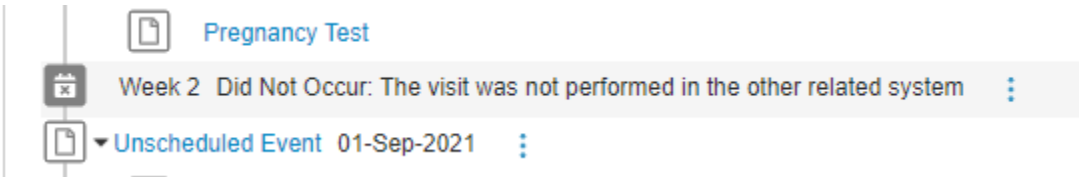
For the EDC visit without a date: (Week 2)




The setting of **Did Not Occu**r status + reason:




…resulting in the EDC UI showing status + reason:



# 9 Forms

## 9.1 Retrieve Forms / Item Data

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#retrieve-forms |
| Style | **GET** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/forms |
| Parameters | |

| Name | Req/Opt | Notes |
|---|---|---|
| study_name | Required | Name of the study |
| study_country | Required | Name of the study country (of the subject's site) |
| site | Required | Name/number of the site |
| subject | Required | Subject to act on |
| eventgroup_name | Required | Design **name** in the study of the event group to filter to |
| eventgroup_sequence | Optional | The specific sequence of an event group |
| event_name | Required | Design **name** in the study of the event to filter to |
| form_name | Optional | Design **name** in the study of the form to filter to |
| form_sequence | Optional | The specific form sequence (if a repeating form) to filter to. (1 is assumed if omitted) |
| limit | Optional | Pagination API - different limit from the default 1000 can be used |
| offset | Optional | Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd |

Notes

- Use this API to retrieve existing forms – their data (items, itemgroups) of one subject's event/visit.
- The details of the form's status, frozen/locked/ILB are also returned, in addition to the data on the form.
- The largest scope of the retrieval is one subject, and one event of the subject (the forms/data in that event)
- At present, there is no ability for a subset of form data, i.e., just certain fields from the filter. Forms of the event can be filtered, but all data of the form(s) is returned.

### Examples

| Description / Request (URL) | Response |
|---|---|
| Return a Demography form | |
| ....app/cdm/forms?study_name=ABCP-2022-01_DEV1&study_country=Canada&site=201&subject=SCR-0001&eventgroup_name=egSCR&eventgroup_sequence=1&event_n | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000, |

| Description / Request (URL) | Response |
| --- | --- |
| ame=evSCR&form_name=Demographics | "offset": 0,<br>"size": 1,<br>"total": 1<br>},<br>"forms": [<br>{<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"event_external_id": null,<br>"form_name": "Demographics",<br>"form_sequence": 1,<br>"form_status": "submitted__v",<br>"locked": false,<br>"frozen": false,<br>"intentionally_left_blank": false,<br>"intentionally_left_blank_reason": null,<br>"itemgroups": [<br>{<br>"itemgroup_name": "ig-DM",<br>"itemgroup_sequence": 1,<br>"itemgroup_external_id": "ig.DM",<br>"items": [<br>{<br>"item_name": "CHILDPSP",<br>"value": null<br>},<br>{<br>"item_name": "RACE_OTHER",<br>"value": "true"<br>},<br>{<br>"item_name": "RACE_NHORPI",<br>"value": "false"<br>},<br>{<br>"item_name": "RACE_AIORAN",<br>"value": null<br>},<br>:<br>:<br>:<br>{<br>"item_name": "CHILDSP", |

| Description / Request (URL) | Response |
|---|---|
| | "value": "SURGICAL PROCEDURE"<br>}<br>]<br>}<br>]<br>}<br>]<br>} |
| Return data from a form with one of every type of item | |
| ....app/cdm/forms?study_name=ABCP-2022-01_DEV1&study_country=Canada&site=201&subject=SCR-0001&eventgroup_name=egUNS&eventgroup_sequence=1&event_name=evUNS&form_name=OE | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 1,<br>"total": 1<br>},<br>"forms": [<br>{<br>"study_country": "Canada",<br>"site": "201",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 1,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"event_external_id": null,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"form_status": "submitted__v",<br>"locked": false,<br>"frozen": false,<br>"intentionally_left_blank": false,<br>"intentionally_left_blank_reason": null,<br>"itemgroups": [<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"itemgroup_external_id": "igOE",<br>"items": [<br>{<br>"item_name": "URL_FIELD",<br>"value": "https://google.com"<br>},<br>{<br>"item_name": "LONGTXT",<br>"value": "Here is some long text with a line return, etc\nand |

| Description / Request (URL) | Response |
| --- | --- |


```
more\nand more\nand so on..   the end"
                        },
                        {
                            "item_name": "DATEONLY_UNKS",
                            "value": "UN-Jun-2022"
                        },
                        {
                            "item_name": "INTFIELD",
                            "value": "23"
                        },
                        {
                            "item_name": "DT_WITH_UNKS",
                            "value": "UN-UNK-2022 UN:UN"
                        },
                        {
                            "item_name": "DATEONLY",
                            "value": "07-Jun-2022"
                        },
                        {
                            "item_name": "SHORTTXT",
                            "value": "Some text here"
                        },
                        {
                            "item_name": "DATEANDTIME",
                            "value": "06-Jun-2022 13:30"
                        },
                        {
                            "item_name": "CL",
                            "value": "Y"
                        },
                        {
                            "item_name": "DECIMAL",
                            "value": "3.2"
                        },
                        {
                            "item_name": "FORM_LINK_FIELD",
                            "value": "OPT000000021003"
                        },
                        {
                            "item_name": "UCLFIELD",
                            "value": "0.45"
                        },
                        {
                            "item_name": "CB",
                            "value": "true"
                        }
                    ]
                }
```

| Description / Request (URL) | Response |
| --- | --- |
| | ]<br>}<br>]<br>} |

## 9.2 Create Forms

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#create-forms">https://developer-cdms.veevavault.com/api/22.2/#create-forms</a></td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/forms</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>forms</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>forms</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>forms</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>forms</td><td>Required</td><td>Design name in the study of the event group</td></tr>
<tr><td>eventgroup_sequence</td><td>forms</td><td>Optional</td><td>The specific sequence of an event group. 1 assumed if omitted</td></tr>
<tr><td>event_name</td><td>forms</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>form_name</td><td>forms</td><td>Required</td><td>Design name in the study of the form to add</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is for repeating form area of the study schedule, to add new ones as necessary</li>
<li>It can also be used for a specific dynamic form area of the study design, to an event that already exists. (event non repeating)</li>
<li>When the form does not yet exist, this action MUST be performed first, prior to setting data on the form. (Combination API(s) are part of coming release)</li>
<li>Like many other API calls, batch add of many is allowed in one call. (Just more entries in the JSON array for forms)</li>
<li>Like event groups, the POST / create will blindly add the ‘next’, in sequence. It's good practice to inspect what forms exist already as part of a sequence of API calls. Or one can use the PUT / upsert instead, passing a specific form_sequence – which will create if not present or skip/do nothing when the sequence does exist. (see next section Upsert Forms)</li>
<li>TIP: For a non-repeating event group, and eventgroup_sequence not strictly required, it is good practice to always include this value. Repeating event groups, later forms will need specific values.</li>
</ul>
</td></tr>
</table>

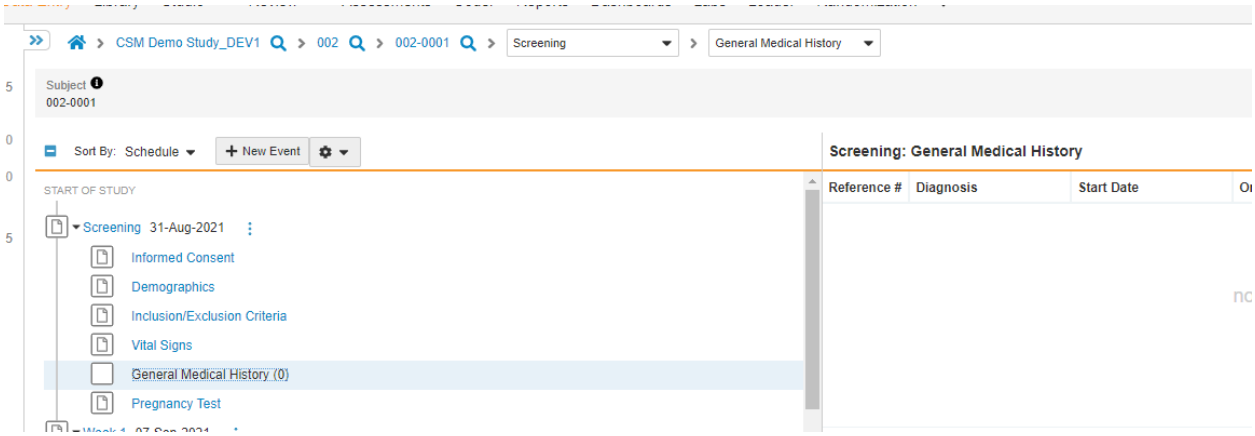
### Examples

| Description / Request (URL) | Response |
|---|---|
| Add of one form | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"forms": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"forms": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 1<br>}<br>]<br>} |
| Add of two more forms, N+1 and N+2 are added (when current N exist) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"forms": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE"<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"forms": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 2<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE", |

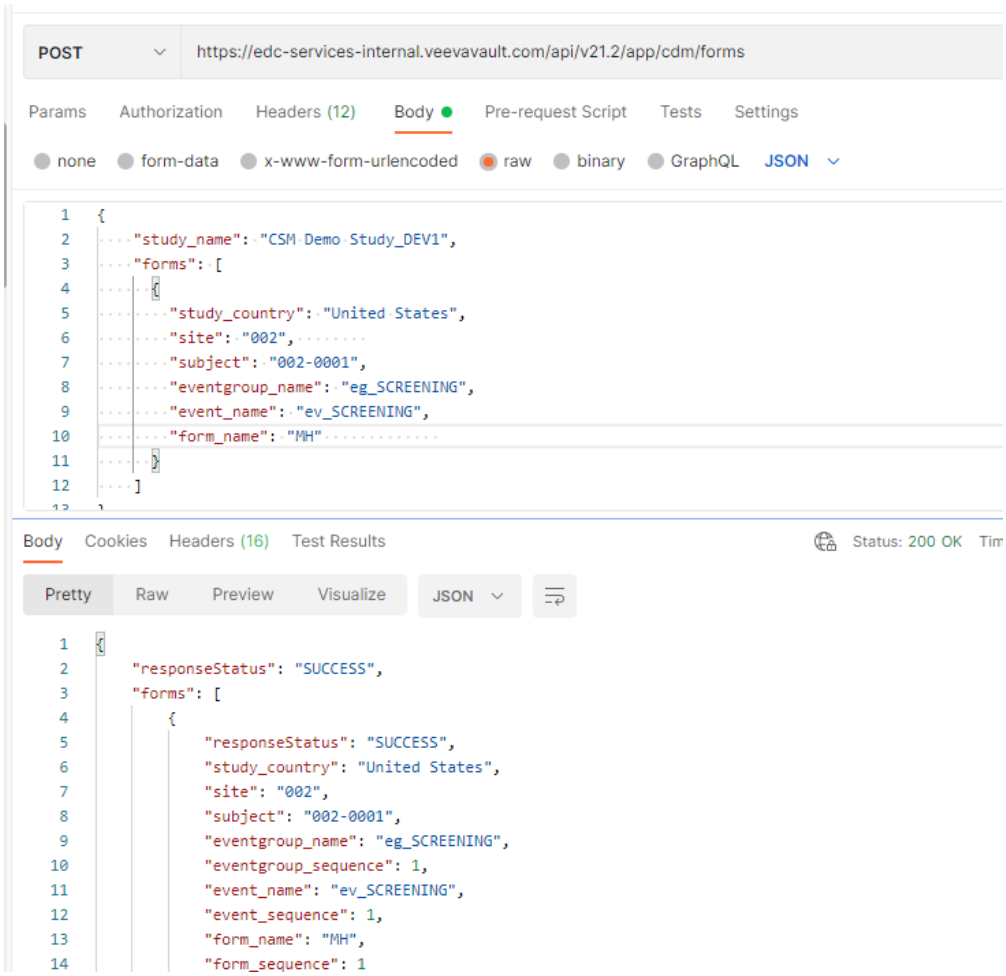
| Description / Request (URL) | Response |
|---|---|
| | "form_sequence": 3<br>}<br>]<br>} |
| Attempt to add two, but the last violates the maximum repeats of that design location in the study | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"forms": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE"<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"forms": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 4<br>},<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "Maximum number of repeating records reached",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE"<br>}<br>]<br>} |

## Flow Based Examples

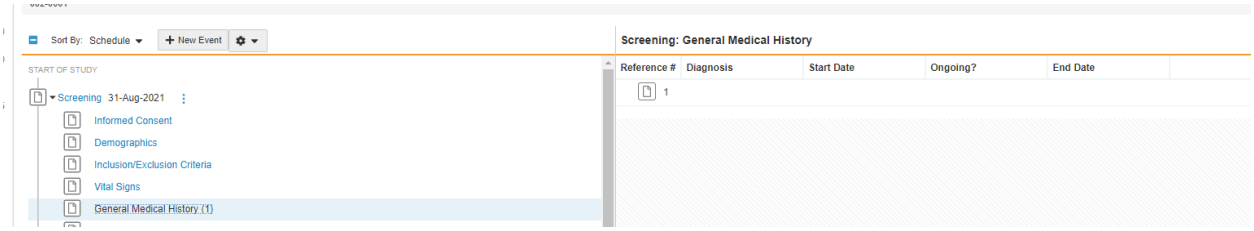
Consider this EDC form area, **General Medical History**, a repeating section:




The location, design name wise (use SDS or other to get mappings) is Event Group = **eg_SCREENING**, Event = **ev_SCREENING**, Form = **MH**. The attempt to add a new form:

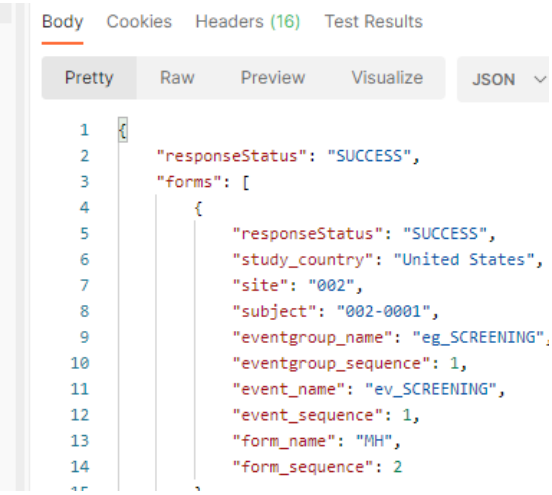


…which yields the new form in the EDC UI (blank):




The visual **'Reference #'** in the EDC UI is the API **form_sequence**.

Calling the API again (blindly):




.. note the **form_sequence**: **2** – i.e., a 2nd form is/was added. This can be dangerous in sequences where data is being updated, so inspecting existing forms is always recommended.

## 9.3 Upsert Forms

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#upsert-forms">https://developer-cdms.veevavault.com/api/22.2/#upsert-forms</a></td></tr>
<tr><td>Style</td><td>PUT</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/forms</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>forms</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>forms</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>forms</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>forms</td><td>Required</td><td>Design name in the study of the event group</td></tr>
<tr><td>eventgroup_sequence</td><td>forms</td><td>Optional</td><td>The specific sequence of an event group. 1 assumed if omitted</td></tr>
<tr><td>event_name</td><td>forms</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>form_name</td><td>forms</td><td>Required</td><td>Design name in the study of the form to add</td></tr>
<tr><td>form_sequence</td><td>forms</td><td>Required</td><td>The specific sequence to add (or skip, if already exists)</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is for repeating form area of the study schedule, to add new ones as necessary</li>
<li>It can also be used for a specific dynamic form area of the study design, to an event that already exists. (event non repeating)</li>
<li>When the form does not yet exist, this action MUST be performed first, prior to setting data on the form. (Combination API(s) are part of coming release)</li>
<li>Like many other API calls, batch add of many is allowed in one call. (Just more entries in the JSON array for forms)</li>
<li>The Add Forms (previous section) is the equivalent of clicking Add in the EDC UI, receiving the ‘next’ in sequence. This endpoint differs in that one is asking for a very specific sequence to be added (or skipped in add, if existing).</li>
<li>TIP: For a non-repeating event group, and eventgroup_sequence not strictly required, it is good practice to always include this value. Repeating event groups, later forms will need specific values.</li>
<li>WARNING: The sequences of forms should be added in order. Leaving a ‘hole’ (e.g. Form Sequence = 2, where Form Sequence = 1 entry does not exist) should be avoided, and will be enforced as an error at this or later release.</li>
</ul>
</td></tr>
</table>

### Examples

| Description / Request (URL) | Response |
|---|---|
| Add of one (doesn't currently exist) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"forms": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE".<br>"form_sequence": 1<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"forms": [<br>{<br>"responseStatus": "SUCCESS:CREATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 1<br>}<br>]<br>} |
| Attempt Upsert at location where form exists already (Essentially SKIPPED) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"forms": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE".<br>"form_sequence": 1<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"forms": [<br>{<br>"responseStatus": "SUCCESS:UPDATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 1<br>}<br>]<br>} |
| Add of many, with the last hitting the maximum repeats of study design at that location | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"forms": [<br>{<br>"study_country": "United States", | {<br>"responseStatus": "SUCCESS",<br>"forms": [<br>{ |

| Description / Request (URL) | Response |
| --- | --- |
| <pre>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE".<br>"form_sequence": 4<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE".<br>"form_sequence": 5<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE".<br>"form_sequence": 6<br>}<br>]<br>}</pre> | <pre>"responseStatus": "SUCCESS:CREATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 4<br>},<br>{<br>"responseStatus": "SUCCESS:CREATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 5<br>},<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "Maximum number of repeating records reached",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 6<br>}<br>]<br>}</pre> |

## 9.4 Create Item Groups

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#create-item-groups |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/itemgroups |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| study_country | itemgroups | Required | Name of the study country (of the subject's site) |
| site | itemgroups | Required | Name/number of the site |
| subject | itemgroups | Required | Subject to act on |
| eventgroup_name | itemgroups | Required | Design **name** in the study of the event group |
| eventgroup_sequence | itemgroups | Optional | The specific sequence of an event group. 1 assumed if omitted |
| event_name | itemgroups | Required | Design **name** in the study of the event |
| form_name | itemgroups | Required | Design **name** in the study of the form |
| form_sequence | itemgroups | Optional | The specific sequence of the form where add will occur. 1 assumed if omitted |
| itemgroup_name | itemgroups | Required | Design **name** in the study of the item group to add |

Notes

- The action is for repeating item groups, within a form that are repeating in nature (only).
- The endpoint is the equivalent of the EDC UI **Add Section** button, i.e. adds the 'next' in sequence
- The **Maximum Repeats** value of the study design will be enforced when attempting to add more than allowed by study design.
- When later setting data, the **Upsert Form Data** (later section) will automatically add any item groups that do not exist (and are repeating).
- Like many other API calls, batch add of many is allowed in one call. (Just more entries in the JSON array for **itemgroups**)
- Like event groups, forms that repeat, the **POST** / create will blindly add the 'next', in sequence. It's good practice to inspect what forms exist already as part of a sequence of API calls. Or one can use the **PUT** / Upsert instead (next section), passing a specific **itemgroup_sequence** – which will create if not present or skip/do nothing when the sequence does exist.

### Examples

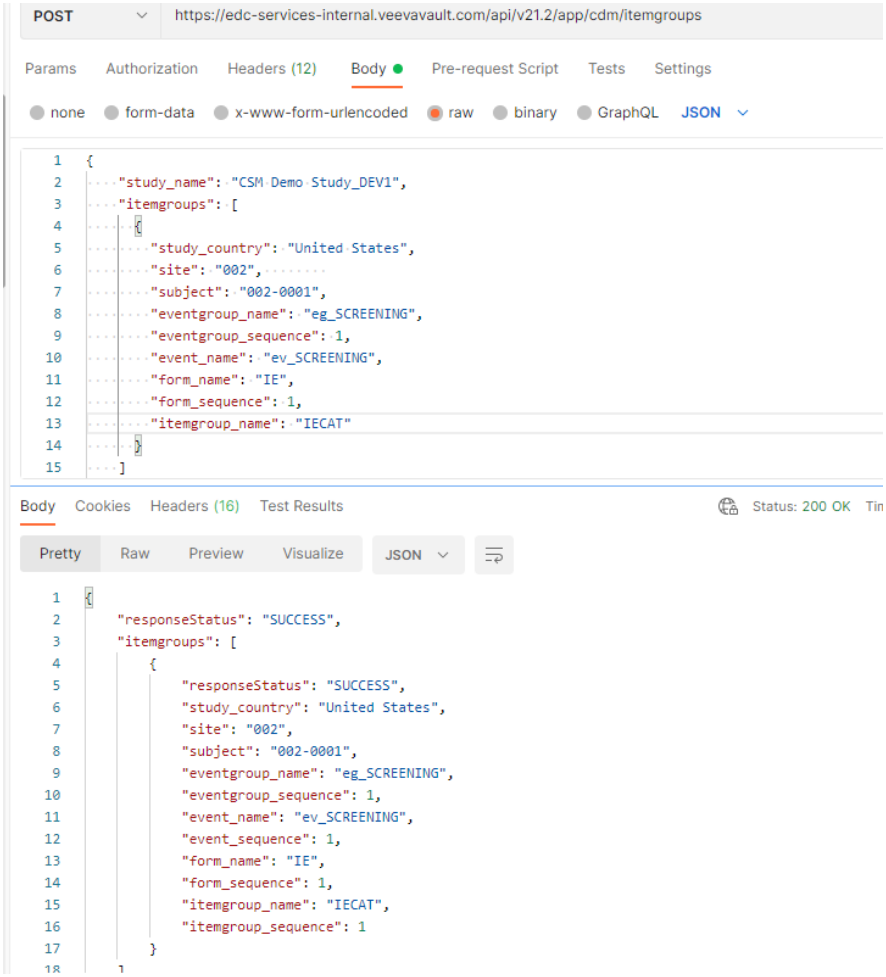
| Description / Request (URL) | Response |
|---|---|
| Add of one (next) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"itemgroups": [<br>{<br>"study_country": "United States", | {<br>"responseStatus": "SUCCESS",<br>"itemgroups": [<br>{<br>"responseStatus": "SUCCESS", |

| Description / Request (URL) | Response |
|---|---|
| <pre>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2"<br>}<br>]<br>}</pre> | <pre>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 1<br>}<br>]<br>}</pre> |
| Add of three, but maximum repeats is 2 at that location | |
| <pre>{<br>"study_name": "ABCP-2022-01_DEV1",<br>"itemgroups": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2"<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2"<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2"</pre> | <pre>{<br>"responseStatus": "SUCCESS",<br>"itemgroups": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 1<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 2<br>},<br>{</pre> |

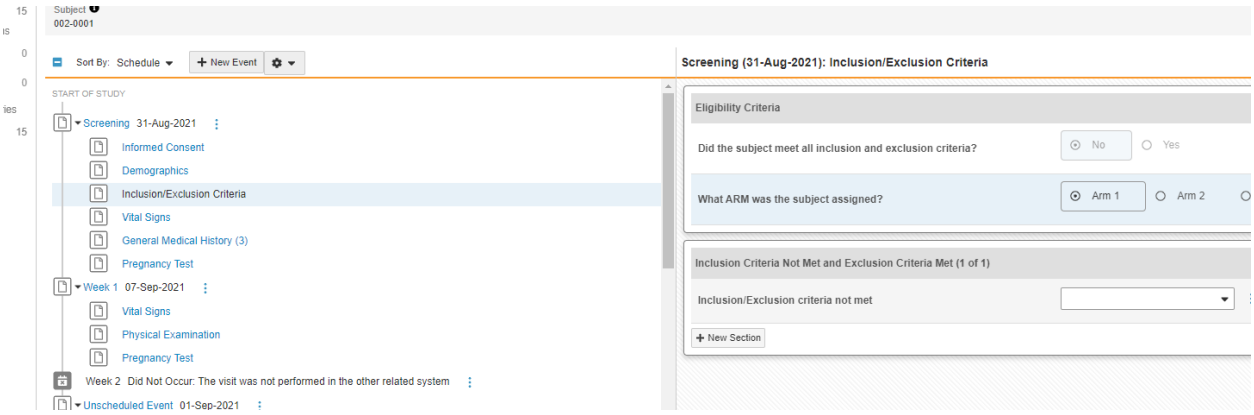
| Description / Request (URL) | Response |
| --- | --- |
| }<br>]<br>} | "responseStatus": "FAILURE",<br>"errorMessage": "Maximum number of repeating records reached",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2"<br>}<br>]<br>} |

## Flow Based Example

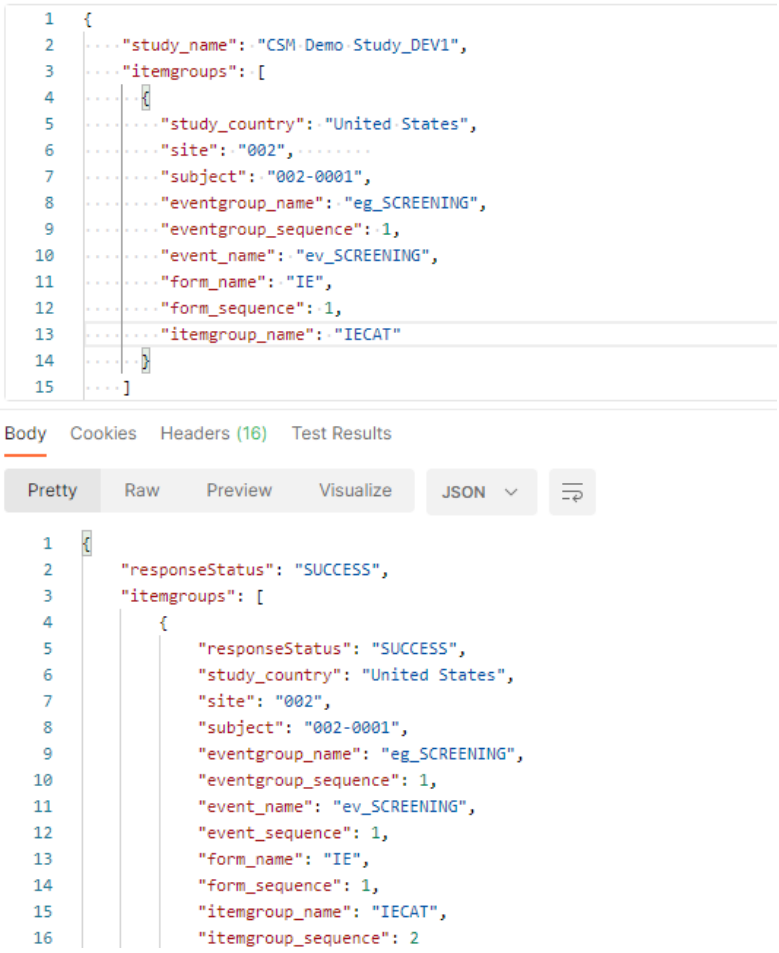
The location, design name wise (use SDS or other to get mappings) is Event Group = **eg_SCREENING**, Event = **ev_SCREENING**, Form = **IE,** Item Group = **IECAT**.  The first item group add:




… resulting in the EDC UI view:



Calling the **POST** again, a 2nd will be blindly added. Always consider inspecting what is currently on the form during an update sequence.



```
{
    "study_name": "CSM Demo Study_DEV1",
    "itemgroups": [
        {
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "form_name": "IE",
            "form_sequence": 1,
            "itemgroup_name": "IECAT"
        }
    ]
```

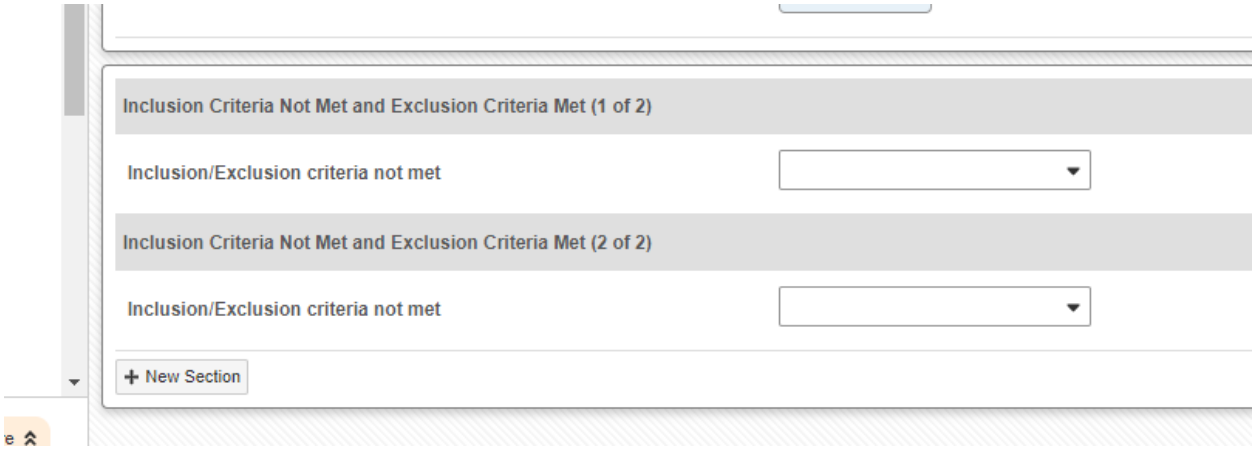
Body Cookies Headers (16) Test Results

Pretty Raw Preview Visualize JSON



```
{
    "responseStatus": "SUCCESS",
    "itemgroups": [
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "event_sequence": 1,
            "form_name": "IE",
            "form_sequence": 1,
            "itemgroup_name": "IECAT",
            "itemgroup_sequence": 2
```

… with result a 2nd item group / area on the form:



## 9.5 Upsert Item Groups

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#upsert-item-groups">https://developer-cdms.veevavault.com/api/22.2/#upsert-item-groups</a></td></tr>
<tr><td>Style</td><td>PUT</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/itemgroups</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>itemgroups</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>itemgroups</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>itemgroups</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>itemgroups</td><td>Required</td><td>Design name in the study of the event group</td></tr>
<tr><td>eventgroup_sequence</td><td>itemgroups</td><td>Optional</td><td>The specific sequence of an event group. 1 assumed if omitted</td></tr>
<tr><td>event_name</td><td>itemgroups</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>form_name</td><td>itemgroups</td><td>Required</td><td>Design name in the study of the form</td></tr>
<tr><td>form_sequence</td><td>itemgroups</td><td>Optional</td><td>The specific sequence of the form where add will occur. 1 assumed if omitted</td></tr>
<tr><td>itemgroup_name</td><td>itemgroups</td><td>Required</td><td>Design name in the study of the item group to add</td></tr>
<tr><td>itemgroup_sequence</td><td>itemgroups</td><td>Required</td><td>The specific sequence to add, or skip if it does not exist yet</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is for repeating item groups, within a form that are repeating in nature (only).</li>
<li>The endpoint is the equivalent of the EDC UI Add Section button, but for a specific sequence. The action will add the item group if it does not exist, or skip the action if it already exists.</li>
<li>The Maximum Repeats value of the study design will be enforced when attempting to add more than allowed by study design.</li>
<li>Unlike the upsert for event groups and forms, the attempt to upsert at a sequence where it would otherwise leave ‘gaps’ in the sequences, the attempt for itemgroups will instead add all necessary. Example: No item groups exist currently, and attempt uses itemgroup_sequence=3, the result will be the addition of 3 item groups, sequence 1, 2, and 3.</li>
<li>TIP: When later setting data, the Upsert Form Data (later section) will automatically add any item groups that do not exist (and are repeating). This alleviates the need to inspect and add item groups before setting specific data (i.e. need to do both)</li>
<li>Like many other API calls, batch add of many is allowed in one call. (Just more entries in the JSON array for itemgroups)</li>
</ul>
</td></tr>
</table>

### Examples

<table>
<tr><th>Description / Request (URL)</th><th>Response</th></tr>
<tr><td colspan="2">Two adds, successful</td></tr>
<tr><td>

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "itemgroups": [
            {
                "study_country": "United States",
                "site": "101",
                "subject": "SCR-0006",
                "eventgroup_name": "egSCR",
                "event_name": "evSCR",
                "form_name": "IE",
                "form_sequence": 1,
                "itemgroup_name": "ig-IE2",
                "itemgroup_sequence": 1
            },
             {
                "study_country": "United States",
                "site": "101",
                "subject": "SCR-0006",
                "eventgroup_name": "egSCR",
                "event_name": "evSCR",
                "form_name": "IE",
                "form_sequence": 1,
                "itemgroup_name": "ig-IE2",
                "itemgroup_sequence": 2
            }
    ]
}
```

</td><td>

```
{
    "responseStatus": "SUCCESS",
    "itemgroups": [
        {
            "responseStatus": "SUCCESS:CREATED",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0006",
            "eventgroup_name": "egSCR",
            "eventgroup_sequence": 1,
            "event_name": "evSCR",
            "event_sequence": 1,
            "form_name": "IE",
            "form_sequence": 1,
            "itemgroup_name": "ig-IE2",
            "itemgroup_sequence": 1
        },
        {
            "responseStatus": "SUCCESS:CREATED",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0006",
            "eventgroup_name": "egSCR",
            "eventgroup_sequence": 1,
            "event_name": "evSCR",
            "event_sequence": 1,
            "form_name": "IE",
            "form_sequence": 1,
            "itemgroup_name": "ig-IE2",
            "itemgroup_sequence": 2
        }
    ]
}
```

</td></tr>
<tr><td colspan="2">Add attempt, but the item group already exists (essentially SKIPPED)</td></tr>
<tr><td>

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "itemgroups": [
            {
                "study_country": "United States",
                "site": "101",
                "subject": "SCR-0006",
                "eventgroup_name": "egSCR",
                "event_name": "evSCR",
```

</td><td>

```
{
    "responseStatus": "SUCCESS",
    "itemgroups": [
        {
            "responseStatus": "SUCCESS:UPDATED",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0006",
            "eventgroup_name": "egSCR",
```

</td></tr>
</table>

| Description / Request (URL) | Response |
|---|---|
| <pre>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 1<br>}<br>]<br>}</pre> | <pre>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 1<br>}<br>]<br>}</pre> |
| Add of a later sequence - the result is add of all 'up to', i.e. gaps filled | |
| <pre>{<br>"study_name": "ABCP-2022-01_DEV1",<br>"itemgroups": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0007",<br>"eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 2<br>}<br>]<br>}</pre> | <pre>{<br>"responseStatus": "SUCCESS",<br>"itemgroups": [<br>{<br>"responseStatus": "SUCCESS:CREATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0006",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 1<br>},<br>{<br>"responseStatus": "SUCCESS:CREATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0006",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 2<br>}<br>]<br>}</pre> |
| Add of a later sequence - the result is add of all 'up to', i.e. gaps filled | |
| <pre>{</pre> | <pre>{</pre> |

| Description / Request (URL) | Response |
| --- | --- |
| "study_name": "ABCP-2022-01_DEV1",<br>"itemgroups": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0006",<br>"eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 5<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0006",<br>"eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 6<br>}<br>]<br>} | "responseStatus": "SUCCESS",<br>"itemgroups": [<br>{<br>"responseStatus": "SUCCESS:CREATED",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0006",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>"itemgroup_sequence": 5<br>},<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "Maximum number of repeating records reached",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0006",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE2",<br>}<br>]<br>} |

## 9.6 Set/Upsert Form Data (Items)

<table>
<tr><td>Help Main</td><td>https://developer-cdms.veevavault.com/api/22.2/#set-form-data (POST)<br>https://developer-cdms.veevavault.com/api/22.2/#upsert-form-data (PUT, to same endpoint)</td></tr>
<tr><td>Style(s)</td><td>POST (runs as add / update)<br>PUT (runs as upsert)</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/items</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>forms</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>forms</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>forms</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>forms</td><td>Required</td><td>Design name in the study of the event group</td></tr>
<tr><td>eventgroup_sequence</td><td>forms</td><td>Optional</td><td>The specific sequence of an event group. 1 assumed if omitted</td></tr>
<tr><td>event_name</td><td>forms</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>form_name</td><td>forms</td><td>Required</td><td>Design name in the study of the form</td></tr>
<tr><td>form_sequence</td><td>forms</td><td>Optional</td><td>The specific sequence of the form. 1 assumed if omitted</td></tr>
<tr><td>itemgroup_name</td><td>forms/items</td><td>Required</td><td>Design name in the study of the item group location</td></tr>
<tr><td>itemgroup_sequence</td><td>forms/items</td><td>Optional</td><td>The specific item group sequence where the item resides. 1 assumed if omitted</td></tr>
<tr><td>item_name</td><td>forms/items</td><td>Required</td><td>Design name in the study of the item to set with value</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is for setting data on fields of a form</li>
<li>The differences between POST and PUT style:
<ul>
<li>POST style requires that all item groups (i.e. if repeating) must exist first.</li>
<li>PUT style does not require the specific upsert of any necessary item groups on the form. They are automatically added where needed</li>
<li>The ‘inner’ status on the PUT style includes the secondary status of ‘:CREATED’ (item group of the item was created) or ‘:UPDATED’ (item already existed)</li>
<li>TIP: Consider always using the PUT / upsert.</li>
</ul>
</li>
<li>Note the array at top level is the forms (json array) to direct data to, multiple forms can be targeted in one API call. The 2nd level items, which are the items on that form, each of which requires itemgroup_name, item_name, value, and other parameters, based on the field type.</li>
<li>See the table below for guidelines based on EDC field type</li>
<li>WARNING: If a form is submitted state, it must first be opened for edit via a different API endpoint, Edit Form (later section), before proceeding with the update of data. Currently there is no parameter to allow for conditional open for edit, during the call to update data on a form, not a parameter to additionally submit the form once data is updated/added.</li>
<li>TIP: Although the example above is from a Screening event, in a non-repeating event group, and eventgroup_sequence is not strictly required it is good practice to always include this value. The same is true for the sequence of form, and of course itemgroup (of the items being updated)</li>
</ul>
</td></tr>
</table>

| | |
|---|---|
| | • **WARNING**: As of the 22R2 release - item groups on forms that are used in the Local Labs module are disallowed from updates through the API. |

### Setting Data by Item Type

| Item Type | Example | Notes |
|---|---|---|
| **Checkbox** | | |
| | "value": "true" | i.e. checked |
| | "value": "false" | i.e. unchecked |
| | "value": "" | Has the same effect as false - Suggestion is to use true / false always in payload. |
| | (any other value) | **Error returned** |
| **Codelist** | | |
| | "value": "Y" | I.e. assuming that codelist has a **code** value of "Y' in its |
| | "value": "" | Empty / unset the selection |
| | (any other value, i.e. not a codelist code) | **Error returned** |
| **URL** | | |
| | "value": "https://www.google.com" | Set the value to a URL, on screen user can click to navigate to the URL, in another tab |
| | "value": "" | Empty / unset the value |
| | (any value that does not start http or ftp) | **Error returned** |
| **Text Field** | | |
| | "value": "Here is my text in the field" | Set a value |
| | "value": "" | Empty / unset the value |
| | (value over max length) | **Error returned** |
| **Numeric (Integer or Decimal)** | | |
| | "value": "3" | E.g. a field with precision = 0 (no decimal portion) |
| | "value": "13.4" | E.g. a decimal portion allowed field (precision > 0) |
| | "value": "" | Empty / unset the value |
| | (not numeric or violates length or precision) | **Error returned** |
| **Date Field** | | |
| | "value": "2022-06-01" | E.g. to set June 1st, 2022 |
| | "value": "" | Empty / unset the value |
| | "value": "2022-UN-UN" | E.g. when the field allows unknowns in month and/or day. **UN** must be used in the appropriate portion |
| | (bad date, not of proper format, YYYY-MM-DD) | **Error returned** |
| **Date Time Field** | | |
| | "value": "2022-06-01T12:30:00Z" | Set the datetime in full, this will take on the 'on screen' value (not the normalized) |
| | "value": "" | Empty / unset the value |

| Item Type | Example | Notes |
|---|---|---|
| | (bad date, not of proper format, yyyy-MM-ddTHH:mmZ) | **Error returned** |
| **Time Field** | | |
| | "value": "12:30" | Set the time value |
| | "value": "" | Empty / unset the value |
| | (bad date, not of proper format, HH:mm) | **Error returned** |
| **Unit Codelist** | | |
| | "value": "62.2"<br>"unit_value": "in" | E.g. 'in' is a valid **code** in the unit codelist definition. |
| | "value": ""<br>"unit_value": "" | Empty / unset the value |
| | "value": "62.2" | **Error returned (must have both)** |
| | (any violation of length, precision, or invalid unit code) | **Error returned** |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Set of data on a form with one of each field/item type | |


```
{
    "study_name": "ABCP-2022-01_DEV1",
    "forms": [
            {
                "site": "101",
                "study_country": "United States",
                "subject": "SCR-0001",
                "eventgroup_name": "egUNS",
                "eventgroup_sequence": 3,
                "event_name": "evUNS",
                "form_name": "OE",
                "form_sequence": 1,
                "items": [
                    {
                    "itemgroup_name": "igOE",
                    "itemgroup_sequence": 1,
                    "item_name": "CB",
                     "value": "true"
                    },
                    {
                    "itemgroup_name": "igOE",
                    "itemgroup_sequence": 1,
                    "item_name": "CL",
```

```
{
    "responseStatus": "SUCCESS",
    "items": [
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0001",
            "eventgroup_name": "egUNS",
            "eventgroup_sequence": 3,
            "event_name": "evUNS",
            "event_sequence": 1,
            "form_name": "OE",
            "form_sequence": 1,
            "itemgroup_name": "igOE",
            "itemgroup_sequence": 1,
            "item_name": "CB",
            "value": "true"
         },
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
```

| Description / Request (URL) | Response |
| --- | --- |
| "value": "Y"<br>},<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "DATEONLY",<br>"value": "2022-06-01"<br>},<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "DATEONLY_UNKS",<br>"value": "2022-UN-UN"<br>},<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "DATEANDTIME",<br>"value": "2022-06-01T12:30Z"<br>},<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "DT_WITH_UNKS",<br>"value": "2022-06-UNTUN:UNZ"<br>},<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "INTFIELD",<br>"value": "3"<br>},<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "DECIMAL",<br>"value": "5.2"<br>},<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "UCLFIELD",<br>"value": "2.1",<br>"unit_value": "double_std"<br>},<br>{<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1, | "subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 3,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "CL",<br>"value": "Y"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 3,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "DATEONLY",<br>"value": "2022-06-01"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 3,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "DATEONLY_UNKS",<br>"value": "2022-UN-UN"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101", |

| Description / Request (URL) | Response |
|---|---|

```
                    "item_name": "URL_FIELD",
                     "value": "https://www.google.com"
                    },
                    {
                    "itemgroup_name": "igOE",
                    "itemgroup_sequence": 1,
                    "item_name": "SHORTTXT",
                     "value": "Some text short"
                    },
                    {
                    "itemgroup_name": "igOE",
                    "itemgroup_sequence": 1,
                    "item_name": "LONGTXT",
                    "value": "Some text in a long text
field \n with a line break in the payload"
                    }
                ]
            }
        ]
}
```

```
            "subject": "SCR-0001",
            "eventgroup_name": "egUNS",
            "eventgroup_sequence": 3,
            "event_name": "evUNS",
            "event_sequence": 1,
            "form_name": "OE",
            "form_sequence": 1,
            "itemgroup_name": "igOE",
            "itemgroup_sequence": 1,
            "item_name": "DATEANDTIME",
            "value": 2022-06-01T12:30Z"
        },
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0001",
            "eventgroup_name": "egUNS",
            "eventgroup_sequence": 3,
            "event_name": "evUNS",
            "event_sequence": 1,
            "form_name": "OE",
            "form_sequence": 1,
            "itemgroup_name": "igOE",
            "itemgroup_sequence": 1,
            "item_name": "DT_WITH_UNKS",
            "value": "2022-06-UNTUN:UNZ"
        },
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0001",
            "eventgroup_name": "egUNS",
            "eventgroup_sequence": 3,
            "event_name": "evUNS",
            "event_sequence": 1,
            "form_name": "OE",
            "form_sequence": 1,
            "itemgroup_name": "igOE",
            "itemgroup_sequence": 1,
            "item_name": "INTFIELD",
            "value": "3"
        },
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
```

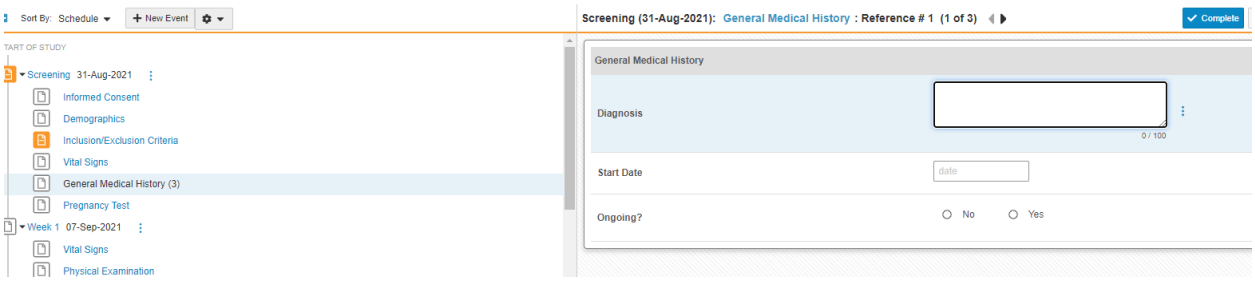
| Description / Request (URL) | Response |
| --- | --- |
| | "subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 3,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "DECIMAL",<br>"value": "5.2"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 3,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "UCLFIELD",<br>"value": "2.1",<br>"unit_value": "double_std"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 3,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "URL_FIELD",<br>"value": "https://www.google.com"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States", |

| Description / Request (URL) | Response |
| --- | --- |
| | "site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 3,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "SHORTTXT",<br>"value": "Some text short"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "egUNS",<br>"eventgroup_sequence": 3,<br>"event_name": "evUNS",<br>"event_sequence": 1,<br>"form_name": "OE",<br>"form_sequence": 1,<br>"itemgroup_name": "igOE",<br>"itemgroup_sequence": 1,<br>"item_name": "LONGTXT",<br>"value": "Some text in a long text field \n with a line break in the payload"<br>}<br>]<br>} |
| Change of an item (i.e. reopened form, then update) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"forms": [<br>{<br>"site": "101",<br>"study_country": "United States",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"form_name": "DM",<br>"form_sequence": 1,<br>"items": [<br>{ | {<br>"responseStatus": "SUCCESS",<br>"items": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "DM",<br>"form_sequence": 1, |

| Description / Request (URL) | Response |
|---|---|
| <pre>"itemgroup_name": "ig-DM",<br>"itemgroup_sequence": 1,<br>"item_name": "BRTHDAT",<br>"value": "1981-03-03",<br>"change_reason": "Updated by integration"<br>}<br>]<br>}<br>]<br>}</pre> | <pre>"itemgroup_name": "ig-DM",<br>"itemgroup_sequence": 1,<br>"item_name": "BRTHDAT",<br>"value": "1981-03-03",<br>"change_reason": "Updated by integration"<br>}<br>]<br>}</pre> |
| PUT style (upsert) | |
| Other examples apply, except the response status indicates if the item was created (in an item group that was also created, or updated if it already exists). 'SUCCESS:UPDATED' is also returned if no changes were made. | <pre>{<br>"responseStatus": "SUCCESS",<br>"items": [<br>{<br>"responseStatus": "SUCCESS:CREATED",<br>:<br>:<br><br>OR<br><br>{<br>"responseStatus": "SUCCESS",<br>"items": [<br>{<br>"responseStatus": "SUCCESS:UPDATED",<br>:<br>:</pre> |

## Flow Based Examples

Consider the **General Medical History** form of **Screening** visit of the example study:




… it has 3 items on it. It is a repeating form, so **form_sequence** value in the API call is vital. There are no repeating item groups on the form, so **itemgroup_sequence** for each of the items is '**1**' (or can be omitted from the request)

The API body setting 2 of the 3 items:

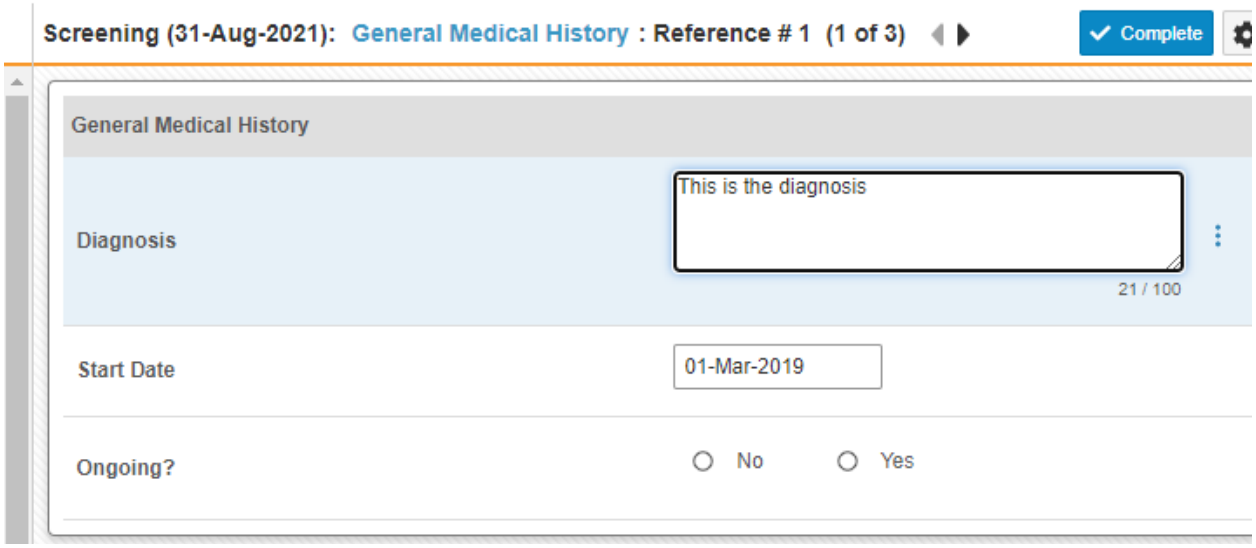
POST https://edc-services-internal.veevavault.com/api/v21.2/app/cdm/items

Params Authorization Headers (12) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "study_name": "CSM Demo Study_DEV1",
    "forms": [
        {
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "form_name": "MH",
            "form_sequence": 1,
            "items": [
                {
                    "itemgroup_name": "MH",
                    "itemgroup_sequence": 1,
                    "item_name": "MHTERM",
                    "value": "This is the diagnosis"
                },
                {
                    "itemgroup_name": "MH",
                    "itemgroup_sequence": 1,
                    "item_name": "MHSTDAT",
                    "value": "2019-03-01"
```

Yields, in EDC UI:

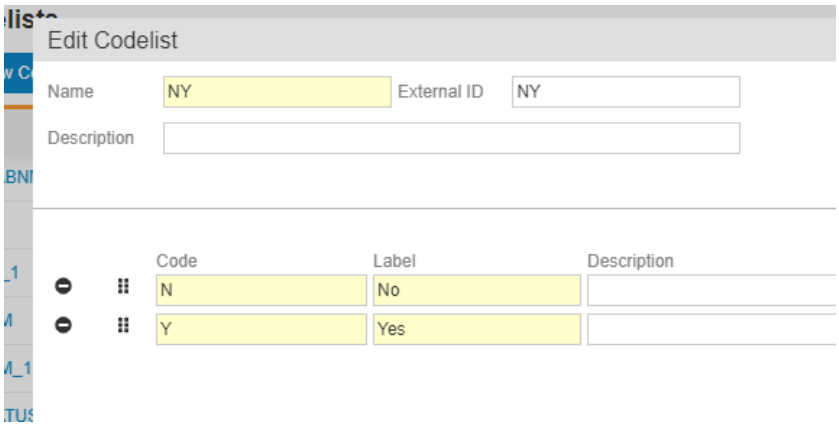



The form is still in edit mode, to submit a form the Submit Form (next section) is called afterwards. There is not currently a parameter on the set items endpoint to also submit the form, although combination APIs are part of coming releases.

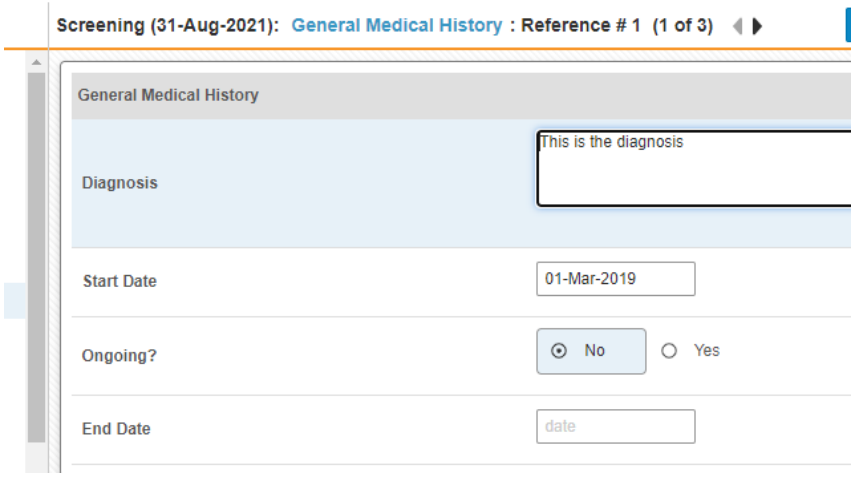
A 2nd API submit to the form, to set **Ongoing** = **No**, uses this additional section of the items JSON array:

```
            "value": "2019-03-01"
        },
        {
            "itemgroup_name": "MH",
            "itemgroup_sequence": 1,
            "item_name": "MHONGO",
            "value": "N"
        }
```

Importantly, the **N** value is from a codelist code of a study design, which can be found in the study SDS, or from Studio:



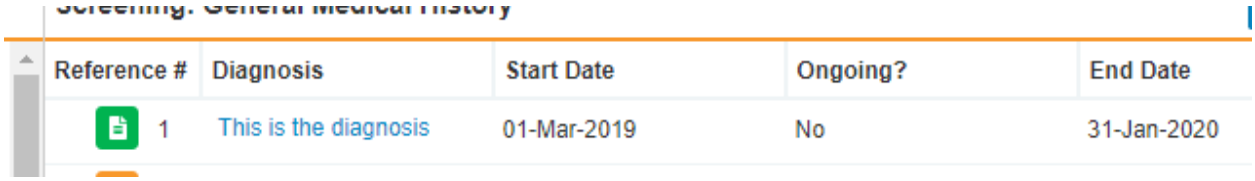
**Dynamic** fields are often part of form design, in this case, the set of Ongoing = **No**, unhides in the EDC UI a 4th field, **End Date**:




Although this form has been set with multiple API endpoint calls, it can be set – all 4 fields – in one endpoint call, even for fields that are hidden from view:


```
"items": [
    {
        "itemgroup_name": "MH",
        "itemgroup_sequence": 1,
        "item_name": "MHTERM",
        "value": "This is the diagnosis"
    },
    {
        "itemgroup_name": "MH",
        "itemgroup_sequence": 1,
        "item_name": "MHSTDAT",
        "value": "2019-03-01"
    },
    {
        "itemgroup_name": "MH",
        "itemgroup_sequence": 1,
        "item_name": "MHONGO",
        "value": "N"
    },
    {
        "itemgroup_name": "MH",
        "itemgroup_sequence": 1,
        "item_name": "MHENDAT",
        "value": "2020-01-31"
    }
]
}
```


**IMPORTANT:** Always be reticent of any dynamic fields on the form. The set of Ongoing = **Y** (Yes) would be allowed, with an end date via the API endpoint, even though the data would not make sense per the form's dynamic rules. For submitted forms (different API, next section), like:



| Reference # | Diagnosis | Start Date | Ongoing? | End Date |
|---|---|---|---|---|
| 1 | This is the diagnosis | 01-Mar-2019 | No | 31-Jan-2020 |

… a reason for change will be required for any field update, after the first submit of the form. Additionally, the form won't allow edits when in submitted state. So, this attempt to update the **Diagnosis** field:

```
{
    "study_name": "CSM Demo Study_DEV1",
    "forms": [
        {
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "form_name": "MH",
            "form_sequence": 1,
            "items": [
                {
                    "itemgroup_name": "MH",
                    "itemgroup_sequence": 1,
                    "item_name": "MHTERM",
                    "value": "Headaches",
                    "change_reason": "This is a custom reason for field value"
                }
            ]
        }
    ]
}
```

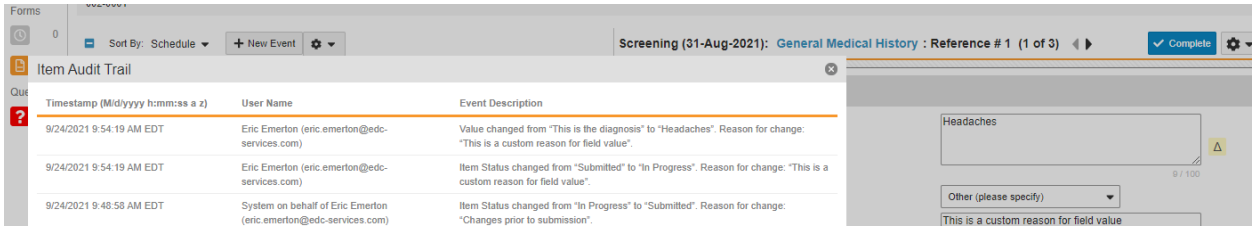
Is met with:

```
{
    "responseStatus": "SUCCESS",
    "items": [
        {
            "responseStatus": "FAILURE",
            "errorMessage": "Items on submitted forms cannot be edited",
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "event_sequence": 1,
            "form_name": "MH",
            "form_sequence": 1,
            "itemgroup_name": "MH",
```

With the form opened for edit (its API endpoint is later section), notice the **change_reason** parameter, which can be any text necessary to the update:

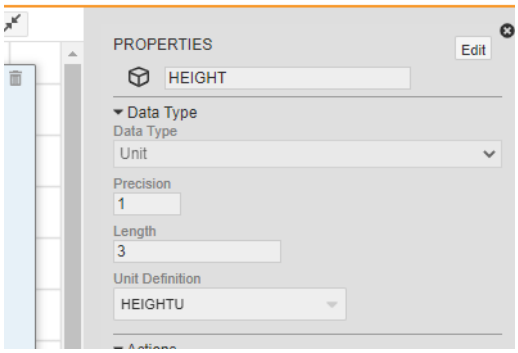
```
16                "itemGroup_sequence": 1,
17                "item_name": "MHTERM",
18                "value": "Headaches",
19                "change_reason": "This is a custom reason for field value"
20            }
21        ]
```

.. is shown in the UI reason in the audit trail:

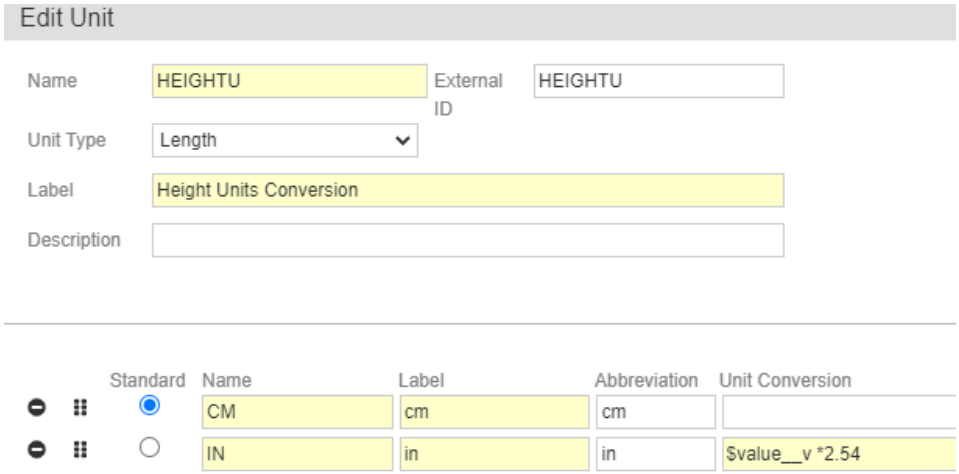


Next, consider the Vitals form, with unit codelist fields for Height, Weight, others:

The Height field definition:



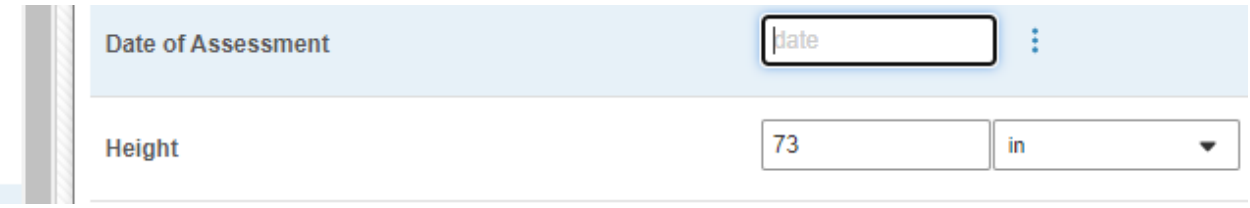

.. with unit codelist HEIGHTU definition prescribing codes of only **CM** and **IN**:



The API body to update the Height field, then, to set 73 Inches is:

```
{
    "study_name": "CSM Demo Study_DEV1",
    "forms": [
        {
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "form_name": "VS",
            "form_sequence": 1,
            "items": [
                {
                    "itemgroup_name": "VS",
                    "itemgroup_sequence": 1,
                    "item_name": "HEIGHT",
                    "value": "73",
                    "unit_value": "IN"
                }
            ]
        }
```

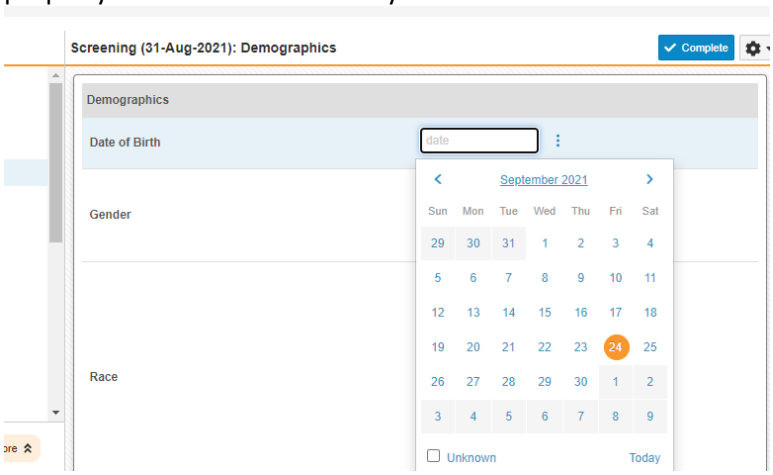
Yielding the EDC UI set as:




Although the definition of the field is 3 places for the whole number portion, and 1 for decimal, it is not required to enter '73.0' through the API. Attempting 73.12, though, violates the design of this field, and is met with this on attempt:

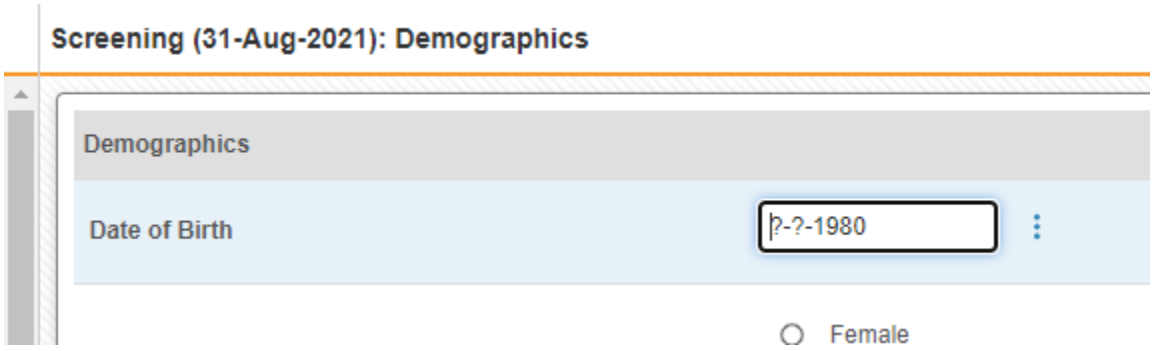
```
{
    "responseStatus": "SUCCESS",
    "items": [
        {
            "responseStatus": "FAILURE",
            "errorMessage": "This field only accepts 3 digit(s) before the decimal and 1 digit(s) after the decimal.",
            "study_country": "United States",
            "site": "002",
```

Next in consideration is fields that allow for unknown parts (dates / datetimes). The Date of Birth field is set with the property for both month and day:

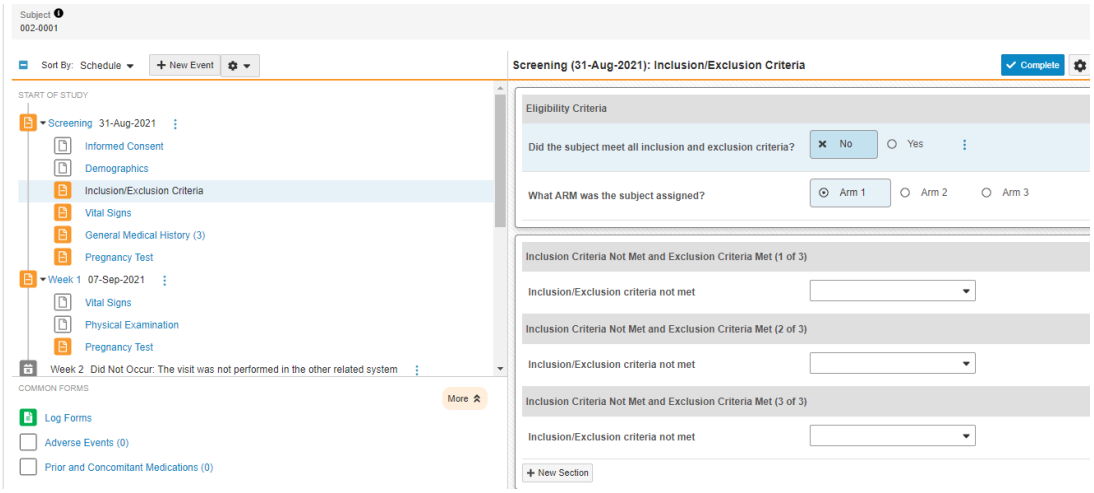



The API body to set an unknown value for both Month and Day: (**UN** is used in each of the portions)

.. which yields in EDC the UI formatted value of **?** in the unknown parts. (**?** Is NOT used in the API call)



Next in consideration is forms with repeating item groups. This form in the example study has a non-repeating item group of summary information, then a repeating item group for criteria not met:

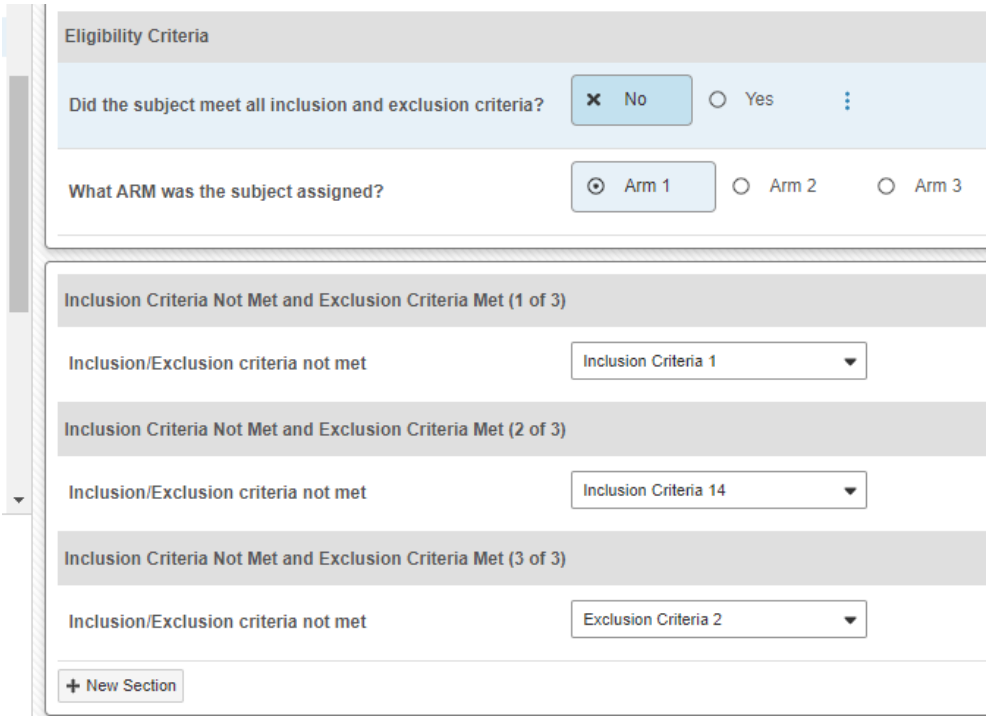



The pictured form has 3 item groups added, which can be done in separate API calls (see Add Item Groups or Upsert Item Groups sections). **Alternatively**, the Upsert Form Data (**PUT** used) will **automatically** run the upsert of item groups, adding any that are necessary for the data update.

This API body uses codelist codes for values, and identifies the item group by sequence for each value:

```
    {
        "study_country": "United States",
        "site": "002",
        "subject": "002-0001",
        "eventgroup_name": "eg_SCREENING",
        "eventgroup_sequence": 1,
        "event_name": "ev_SCREENING",
        "form_name": "IE",
        "form_sequence": 1,
        "items": [
            {
                "itemgroup_name": "IECAT",
                "itemgroup_sequence": 1,
                "item_name": "IETESTCD1",
                "value": "INC1"
            },
            {
                "itemgroup_name": "IECAT",
                "itemgroup_sequence": 2,
                "item_name": "IETESTCD1",
                "value": "INC14"
            },
            {
                "itemgroup_name": "IECAT",
                "itemgroup_sequence": 3,
                "item_name": "IETESTCD1",
                "value": "EXC2"
            }
        ]
    }
]
```

.. which yields the EDC UI result:


Eligibility Criteria
Did the subject meet all inclusion and exclusion criteria?
No
Yes
What ARM was the subject assigned?
Arm 1
Arm 2
Arm 3
Inclusion Criteria Not Met and Exclusion Criteria Met (1 of 3)
Inclusion/Exclusion criteria not met
Inclusion Criteria 1
Inclusion Criteria Not Met and Exclusion Criteria Met (2 of 3)
Inclusion/Exclusion criteria not met
Inclusion Criteria 14
Inclusion Criteria Not Met and Exclusion Criteria Met (3 of 3)
Inclusion/Exclusion criteria not met
Exclusion Criteria 2
New Section

## 9.7 Submit Forms

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#submit-forms |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/forms/actions/submit |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| study_country | forms | Required | Name of the study country (of the subject's site) |
| site | forms | Required | Name/number of the site |
| subject | forms | Required | Subject to act on |
| eventgroup_name | forms | Required | Design **name** in the study of the event group |
| eventgroup_sequence | forms | Optional | The specific sequence of an event group. 1 assumed if omitted |
| event_name | forms | Required | Design **name** in the study of the event |
| form_name | forms | Required | Design **name** in the study of the form to add |
| form_sequence | forms | Required | The specific sequence to submit |

| | |
|---|---|
| Notes | • The action is for submitting a form, the equal of clicking Complete in the UI<br>• This action is important in that it will run all study design rules of that form, which can/might include dynamic add of other events/forms in the study. |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Submit of one form | |

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "forms": [
            {
                "site": "101",
                "study_country": "United States",
                "subject": "SCR-0002",
                "eventgroup_name": "egSCR",
                "event_name": "evSCR",
                "form_name": "IE"
            }
        ]
}
```

```
{
    "responseStatus": "SUCCESS",
    "forms": [
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0002",
            "eventgroup_name": "egSCR",
            "eventgroup_sequence": 1,
            "event_name": "evSCR",
            "event_sequence": 1,
            "form_name": "IE",
            "form_sequence": 1
        }
    ]
```

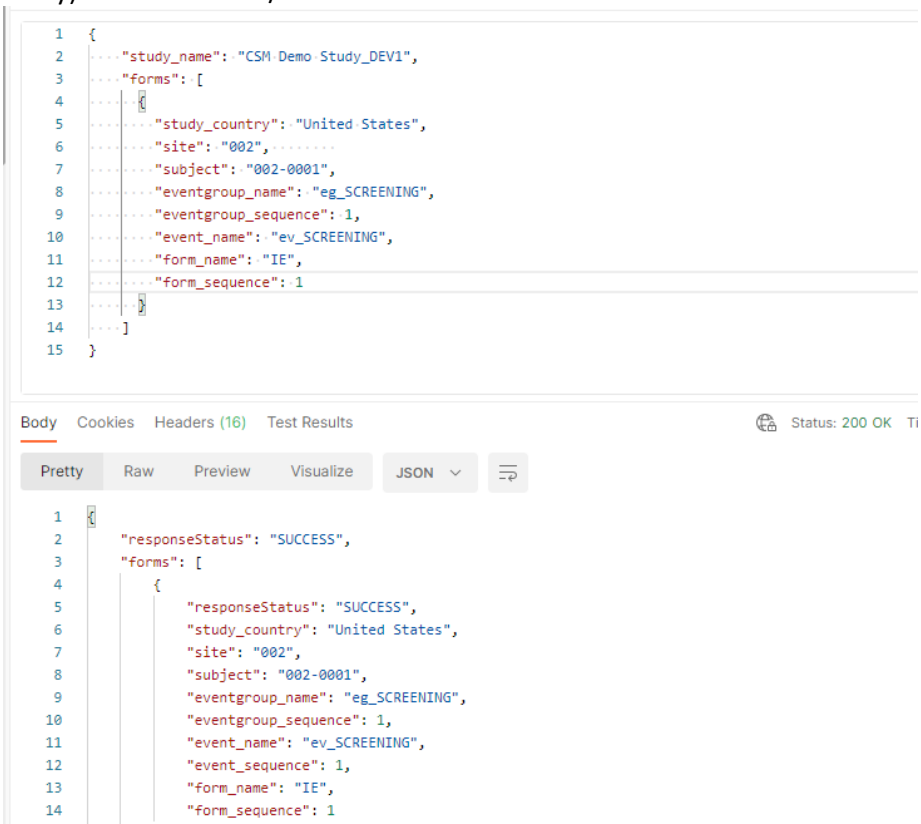
| Description / Request (URL) | Response |
|---|---|
| | } |
| Submit of specific repeating form | |
| { "study_name": "ABCP-2022-01_DEV1", "forms": [ { "site": "101", "study_country": "United States", "subject": "SCR-0002", "eventgroup_name": "LOGS", "event_name": "LOGS", "form_name": "AE", "form_sequence": 3 } ] } | { "responseStatus": "SUCCESS", "forms": [ { "responseStatus": "SUCCESS", "study_country": "United States", "site": "101", "subject": "SCR-0002", "eventgroup_name": "LOGS", "eventgroup_sequence": 1, "event_name": "LOGS", "event_sequence": 1, "form_name": "AE", "form_sequence": 3 } ] } |
| Multiple attempted, one succeeds, one fails | |
| { "study_name": "ABCP-2022-01_DEV1", "forms": [ { "site": "101", "study_country": "United States", "subject": "SCR-0002", "eventgroup_name": "LOGS", "event_name": "LOGS", "form_name": "AE", "form_sequence": 3 }, { "site": "101", "study_country": "United States", "subject": "SCR-0002", "eventgroup_name": "LOGS", "event_name": "LOGS", "form_name": "AE", "form_sequence": 4 } ] } | { "responseStatus": "SUCCESS", "forms": [ { "responseStatus": "SUCCESS", "study_country": "United States", "site": "101", "subject": "SCR-0002", "eventgroup_name": "LOGS", "eventgroup_sequence": 1, "event_name": "LOGS", "event_sequence": 1, "form_name": "AE", "form_sequence": 3 }, { "responseStatus": "FAILUIRE", "errorMessage": "Form is submitted", "study_country": "United States", "site": "101", "subject": "SCR-0002", "eventgroup_name": "LOGS", "eventgroup_sequence": 1, "event_name": "LOGS", "event_sequence": 1, "form_name": "AE", "form_sequence": 4 } |

| Description / Request (URL) | Response |
|---|---|
| | ]<br>} |

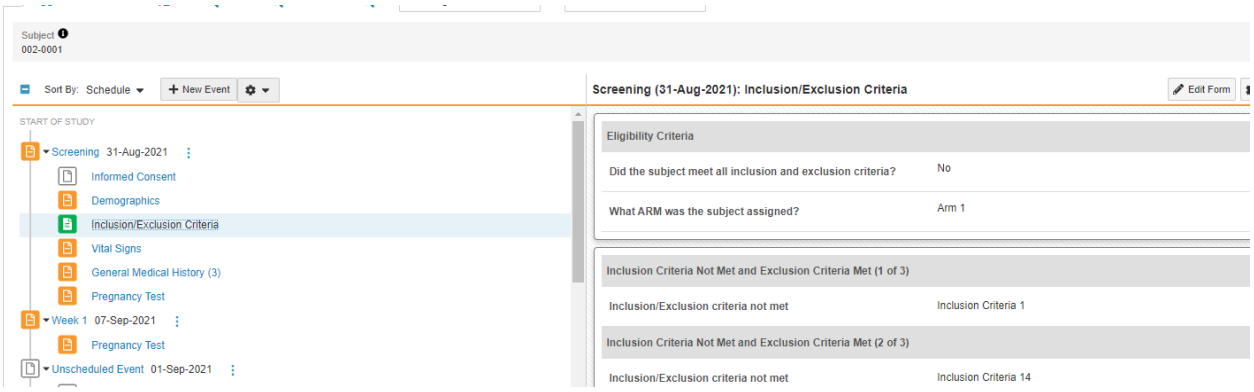
### Flow Based Examples

For an unsubmitted form the EDC UI:

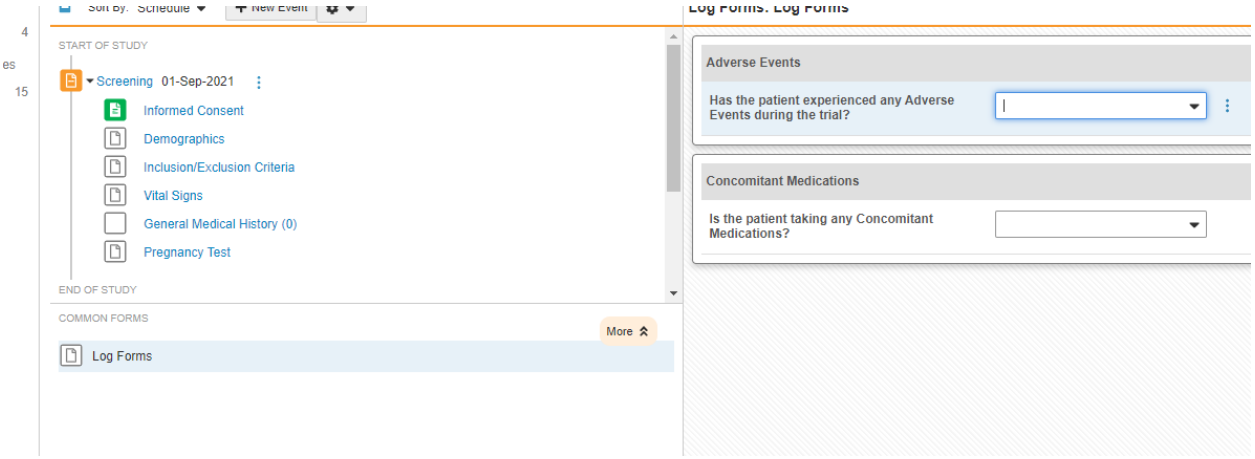
The post to submit the form will run study design rules (potentially adding dynamic events/forms elsewhere in the study). The Inclusion / Exclusion is submitted with:



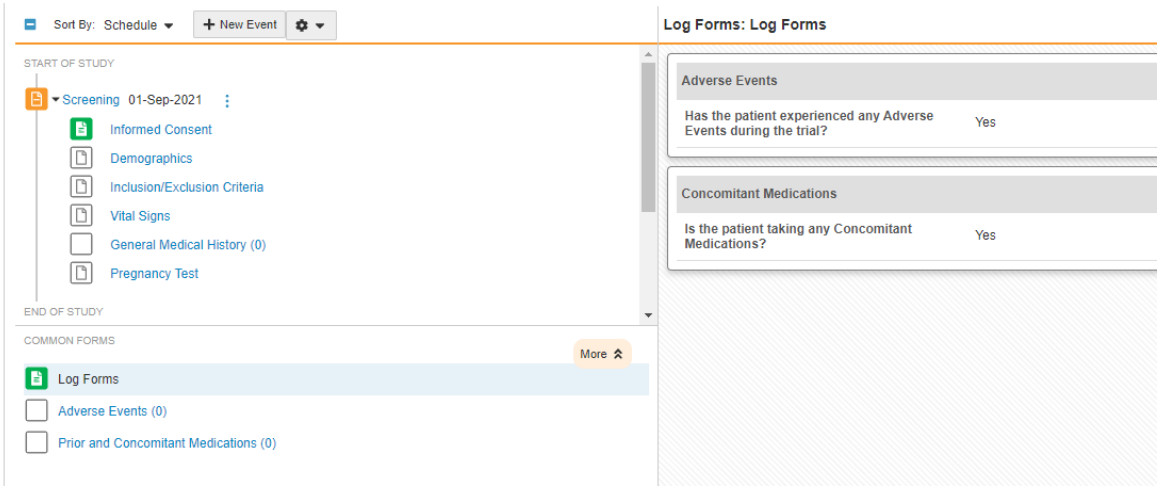

.. yielding the form in EDC UI:




**IMPORTANT:** When forms appear dynamically, based on answers of other forms, those forms must be submitted to run the rules that drive the appearance of the form. This example in EDC:




.. means the Adverse Events or Prior and Concomitant Medication forms (of this study design) are not available for Add Form (repeating form) or Set Form Data **yet**. Only on using the endpoint to set data to the Log Forms questions above, then submit of that form will yield the AE and CM form areas:



## 9.8 Reopen Submitted Forms

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#reopen-submitted-forms |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/forms/actions/edit |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| study_country | forms | Required | Name of the study country (of the subject's site) |
| site | forms | Required | Name/number of the site |
| subject | forms | Required | Subject to act on |
| eventgroup_name | forms | Required | Design **name** in the study of the event group |
| eventgroup_sequence | forms | Optional | The specific sequence of an event group. 1 assumed if omitted |
| event_name | forms | Required | Design **name** in the study of the event |
| form_name | forms | Required | Design **name** in the study of the form to add |
| form_sequence | forms | Optional | The specific sequence to open. 1 assumed if omitted |
| change_reason | forms | Required | The reason to be applied as part of opening the form for edit |

**Notes**

- The action is for opening a form back up, i.e., to further edit data on the form.
- Currently there is no parameter on the set items (updating a form) to also open the form conditionally for edit (should it be in submitted state). Inspection of the current state of the form is suggested. A coming release will include combination API(s) that do multiple actions, i.e. open a form (if it needs opening), setting data, submitting, as one API call.
- The parameter **change_reason** in the body is required for opening a form and will be added to the audit trail of the form.
- **WARNING:** Any form in **frozen** or **locked** status **cannot** be opened for edit via the API. There is no current API to unfreeze or unlock. These actions were human based decisions, and CDMS stance is that one should not be able to override that via an API call

### Examples

| Description / Request (URL) | Response |
|---|---|
| Open one form for edit | |

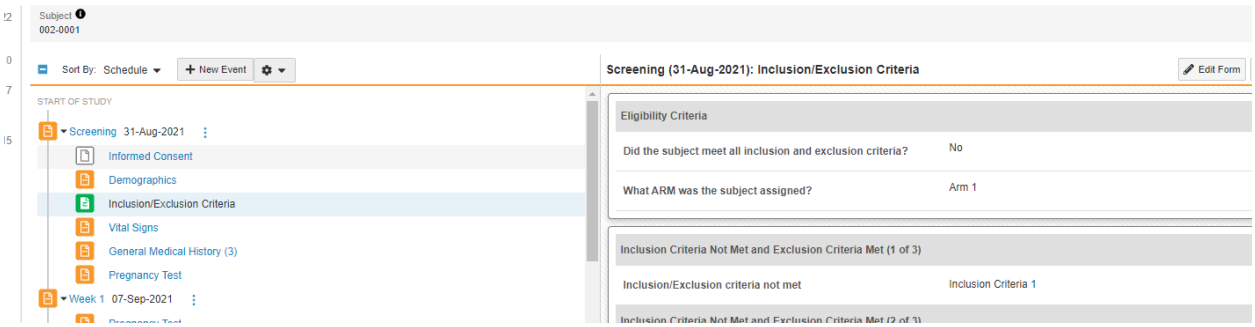
```
{
    "study_name": "ABCP-2022-01_DEV1",
    "forms": [
            {
                "site": "101",
                "study_country": "United States",
                "subject": "SCR-0002",
                "eventgroup_name": "LOGS",
```

```
{
    "responseStatus": "SUCCESS",
    "forms": [
        {
            "responseStatus": "SUCCESS",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0002",
```

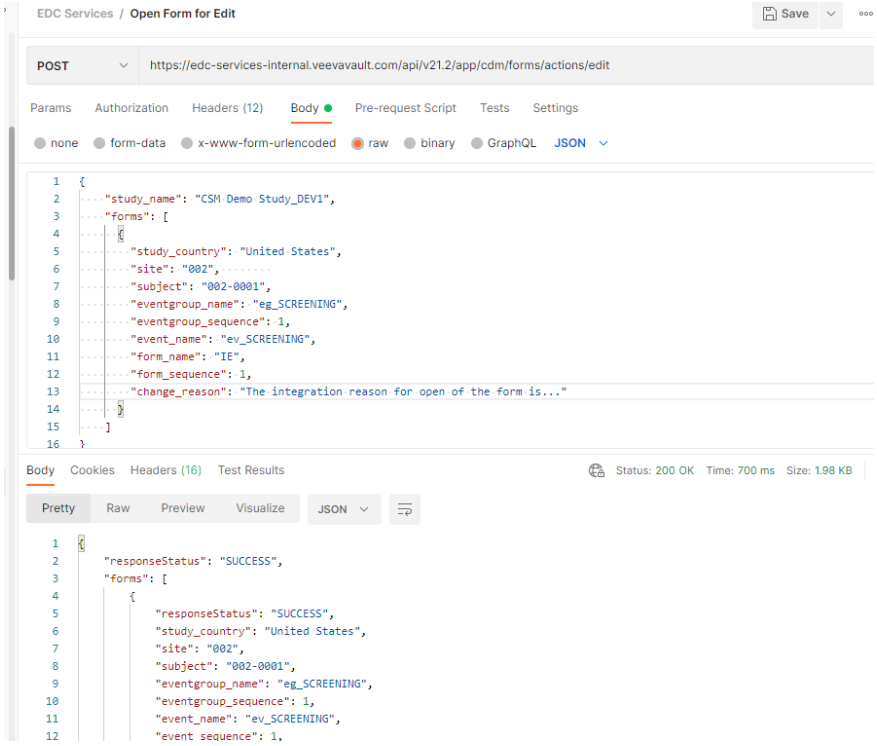
| Description / Request (URL) | Response |
|---|---|
| <pre>"event_name": "LOGS",<br>"form_name": "AE",<br>"form_sequence": 3,<br>"change_reason": "my reason for open form"<br>}<br>]<br>}</pre> | <pre>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 3<br>}<br>]<br>}</pre> |
| Example 2 | |
| <pre>{<br>"study_name": "ABCP-2022-01_DEV1",<br>"forms": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE",<br>"form_sequence": 3,<br>"change_reason": "my reason for open form"<br>},<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"event_name": "LOGS",<br>"form_name": "AE",<br>"form_sequence": 4,<br>"change_reason": "my reason for open form"<br>}<br>]<br>}</pre> | <pre>{<br>"responseStatus": "SUCCESS",<br>"forms": [<br>{<br>"responseStatus": "SUCCESS",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 3<br>},<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "Form is not submitted",<br>"site": "101",<br>"study_country": "United States",<br>"subject": "SCR-0002",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 4<br>}<br>]<br>}</pre> |

## Flow Based Example

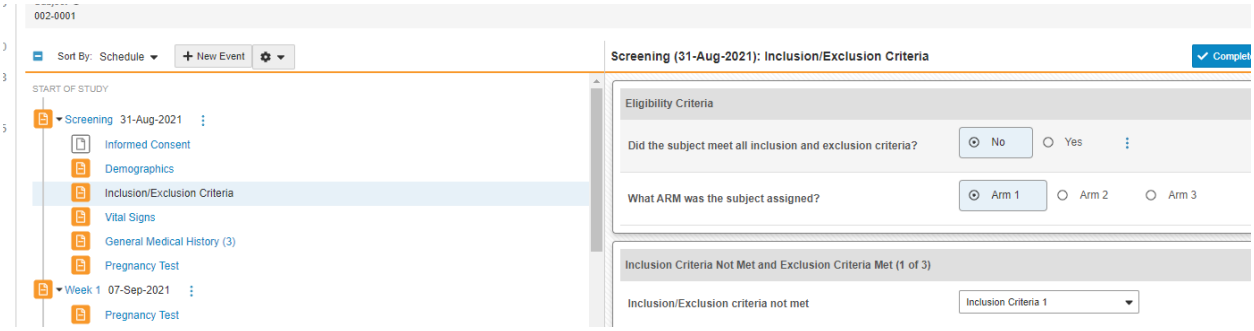
For the currently submitted form in EDC:

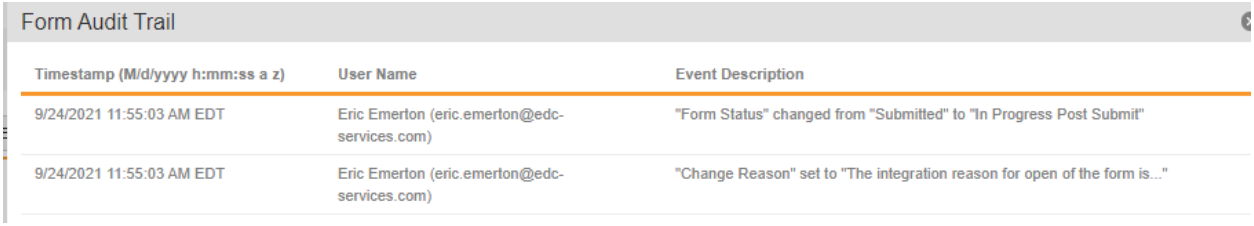



The API post:




.. opens the form in the EDC UI, with the reason provided in the audit trail at the form level:


002-0001
Sort By: Schedule
New Event
START OF STUDY
Screening 31-Aug-2021
Informed Consent
Demographics
Inclusion/Exclusion Criteria
Vital Signs
General Medical History (3)
Pregnancy Test
Week 1 07-Sep-2021
Pregnancy Test
Screening (31-Aug-2021): Inclusion/Exclusion Criteria
Eligibility Criteria
Did the subject meet all inclusion and exclusion criteria?
No
Yes
What ARM was the subject assigned?
Arm 1
Arm 2
Arm 3
Inclusion Criteria Not Met and Exclusion Criteria Met (1 of 3)
Inclusion/Exclusion criteria not met
Inclusion Criteria 1


Form Audit Trail
Timestamp (M/d/yyyy h:mm:ss a z)
User Name
Event Description
9/24/2021 11:55:03 AM EDT
Eric Emerton (eric.emerton@edc-services.com)
"Form Status" changed from "Submitted" to "In Progress Post Submit"
9/24/2021 11:55:03 AM EDT
Eric Emerton (eric.emerton@edc-services.com)
"Change Reason" set to "The integration reason for open of the form is..."

# 10 Queries

## 10.1 Retrieve Queries - By Filters

<table>
<tr><td>Help Main</td><td>https://developer-cdms.veevavault.com/api/22.2/#retrieve-queries</td></tr>
<tr><td>Style</td><td>GET</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/queries</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>form_name</td><td>Optional</td><td>Design name in the study of form to filter return of queries, i.e. queries that reside on that form only</td></tr>
<tr><td>query_status</td><td>Optional</td><td>Filter for the current status of the queries. The vault picklist name is used - one of:<br>open__v<br>answered__v<br>reopened__v<br>closed__v</td></tr>
<tr><td>event_name</td><td>Optional</td><td>Design name in the study of the event to filter to</td></tr>
<tr><td>last_modified_date</td><td></td><td>Filter to subject’s modified since a certain date. Format: yyyy-MM-ddTHH:mm:ssZ</td></tr>
<tr><td>limit</td><td>Optional</td><td>Pagination API - different limit from the default 1000 can be used</td></tr>
<tr><td>offset</td><td>Optional</td><td>Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Use this API to retrieve the existing queries of a study</li>
<li>Return specific subsets based on various filtering options.</li>
<li>If filtered by site / subject, the ‘level above’ must also be included. Example: if filtering by a site, it’s study country must also be included</li>
<li>The return value manual is such that false = added by the system.</li>
<li>When the query was added via a specific design rule, the rule_definition indicates the name of the rule that originated the query.</li>
<li>Queries are added at the item or event level. For event level, the form ‘down’ information is of course omitted</li>
<li>The return includes a messages array with each comment/message appended to the query. Reading top down in that array also reveals the progression of query status, and by whom</li>
<li>With known query ID(s) - perhaps stored as the API added queries - use of the Retrieve Queries - By ID endpoint is more straightforward.</li>
</ul>
</td></tr>
</table>

## Examples

| Description / Request (URL) | Response |
|---|---|
| Get of all queries in a study | |
| .../app/cdm/queries?study_name=ABCP-2022-01_DEV1<br><br>Return includes any status, those fired by system (with rule_definition), or added manual, query on an item, query on an event date, etc. | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 5,<br>"total": 5<br>},<br>"queries": [<br>{<br>"id": "OPW00000000T001",<br>"query_name": "VV-000230",<br>"manual": false,<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IC",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IC2",<br>"itemgroup_sequence": 1,<br>"item_name": "RFICDAT",<br>"rule_definition": "IC_001",<br>"created_date": "2022-05-23T17:38:15Z",<br>"created_by": "System",<br>"messages": [<br>{<br>"id": "OPY00000000U001",<br>"activity": "open__v",<br>"message": "Informed consent date is not equal to screening visit date. Please review.",<br>"message_date": "2022-05-23T17:38:15Z",<br>"message_by": "System"<br>}<br>]<br>},<br>{<br>"id": "OPW00000000U001",<br>"query_name": "VV-000231",<br>"manual": true, |

| Description / Request (URL) | Response |
| --- | --- |


```
            "query_status": "closed__v",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0006",
            "eventgroup_name": "egSCR",
            "eventgroup_sequence": 1,
            "event_name": "evSCR",
            "event_sequence": 1,
            "created_date": "2022-06-08T20:05:21Z",
            "created_by": "Eric Emerton",
            "messages": [
                {
                    "id": "OPY00000000V001",
                    "activity": "open__v",
                    "message": "Here is query at the event level",
                    "message_date": "2022-06-08T20:05:22Z",
                    "message_by": "Eric Emerton"
                },
                {
                    "id": "OPY00000000V004",
                    "activity": "closed__v",
                    "message": null,
                    "message_date": "2022-06-08T21:22:34Z",
                    "message_by": "Eric Emerton"
                }
            ]
        },
        {
            "id": "OPW00000000U002",
            "query_name": "VV-000232",
            "manual": true,
            "query_status": "closed__v",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0006",
            "eventgroup_name": "egSCR",
            "eventgroup_sequence": 1,
            "event_name": "evSCR",
            "event_sequence": 1,
            "form_name": "Vital-Signs",
            "form_sequence": 1,
            "itemgroup_name": "ig-VS",
            "itemgroup_sequence": 1,
            "item_name": "VSDAT",
            "created_date": "2022-06-08T20:05:39Z",
            "created_by": "Eric Emerton",
            "messages": [
                {
```

| Description / Request (URL) | Response |
| --- | --- |
| | "id": "OPY00000000V002",<br>"activity": "open__v",<br>"message": "This is my query on a form not visited yet, blank",<br>"message_date": "2022-06-08T20:05:39Z",<br>"message_by": "Eric Emerton"<br>},<br>{<br>"id": "OPY00000000V003",<br>"activity": "closed__v",<br>"message": "This is my reason for closing this query",<br>"message_date": "2022-06-08T20:06:30Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>},<br>{<br>"id": "OPW00000000V001",<br>"query_name": "VV-000233",<br>"manual": true,<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 2,<br>"itemgroup_name": "ig-AE",<br>"itemgroup_sequence": 1,<br>"item_name": "AETERM",<br>"created_date": "2022-06-08T20:09:48Z",<br>"created_by": "Eric Emerton",<br>"messages": [<br>{<br>"id": "OPY00000000W001",<br>"activity": "open__v",<br>"message": "Please check this value, this is a query from coding add via API",<br>"message_date": "2022-06-08T20:09:48Z",<br>"message_by": "Eric Emerton"<br>},<br>{<br>"id": "OPY00000000X001",<br>"activity": "open__v",<br>"message": "Please check this value, this is a query from |

| Description / Request (URL) | Response |
|---|---|
| | coding add via API (2nd add)",<br>"message_date": "2022-06-08T20:54:33Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>},<br>{<br>"id": "OPW00000000W001",<br>"query_name": "VV-000234",<br>"manual": true,<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "Concomitant-Medication",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-CM",<br>"itemgroup_sequence": 1,<br>"item_name": "CMTRT",<br>"created_date": "2022-06-08T20:57:31Z",<br>"created_by": "Eric Emerton",<br>"messages": [<br>{<br>"id": "OPY00000000X002",<br>"activity": "open__v",<br>"message": "Please check this value, this is a query from coding add via API (3rd add to CM)",<br>"message_date": "2022-06-08T20:57:31Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>}<br>]<br>} |
| Filter to just those in Open status | |
| .../app/cdm/queries?study_name=ABCP-2022-01_DEV1&query_status=open__v | (Like above, but to the queries where a touch of the query.. OR message exists, >= the parameter passed) |
| Filter to a specific point in time | |
| .../app/cdm/queries?study_name=ABCP-2022-01_DEV1&last_modified_date=2022-06-01T12:30:00Z | (Like above, but to the queries where a touch of the query.. OR message exists, >= the parameter passed) |

| Description / Request (URL) | Response |
| --- | --- |
| Filter to queries from a specific form design name | |
| `.../app/cdm/queries?study_name=ABCP-2022-01_DEV1&form_name=AE` | (Like above, but to the filter) |
| Filter to a specific subject | |
| `.../app/cdm/queries?study_name=ABCP-2022-01_DEV1&study_country=United States&site=101&subject=SCR-0002` | (Like above, but to the filter) |
| Filter to a specific country | |
| `.../app/cdm/queries?study_name=ABCP-2022-01_DEV1&study_country=United States` | (Like above, but to the filter) |

## 10.2 Retrieve Queries - By ID(s)

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#retrieve-queries-by-id">https://developer-cdms.veevavault.com/api/22.2/#retrieve-queries-by-id</a></td></tr>
<tr><td>Style</td><td>GET</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/queries</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td>Required</td><td>Name of the study</td></tr>
<tr><td>id</td><td>Required</td><td>The vault ID(s) of the queries to return. Multiple can be delimited by a comma</td></tr>
<tr><td>limit</td><td>Optional</td><td>Pagination API - different limit from the default 1000 can be used</td></tr>
<tr><td>offset</td><td>Optional</td><td>Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd</td></tr>
</table>
NOTE: Additional filtering / retrieval of queries is discussed in other sections (By Filters or By Medical Coding Definition)
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Use this API to retrieve the existing queries of a specific ID (or IDs). The ID of the query is returned on add via the API, or from previous usage of Retrieve Queries - By Filters endpoint.</li>
<li>The return value manual is such that false = added by the system.</li>
<li>When the query was added via a specific design rule, the rule_definition indicates the name of the rule that originated the query.</li>
<li>Queries are added at the item or event level. For event level, the form ‘down’ information is of course omitted</li>
<li>The return includes a messages array with each comment/message appended to the query. Reading top down in that array also reveals the progression of query status, and by whom</li>
</ul>
</td></tr>
</table>

## Examples

| Description / Request (URL) | Response |
|---|---|
| Single Query (where found) | |
| .../app/cdm/queries?study_name=ABCP-2022-01_DEV1&id=OPW00000000T001 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 1,<br>"total": 1<br>},<br>"queries": [<br>{<br>"id": "OPW00000000T001",<br>"query_name": "VV-000230",<br>"manual": false,<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IC",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IC2",<br>"itemgroup_sequence": 1,<br>"item_name": "RFICDAT",<br>"rule_definition": "IC_001",<br>"created_date": "2022-05-23T17:38:15Z",<br>"created_by": "System",<br>"messages": [<br>{<br>"id": "OPY00000000U001",<br>"activity": "open__v",<br>"message": "Informed consent date is not equal to screening visit date. Please review.",<br>"message_date": "2022-05-23T17:38:15Z",<br>"message_by": "System"<br>}<br>]<br>}<br>]<br>} |

| Description / Request (URL) | Response |
|---|---|
| **Attempt where ID not found** | |
| .../app/cdm/queries?study_name=ABCP-2022-01_DEV1&id=OPW00000000T011 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 0,<br>"total": 0<br>},<br>"queries": []<br>} |
| **Multiple Queries** | |
| .../app/cdm/queries?study_name=ABCP-2022-01_DEV1&id=OPW00000000T001,OPW00000000T002 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 2,<br>"total": 2<br>},<br>"queries": [<br>{<br>"id": "OPW00000000T001",<br>"query_name": "VV-000230",<br>"manual": false,<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IC",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IC2",<br>"itemgroup_sequence": 1,<br>"item_name": "RFICDAT",<br>"rule_definition": "IC_001",<br>"created_date": "2022-05-23T17:38:15Z",<br>"created_by": "System",<br>"messages": [<br>{<br>"id": "OPY00000000U001",<br>"activity": "open__v",<br>"message": "Informed consent date is not equal to screening |

| Description / Request (URL) | Response |
| --- | --- |
| | visit date. Please review.",<br>"message_date": "2022-05-23T17:38:15Z",<br>"message_by": "System"<br>}<br>]<br>},<br>{<br>"id": "OPW00000000U001",<br>"query_name": "VV-000231",<br>"manual": true,<br>"query_status": "closed__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0006",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"created_date": "2022-06-08T20:05:21Z",<br>"created_by": "Eric Emerton",<br>"messages": [<br>{<br>"id": "OPY00000000V001",<br>"activity": "open__v",<br>"message": "Here is query at the event level",<br>"message_date": "2022-06-08T20:05:22Z",<br>"message_by": "Eric Emerton"<br>},<br>{<br>"id": "OPY00000000V004",<br>"activity": "closed__v",<br>"message": null,<br>"message_date": "2022-06-08T21:22:34Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>}<br>]<br>} |

### 10.3 Retrieve Queries - By Medical Coding Definition

This section is grouped into the Medical Coding chapter, specifically the **Retrieve Coding Queries** section, later in the document.

## 10.4 Open Queries - By Study Context

<table>
<tr><td>Help Main</td><td>https://developer-cdms.veevavault.com/api/22.2/#open-query</td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/queries</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>queries</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>queries</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>queries</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>queries</td><td>Required</td><td>Design name in the study of the event group</td></tr>
<tr><td>eventgroup_sequence</td><td>queries</td><td>Optional</td><td>The specific sequence of an event group. 1 assumed if omitted</td></tr>
<tr><td>event_name</td><td>queries</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>form_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the form. Required when adding query to an item on a form</td></tr>
<tr><td>form_sequence</td><td>queries</td><td>Optional *</td><td>The specific sequence of the form. (if a query on an item of a form) 1 assumed if omitted.</td></tr>
<tr><td>itemgroup_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the item group location. Required when adding query to an item on a form</td></tr>
<tr><td>itemgroup_sequence</td><td>queries</td><td>Optional *</td><td>The specific item group sequence where the item resides (if adding query to an item. 1 assumed if omitted</td></tr>
<tr><td>item_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the item to have query set to (if an item level query)</td></tr>
<tr><td>message</td><td>queries</td><td>Required</td><td>The message of the query on add (255 character limit)</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is used to add a new query to an item on a form, or event, of a subject</li>
<li>WARNING: A locked form will not allow the add a new query, just at the EDC UI would disallow. Frozen forms will accept new queries added via the API.</li>
<li>For further manipulation of queries (answer, close, re-open), it's good practice to inspect the return on query API actions, for the id value. Later actions are simpler, API wise, by using the Query ID.</li>
<li>The API user’s role must have permission to add queries.</li>
<li>Additional abilities to open queries are available at:
<ul>
<li>Open Queries - By Item ID (different style of response)</li>
<li>Open Queries - By Event ID (different style of response)</li>
<li>Open Coding Queries</li>
</ul>
</li>
</ul>
</td></tr>
</table>

### Examples

<table>
<tr><th>Description / Request (URL)</th><th>Response</th></tr>
<tr><td colspan="2">Add one query, to an item inside a repeating event group</td></tr>
<tr><td>

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "queries": [
            {
                "study_country": "United States",
                "site": "101",
                "subject": "SCR-0003",
                "eventgroup_name": "egUNS",
                "eventgroup_sequence": 2,
                "event_name": "evUNS",
                "form_name": "VS",
                "form_sequence": 1,
                "itemgroup_name": "ig-VS",
                "itemgroup_sequence": 1,
                "item_name": "VSDAT",
                "message": "This is my query from API on
an item"
            }
        ]
}
```

</td><td>

```
{
    "responseStatus": "SUCCESS",
    "queries": [
        {
            "responseStatus": "SUCCESS",
            "id": "OPW00000000X003",
            "query_name": "VV-000237",
            "query_status": "open__v",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0003",
            "eventgroup_name": "egUNS",
            "eventgroup_sequence": 2,
            "event_name": "evUNS",
            "event_sequence": 1,
            "form_name": "VS",
            "form_sequence": 1,
            "itemgroup_name": "ig-VS",
            "itemgroup_sequence": 1,
            "item_name": "VSDAT"
        }
    ]
}
```

</td></tr>
<tr><td colspan="2">Add multiple, non-repeating areas (omit sequences)</td></tr>
<tr><td>

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "queries": [
            {
                "study_country": "United States",
                "site": "101",
                "subject": "SCR-0003",
                "eventgroup_name": "egSCR",
                "event_name": "evSCR",
                "form_name": "DM",
                "itemgroup_name": "ig-DM",
                "item_name": "BRTHDAT",
                "message": "This is my query from API on
an item - on DOB of Demography"
            },
            {
                "study_country": "United States",
                "site": "101",
                "subject": "SCR-0003",
```

</td><td>

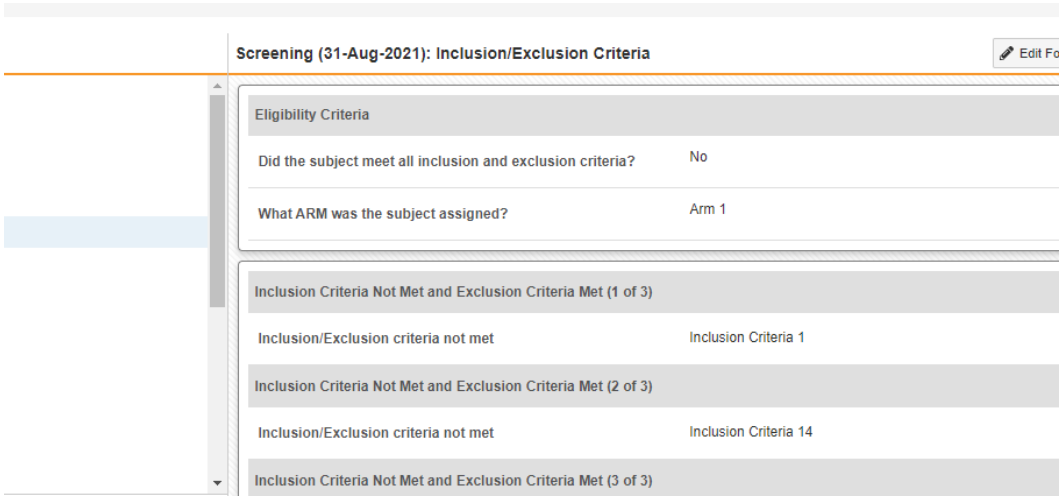
```
{
    "responseStatus": "SUCCESS",
    "queries": [
        {
            "responseStatus": "SUCCESS",
            "id": "OPW00000000Y001",
            "query_name": "VV-000238",
            "query_status": "open__v",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0003",
            "eventgroup_name": "egSCR",
            "eventgroup_sequence": 1,
            "event_name": "evSCR",
            "event_sequence": 1,
            "form_name": "DM",
            "form_sequence": 1,
            "itemgroup_name": "ig-DM",
            "itemgroup_sequence": 1,
```

</td></tr>
</table>

| Description / Request (URL) | Response |
|---|---|
| "eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"form_name": "VS",<br>"itemgroup_name": "ig-VS",<br>"item_name": "HEIGHT",<br>"message": "This is my query from API on an item - on Height of Vitals"<br>}<br>]<br>} | "item_name": "BRTHDAT"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"id": "OPW00000000Y002",<br>"query_name": "VV-000239",<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "VS",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-VS",<br>"itemgroup_sequence": 1,<br>"item_name": "HEIGHT"<br>}<br>]<br>} |
| **Query add to an event** | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"queries": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "egSCR",<br>"event_name": "evSCR",<br>"message": "This is my query on the screening event"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"queries": [<br>{<br>"responseStatus": "SUCCESS",<br>"id": "OPW00000000Y004",<br>"query_name": "VV-000241",<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1<br>}<br>]<br>} |

## Flow Based Examples

For the EDC form Inclusion/Exclusion of the example study:



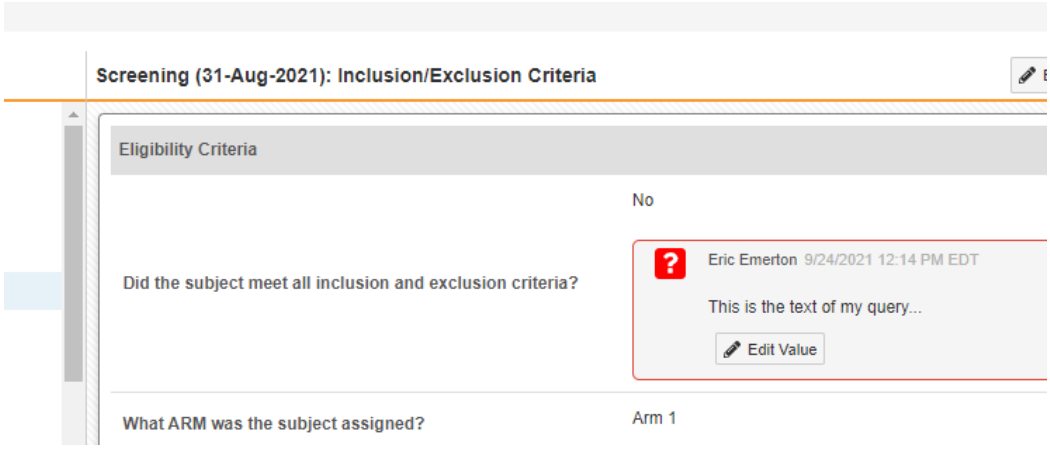

The post of a new query on the first field of the form is below. Note the full path from **event_group_name** through the **item_name**, including **all** sequences at each level.

```
{
    "study_name": "CSM Demo Study_DEV1",
    "queries": [
        {
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "form_name": "IE",
            "form_sequence": 1,
            "itemgroup_name": "IEEL",
            "itemgroup_sequence": 1,
            "item_name": "IEYN1" ,
            "message": "This is the text of my query..."
        }
    ]
```

.. yields the return with **id** and **query_name** values, plus **query_status**:

```
{
    "responseStatus": "SUCCESS",
    "queries": [
        {
            "responseStatus": "SUCCESS",
            "id": "OPW000000042001",
            "query_name": "VV-002811",
            "query_status": "open__v",
            "study_country": "United States",
            "site": "002",
```


The result in the EDC UI:




For adding a query at the event level, the post only needs go through the event name:

Params Authorization Headers (12) Body Pre-request Script Tests Settings

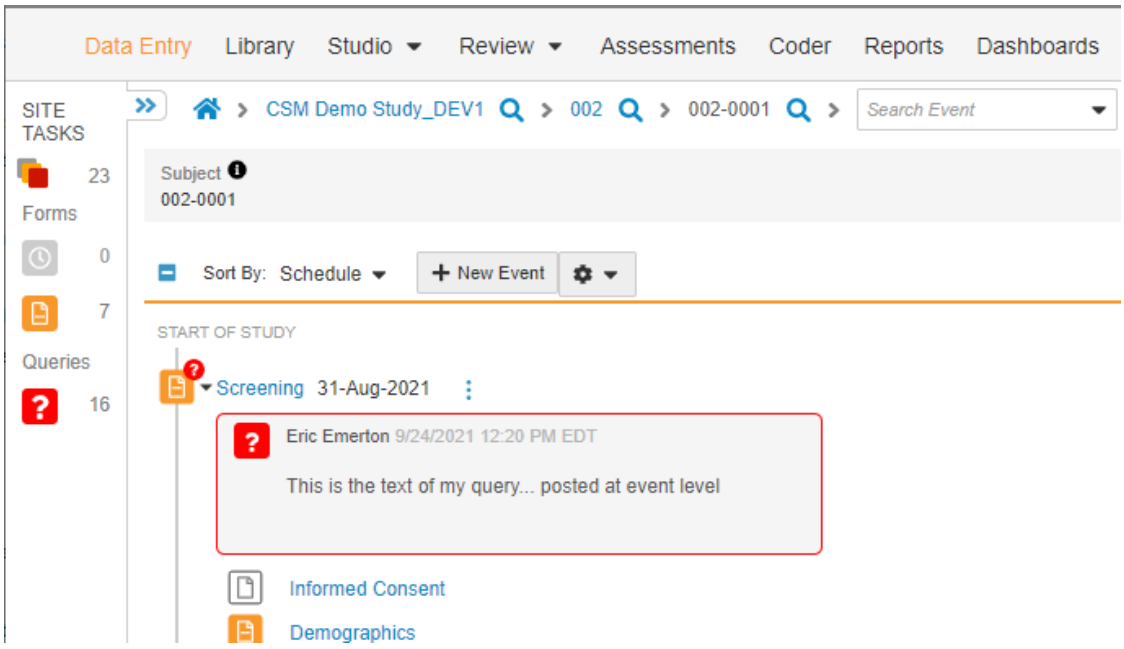
none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "study_name": "CSM Demo Study_DEV1",
    "queries": [
        {
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "message": "This is the text of my query... posted at event level"
        }
    ]
}
```

Body Cookies Headers (16) Test Results Status: 200 OK Time: 1111 ms Si

Pretty Raw Preview Visualize JSON


```
{
    "responseStatus": "SUCCESS",
    "queries": [
        {
            "responseStatus": "SUCCESS",
            "id": "OPW000000041002",
            "query_name": "VV-002812",
            "query_status": "open__v",
            "study_country": "United States",
            "site": "002",
            "subject": "002-0001",
            "eventgroup_name": "eg_SCREENING",
            "eventgroup_sequence": 1,
            "event_name": "ev_SCREENING",
            "event_sequence": 1
```

## 10.5 Open Queries - By Item ID

<table>
<tr><td>Help Main</td><td>https://developer-cdms.veevavault.com/api/22.2/#open-query-by-item-id</td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/items/actions/openquery</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>manual</td><td></td><td>Optional</td><td>Boolean - use only true/false. When omitted, true is used. In this context, manual = false, or ‘System’ classified would imply not added by an end user in the UI, instead ‘a system’ (even though not the CDMS system, the way study rules are added by ‘system’). All queries in the body are given the same manual property value on add.</td></tr>
<tr><td>id</td><td>queries</td><td>Required</td><td>The ID of the item where the query is to be added.</td></tr>
<tr><td>message</td><td>queries</td><td>Required</td><td>The message of the query on add (255 character limit)</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is used to add a new query to an item, but using the Vault ID for the item where the query is added. This endpoint is only know if the Item’s ID is known (obtained through exports, other other endpoints that expose the ID)</li>
<li>This endpoint returns slightly different response information from the Open Queries - By Study Context endpoint. It does not include the event / form level design information, i.e. presuming the caller of the endpoint already knows all that. It does, though, return messages (the first one, with its own message ID) and other information similar to the return of Retrieve Queries - By Filters</li>
<li>WARNING: A locked form will not allow the add a new query, just at the EDC UI would disallow. Frozen forms will accept new queries added via the API.</li>
<li>Additional abilities to open queries are available at:
<ul>
<li>Open Queries - By Study Context (different style of response)</li>
<li>Open Queries - By Event ID</li>
<li>Open Coding Queries (different style of response)</li>
</ul>
</li>
</ul>
</td></tr>
</table>

### Examples

| Description / Request (URL) | Response |
|---|---|
| Attempt add of two, one fails, one succeeds | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"queries": [<br>{<br>"id": "V5D00000000K011",<br>"message": ""<br>},<br>{<br>"id": "V5D00000000K012",<br>"message": "Open query on item from open by item ID"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"queries": [<br>{<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "Message is required",<br>"item_id": "V5D00000000K011"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"id": "OPW0000000011001",<br>"query_status": "open__v",<br>"item_id": "V5D00000000K012",<br>"query_name": "VV-000246",<br>"created_date": "2022-06-16T14:52:03Z",<br>"created_by": "Eric Emerton",<br>"messages": [<br>{<br>"id": "OPY000000012001",<br>"activity": "open__v",<br>"message": "Open query on item from open by item ID",<br>"message_date": "2022-06-16T14:52:04Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>}<br>]<br>} |

## 10.6 Open Queries - By Event ID

<table>
<tr><td>Help Main</td><td>https://developer-cdms.veevavault.com/api/22.2/#open-query-by-event-id</td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/events/actions/openquery</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>manual</td><td></td><td>Optional</td><td>Boolean - use only true/false. When omitted, true is used. In this context, manual = false, or ‘System’ classified would imply not added by an end user in the UI, instead ‘a system’ (even though not the CDMS system, the way study rules are added by ‘system’). All queries in the body are given the same manual property value on add.</td></tr>
<tr><td>id</td><td>queries</td><td>Required</td><td>The ID of the event where the query is to be added.</td></tr>
<tr><td>message</td><td>queries</td><td>Required</td><td>The message of the query on add (255 character limit)</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is used to add a new query to an item, but using the Vault ID for the event where the query is added. This endpoint is only know if the Event’s ID is known (obtained through exports, other other endpoints that expose the ID)</li>
<li>This endpoint returns slightly different response information from the Open Queries - By Study Context endpoint. It does not include the event / form level design information, i.e. presuming the caller of the endpoint already knows all that. It does, though, return messages (the first one, with its own message ID) and other information similar to the return of Retrieve Queries - By Filters</li>
<li>WARNING: A locked form will not allow the add a new query, just at the EDC UI would disallow. Frozen forms will accept new queries added via the API.</li>
<li>Additional abilities to open queries are available at:
<ul>
<li>Open Queries - By Study Context (different style of response)</li>
<li>Open Queries - By Item ID</li>
<li>Open Coding Queries (different style of response)</li>
</ul>
</li>
</ul>
</td></tr>
</table>

### Examples

| Description / Request (URL) | Response |
|---|---|
| Attempt add of two, one fails, one succeeds | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"queries": [<br>{<br>"id": "OPS00000001X002",<br>"message": "Open query on Event from open by event ID - 1st"<br>},<br>{<br>"id": "OPS00000001X001",<br>"message": "Open query on Event from open by event ID - 2nd"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"queries": [<br>{<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "[Event] with id [OPS00000001X002] not found",<br>"event_id": "OPS00000001X002"<br>},<br>{<br>"responseStatus": "SUCCESS",<br>"id": "OPW0000000011001",<br>"query_status": "open__v",<br>"event_id": "OPS00000001X001",<br>"query_name": "VV-000248",<br>"created_date": "2022-06-16T15:52:03Z",<br>"created_by": "Eric Emerton",<br>"messages": [<br>{<br>"id": "OPY000000012001",<br>"activity": "open__v",<br>"message": "Open query on Event from open by event ID - 2nd",<br>"message_date": "2022-06-16T15:52:04Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>}<br>]<br>} |

## 10.7 Answer Queries - By Study Context

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#answer-query">https://developer-cdms.veevavault.com/api/22.2/#answer-query</a></td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/queries/actions/answer</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>queries</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>queries</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>queries</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>queries</td><td>Required</td><td>Design name in the study of the event group</td></tr>
<tr><td>eventgroup_sequence</td><td>queries</td><td>Optional</td><td>The specific sequence of an event group. 1 assumed if omitted</td></tr>
<tr><td>event_name</td><td>queries</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>form_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the form. Required when query is on an item on a form</td></tr>
<tr><td>form_sequence</td><td>queries</td><td>Optional *</td><td>The specific sequence of the form. (if a query on an item of a form) 1 assumed if omitted.</td></tr>
<tr><td>itemgroup_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the item group location. Required when query is an item on a form</td></tr>
<tr><td>itemgroup_sequence</td><td>queries</td><td>Optional *</td><td>The specific item group sequence where the item resides (if a query on an item). 1 assumed if omitted</td></tr>
<tr><td>item_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the item (if a query on an item)</td></tr>
<tr><td>message</td><td>queries</td><td>Required</td><td>The message of the query on answer(255 character limit)</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is used to answer an existing query, via the subject information, and study design location.</li>
<li>The action applies whether the query is against an item (on a form), or an event.</li>
<li>The query must be in Open status to perform this action</li>
<li>WARNING: If multiple queries exist at the item (or event), an error will be returned.</li>
<li>WARNING: A locked form will not allow the editing of a query, just at the EDC UI would disallow. Frozen forms will accept the edit added via the API.</li>
<li>One can also answer a query by query ID - see the next section Answer Queries - By Query ID</li>
<li>TIP: If query IDs are known (returned in Retrieve Queries), then use of this option By ID is preferred.</li>
</ul>
</td></tr>
</table>

### Examples

| Description / Request (URL) | Response |
|---|---|
| Answer of one query | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"queries": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IC",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IC2",<br>"itemgroup_sequence": 1,<br>"item_name": "RFICDAT",<br>"message": "This is the text on answer query"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"queries": [<br>{<br>"responseStatus": "SUCCESS",<br>"id": "OPW00000000T001",<br>"query_name": "VV-000230",<br>"query_status": "answered__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0002",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IC",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IC2",<br>"itemgroup_sequence": 1,<br>"item_name": "RFICDAT"<br>}<br>]<br>} |

## 10.8 Answer Queries - By Query ID

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#answer-query |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/queries/actions/answer |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| id | queries | Required | The Vault ID of the query. This value is returned on add (if added by API), and also on **Get Queries** endpoint return of the query. |
| message | queries | Required | The message of the query on answer(255 character limit) |

**Notes**

- The action is used to answer an **existing** query, via the subject information, and known query ID.
- The action applies whether the query is against an item (on a form), or an event
- The query must be in **Open** status to perform this action
- **WARNING:** A locked form will not allow the editing of a query, just at the EDC UI would disallow. Frozen forms will accept the edit added via the API.
- One can also answer a query by subject information / study design also - see the previous section **Answer Queries - By Study Context**
- **TIP:** If query IDs are known (returned in **Retrieve Queries**), then use of this option By ID is preferred.

### Examples

| Description / Request (URL) | Response |
|---|---|
| Answer of one query | |

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "queries": [
        {
            "id": "OPW000000014003",
            "message": "Answered Query via API"
        }
    ]
}
```

```
{
    "responseStatus": "SUCCESS",
    "queries": [
        {
            "responseStatus": "SUCCESS",
            "id": "OPW000000014003",
            "query_name": "VV-000255",
            "query_status": "answered__v",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0003",
            "eventgroup_name": "egSCR",
            "eventgroup_sequence": 1,
            "event_name": "evSCR",
            "event_sequence": 1,
            "form_name": "DM",
            "form_sequence": 1,
            "itemgroup_name": "ig-DM",
```

| Description / Request (URL) | Response |
| --- | --- |
| | "itemgroup_sequence": 1,<br>"item_name": "SEX"<br>}<br>]<br>} |

## 10.9 Close Queries - By Study Context

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#close-query">https://developer-cdms.veevavault.com/api/22.2/#close-query</a></td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/queries/actions/close</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>queries</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>queries</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>queries</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>queries</td><td>Required</td><td>Design name in the study of the event group</td></tr>
<tr><td>eventgroup_sequence</td><td>queries</td><td>Optional</td><td>The specific sequence of an event group. 1 assumed if omitted</td></tr>
<tr><td>event_name</td><td>queries</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>form_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the form. Required when query is on an item on a form</td></tr>
<tr><td>form_sequence</td><td>queries</td><td>Optional *</td><td>The specific sequence of the form. (if a query on an item of a form) 1 assumed if omitted.</td></tr>
<tr><td>itemgroup_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the item group location. Required when query is an item on a form</td></tr>
<tr><td>itemgroup_sequence</td><td>queries</td><td>Optional *</td><td>The specific item group sequence where the item resides (if a query on an item). 1 assumed if omitted</td></tr>
<tr><td>item_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the item (if a query on an item)</td></tr>
<tr><td>message</td><td>queries</td><td>Optional</td><td>The message of the query on close, optional (255 character limit)</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is used to close an existing query, via the subject information, and study design location.</li>
<li>The action applies whether the query is against an item (on a form), or an event</li>
<li>The query must be in Answered status to perform this action</li>
<li>WARNING: If multiple queries exist at the item (or event), an error will be returned.</li>
<li>WARNING: A locked form will not allow the editing of a query, just at the EDC UI would disallow. Frozen forms will accept the edit added via the API.</li>
<li>One can also close a query by query ID - see the next section Close Queries - By Query ID</li>
<li>TIP: If query IDs are known (returned in Retrieve Queries), then use of this option By ID is preferred.</li>
</ul>
</td></tr>
</table>

### Examples

| Description / Request (URL) | Response |
|---|---|
| Close one query (with message) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"queries": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE1",<br>"itemgroup_sequence": 1,<br>"item_name": "IEYN",<br>"message": "This is the text on close query"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"queries": [<br>{<br>"responseStatus": "SUCCESS",<br>"id": "OPW000000012005",<br>"query_name": "VV-000256",<br>"query_status": "closed__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE1",<br>"itemgroup_sequence": 1,<br>"item_name": "IEYN"<br>}<br>]<br>} |

## 10.10 Close Queries - By Query ID

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#close-query-batch |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/queries/actions/closebyid |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| id | queries | Required | The Vault ID of the query. This value is returned on add (if added by API), and also on **Get Queries** endpoint return of the query. |
| message | queries | Optional | The message of the query on close, optional (255 character limit) |

**Notes**

- The action is used to answer an **existing** query, via known query ID(s).
- There is a limit of **500** per request.
- The action applies whether the query is against an item (on a form), or an event
- The query must be in **Answered** status to perform this action
- **WARNING:** A locked form will not allow the editing of a query, just at the EDC UI would disallow. Frozen forms will accept the edit added via the API.
- One can also answer a query by subject information / study design also - see the previous section **Close Queries - By Study Design Context.** The endpoint for study context originally included a 'By ID' style, but that is to be deprecated in favor of this endpoint (../actions/closebyid)
- **TIP:** If query IDs are known (returned in **Retrieve Queries**), then use of this option By ID is preferred.

### Examples

| Description / Request (URL) | Response |
|---|---|
| Close of one query (with message) | |

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "queries": [
        {
            "id": "OPW000000014003",
            "message": "Closed Query via API - by ID"
        }
    ]
}
```

```
{
    "responseStatus": "SUCCESS",
    "queries": [
        {
            "responseStatus": "SUCCESS",
            "id": "OPW000000014003",
            "message": "Closed Query via API - by ID",
            "query_status": "closed__v",
        }
    ]
}
```

## 10.11 Reopen Queries - By Study Context

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#reopen-query">https://developer-cdms.veevavault.com/api/22.2/#reopen-query</a></td></tr>
<tr><td>Style</td><td>POST</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/queries/actions/reopen</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Array</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td></td><td>Required</td><td>Name of the study</td></tr>
<tr><td>study_country</td><td>queries</td><td>Required</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>queries</td><td>Required</td><td>Name/number of the site</td></tr>
<tr><td>subject</td><td>queries</td><td>Required</td><td>Subject to act on</td></tr>
<tr><td>eventgroup_name</td><td>queries</td><td>Required</td><td>Design name in the study of the event group</td></tr>
<tr><td>eventgroup_sequence</td><td>queries</td><td>Optional</td><td>The specific sequence of an event group. 1 assumed if omitted</td></tr>
<tr><td>event_name</td><td>queries</td><td>Required</td><td>Design name in the study of the event</td></tr>
<tr><td>form_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the form. Required when query is on an item on a form</td></tr>
<tr><td>form_sequence</td><td>queries</td><td>Optional *</td><td>The specific sequence of the form. (if a query on an item of a form) 1 assumed if omitted.</td></tr>
<tr><td>itemgroup_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the item group location. Required when query is an item on a form</td></tr>
<tr><td>itemgroup_sequence</td><td>queries</td><td>Optional *</td><td>The specific item group sequence where the item resides (if a query on an item). 1 assumed if omitted</td></tr>
<tr><td>item_name</td><td>queries</td><td>Optional *</td><td>Design name in the study of the item (if a query on an item)</td></tr>
<tr><td>message</td><td>queries</td><td>Required</td><td>The message of the query on answer(255 character limit)</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>The action is used to reopen an existing query, via the subject information, and study design location.</li>
<li>The action applies whether the query is against an item (on a form), or an event</li>
<li>The query must be in Closed status to perform this action</li>
<li>WARNING: If multiple queries exist at the item (or event), an error will be returned.</li>
<li>WARNING: A locked form will not allow the editing of a query, just at the EDC UI would disallow. Frozen forms will accept the edit added via the API.</li>
<li>One can also reopen a query by query ID - see the next section Reopen Queries - By Query ID</li>
<li>TIP: If query IDs are known (returned in Retrieve Queries), then use of this option By ID is preferred.</li>
</ul>
</td></tr>
</table>

## Examples

| Description / Request (URL) | Response |
|---|---|
| Reopen of one query | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"queries": [<br>{<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE1",<br>"itemgroup_sequence": 1,<br>"item_name": "IEYN",<br>"message": "This is the text on reopen query"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"queries": [<br>{<br>"responseStatus": "SUCCESS",<br>"id": "OPW000000012005",<br>"query_name": "VV-000256",<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0003",<br>"eventgroup_name": "egSCR",<br>"eventgroup_sequence": 1,<br>"event_name": "evSCR",<br>"event_sequence": 1,<br>"form_name": "IE",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-IE1",<br>"itemgroup_sequence": 1,<br>"item_name": "IEYN"<br>}<br>]<br>} |

## 10.12 Reopen Queries - By Query ID

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#reopen-query |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/queries/actions/reopen |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| id | queries | Required | The Vault ID of the query. This value is returned on add (if added by API), and also on **Get Queries** endpoint return of the query. |
| message | queries | Required | The message of the query on reopen (255 character limit) |

**Notes**

- The action is used to reopen an **existing** query, via the subject information, and known query ID.
- The action applies whether the query is against an item (on a form), or an event
- The query must be in **Closed** status to perform this action
- **WARNING:** A locked form will not allow the editing of a query, just at the EDC UI would disallow. Frozen forms will accept the edit added via the API.
- One can also reopen a query by subject information / study design also - see the previous section **Reopen Queries - By Study Context**
- **TIP:** If query IDs are known (returned in **Retrieve Queries**), then use of this option By ID is preferred.

### Examples

| Description / Request (URL) | Response |
|---|---|
| Answer of one query | |

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "queries": [
        {
            "id": "OPW000000012006",
            "message": "Reopened query via API by ID."
        }
    ]
}
```

```
{
    "responseStatus": "SUCCESS",
    "queries": [
        {
            "responseStatus": "SUCCESS",
            "id": "OPW000000012006",
            "query_name": "VV-000257",
            "query_status": "open__v",
            "study_country": "United States",
            "site": "101",
            "subject": "SCR-0003",
            "eventgroup_name": "egSCR",
            "eventgroup_sequence": 1,
            "event_name": "evSCR",
            "event_sequence": 1,
            "form_name": "IC",
            "form_sequence": 1,
```

| Description / Request (URL) | Response |
| --- | --- |
| | "itemgroup_name": "ig-IC2",<br>"itemgroup_sequence": 1,<br>"item_name": "RFICDAT"<br>}<br>]<br>} |

# 11 Medical Coding

## 11.1 Retrieve Coding Requests

<table>
<tr><td>Help Main</td><td><a href="https://developer-cdms.veevavault.com/api/22.2/#retrieve-coding-requests">https://developer-cdms.veevavault.com/api/22.2/#retrieve-coding-requests</a></td></tr>
<tr><td>Style</td><td>GET</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/coder/codingrequests</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td>Required</td><td>Name of the study</td></tr>
<tr><td>coding_item_definition</td><td>Required</td><td>See the Study Design -> Retrieve Coding Form Definitions for retrieving these values. There, the <code>coding_item_definition_name</code> is the value to use into this endpoint.</td></tr>
<tr><td>study_country</td><td>Optional</td><td>Name of the study country (of the subject’s site)</td></tr>
<tr><td>site</td><td>Optional</td><td>Name/number of the site, requires study country (of the site) if using</td></tr>
<tr><td>subject</td><td>Optional</td><td>Subject number, requires country and site if using</td></tr>
<tr><td>last_modified_date</td><td></td><td>Filter to coding requests modified since a certain date. Format: yyyy-MM-ddTHH:mm:ssZ</td></tr>
<tr><td>limit</td><td>Optional</td><td>Pagination API - different limit from the default 1000 can be used</td></tr>
<tr><td>offset</td><td>Optional</td><td>Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Use this API to retrieve the medical coding requests</li>
<li>Required is the coding_item_definition (e.g., AE, CM, etc.). Only one coding form (its requests) can be returned at once. To get the appropriate value for the study / coding item definition, use the endpoint Study Design -> Retrieve Coding Form Definitions. Specifically the response parameter: <code>coding_item_definition_name</code></li>
<li>Filter for requests modified since a point is done with the last_modified_date parameter. The format is: yyyy-MM-ddTHH:mm:ssZ</li>
</ul>
</td></tr>
</table>

### Examples

| Description / Request (URL) | Response |
|---|---|
| All requests of specific definition (MedDRA) | |
| .../app/cdm/subjects?study_name=ABCPharma_DEV1&coding_item_definition=VV-000023 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 2,<br>"total": 2<br>},<br>"codingrequests": [<br>{<br>"coding_item_definition": "V0T000000002001",<br>"form_status": "Active",<br>"form_type": "AE",<br>"dictionary_definition": "MedDRA",<br>"requests": [<br>{<br>"id": "V0V000000003001",<br>"site": "101",<br>"subject": "SCR-0001",<br>"verbatim": "My Term 1",<br>"start_date": "",<br>"stop_date": "",<br>"event": "LOGS",<br>"event_sequence": 1,<br>"form": "Adverse-Event",<br>"form_sequence": 1,<br>"item_group": "ig-AE",<br>"item_group_sequence": 1,<br>"item": "AETERM",<br>"coding_status": "Open",<br>"last_coded_by_third_party": "",<br>"last_coded_date": "",<br>"assigned_to_third_party": "",<br>"last_modified_date": "2022-05-23T17:34:46Z",<br>"created_date": "2022-05-23T17:34:46Z",<br>"created_by": "System",<br>"other_properties": {<br>"My Label for SAE Form Uploaded": null<br>},<br>"seriousness": "Yes",<br>"severity": "",<br>"llt_code": "",<br>"llt": "",<br>"pt_code": "", |

| Description / Request (URL) | Response |
| --- | --- |
| | "pt": "",<br>"hlt_code": "",<br>"hlt": "",<br>"hlgt_code": "",<br>"hlgt": "",<br>"soc": "",<br>"soc_code": "",<br>"primary_path": ""<br>},<br>{<br>"id": "V0V000000003002",<br>"site": "101",<br>"subject": "SCR-0001",<br>"verbatim": "My AE 2",<br>"start_date": "2022-05-01T00:00:00Z",<br>"stop_date": "",<br>"event": "LOGS",<br>"event_sequence": 1,<br>"form": "Adverse-Event",<br>"form_sequence": 2,<br>"item_group": "ig-AE",<br>"item_group_sequence": 1,<br>"item": "AETERM",<br>"coding_status": "Coded",<br>"last_coded_by_third_party": "",<br>"last_coded_date": "2022-06-02T18:35:03Z",<br>"assigned_to_third_party": "",<br>"last_modified_date": "2022-06-02T18:35:03Z",<br>"created_date": "2022-05-23T17:35:27Z",<br>"created_by": "System",<br>"other_properties": {<br>"My Label for SAE Form Uploaded": "true"<br>},<br>"seriousness": "No",<br>"severity": "",<br>"llt_code": "10027599",<br>"llt": "Migraine",<br>"pt_code": "10027599",<br>"pt": "Migraine",<br>"hlt_code": "10027603",<br>"hlt": "Migraine headaches",<br>"hlgt_code": "10019231",<br>"hlgt": "Headaches",<br>"soc": "Nervous system disorders",<br>"soc_code": "10029205",<br>"primary_path": "Y"<br>}<br>] |

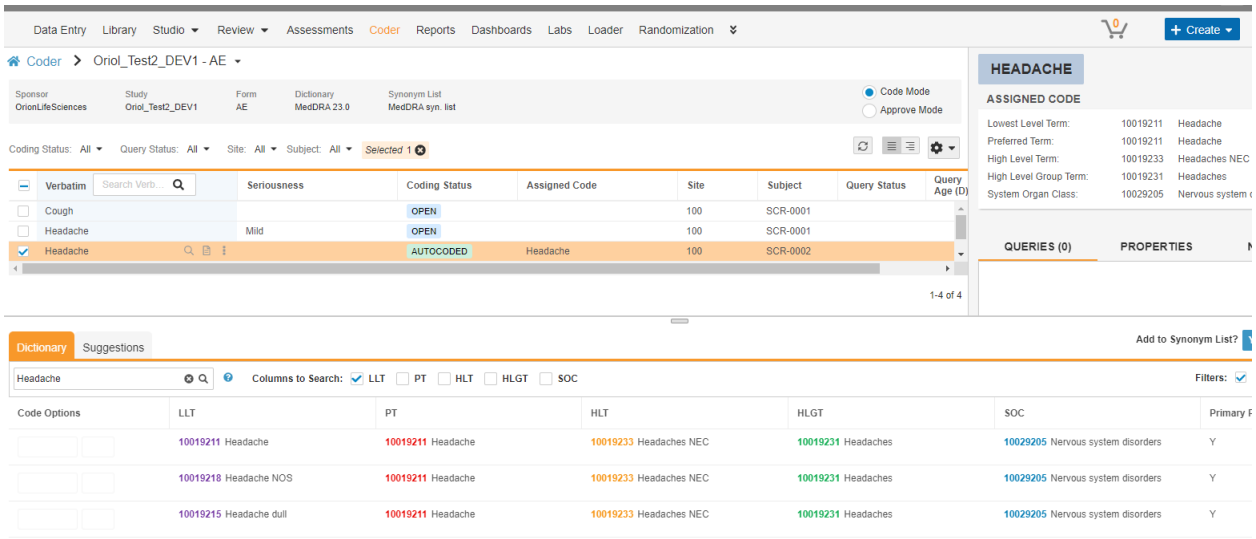
| Description / Request (URL) | Response |
|---|---|
| | }<br>]<br>} |
| All requests of specific definition (WHODrug) | |
| .../app/cdm/subjects?study_name=ABCPharma_DEV1&coding_item_definition=VV-000024 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 2,<br>"total": 2<br>},<br>"codingrequests": [<br>{<br>"coding_item_definition": "V0T000000004001",<br>"form_status": "Active",<br>"form_type": "ConMed",<br>"dictionary_definition": "WHODrug C3",<br>"requests": [<br>{<br>"id": "V0V000000004001",<br>"site": "101",<br>"subject": "SCR-0001",<br>"verbatim": "Aspirin",<br>"start_date": "",<br>"stop_date": "",<br>"event": "LOGS",<br>"event_sequence": 1,<br>"form": "Concomitant-Medication",<br>"form_sequence": 1,<br>"item_group": "ig-CM",<br>"item_group_sequence": 1,<br>"item": "CMTRT",<br>"coding_status": "Coded",<br>"last_coded_by_third_party": "",<br>"last_coded_date": "2022-06-02T20:02:26Z",<br>"assigned_to_third_party": "",<br>"last_modified_date": "2022-06-02T20:02:26Z",<br>"created_date": "2022-06-02T20:00:44Z",<br>"created_by": "System",<br>"other_properties": {},<br>"indication": "Headache",<br>"route": "Oral",<br>"drug_name": "Aspirin",<br>"drug_code": "00002701004",<br>"preferred_code": "00002701001", |

| Description / Request (URL) | Response |
| --- | --- |
| | "preferred_name": "Acetylsalicylic acid",<br>"atc4_code": "N02BA",<br>"atc4": "Salicylic acid and derivatives",<br>"atc3_code": "N02B",<br>"atc3": "OTHER ANALGESICS AND ANTIPYRETICS",<br>"atc2_code": "N02",<br>"atc2": "ANALGESICS",<br>"atc1_code": "N",<br>"atc1": "NERVOUS SYSTEM"<br>},<br>{<br>"id": "V0V000000004002",<br>"site": "101",<br>"subject": "SCR-0001",<br>"verbatim": "Ibuprofen",<br>"start_date": "",<br>"stop_date": "",<br>"event": "LOGS",<br>"event_sequence": 1,<br>"form": "Concomitant-Medication",<br>"form_sequence": 2,<br>"item_group": "ig-CM",<br>"item_group_sequence": 1,<br>"item": "CMTRT",<br>"coding_status": "Open",<br>"last_coded_by_third_party": "",<br>"last_coded_date": "",<br>"assigned_to_third_party": "",<br>"last_modified_date": "2022-06-02T20:01:47Z",<br>"created_date": "2022-06-02T20:01:47Z",<br>"created_by": "System",<br>"other_properties": {},<br>"indication": "My indication",<br>"route": "Nasal",<br>"drug_name": "",<br>"drug_code": "",<br>"preferred_code": "",<br>"preferred_name": "",<br>"atc4_code": "",<br>"atc4": "",<br>"atc3_code": "",<br>"atc3": "",<br>"atc2_code": "",<br>"atc2": "",<br>"atc1_code": "",<br>"atc1": ""<br>}<br>] |

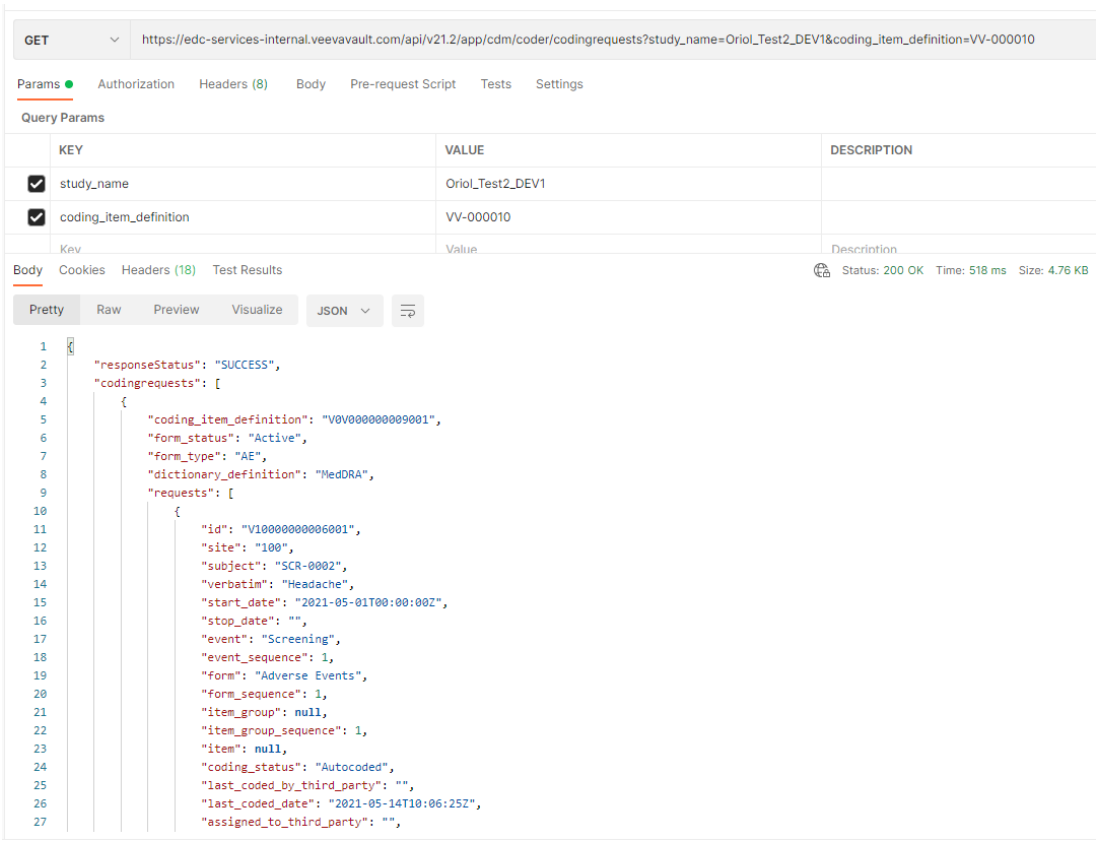
| Description / Request (URL) | Response |
| --- | --- |
| | }<br>]<br>} |
| Filter to a specific subject | |
| .../app/cdm/subjects?study_name=ABCPharma_DEV1&coding_item_definition=VV-000023&study_country=United States&site=101&subject=101-001 | (Like above, but to the filter) |

## Flow Based Examples

These medical coding requests in the Coder UI



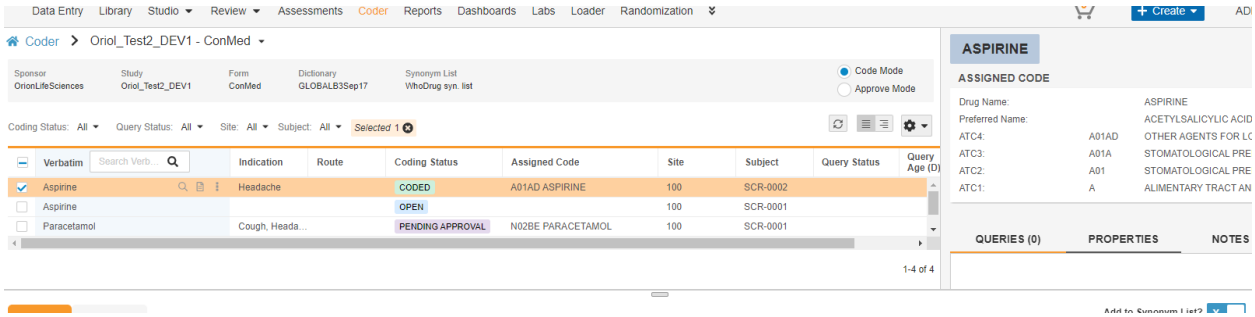

Are retrieved with this API GET:




Further down the return is the coding for the AE:

```
            "created_by": "System",
            "other_properties": {
                "Other Action Taken": null
            },
            "seriousness": "",
            "severity": "",
            "llt_code": "10019211",
            "llt": "Headache",
            "pt_code": "10019211",
            "pt": "Headache",
            "hlt_code": "10019233",
            "hlt": "Headaches NEC",
            "hlgt_code": "10019231",
            "hlgt": "Headaches",
            "soc": "Nervous system disorders",
            "soc_code": "10029205",
            "primary_path": "Y"
        },
```


For WHODrug requests (ConMeds, etc.), in EDC we see coding:




Returned in API for detail of coding as:


```
            "coding_status": "Coded",
            "last_coded_by_third_party": "",
            "last_coded_date": "2021-05-14T10:19:24Z",
            "assigned_to_third_party": "",
            "last_modified_date": "2021-05-14T11:51:16Z",
            "created_date": "2021-05-14T10:18:59Z",
            "created_by": "System",
            "other_properties": {},
            "indication": "Headache",
            "route": "",
            "drug_name": "ASPIRINE",
            "drug_code": "00002701014",
            "preferred_code": "00002701001",
            "preferred_name": "ACETYLSALICYLIC ACID",
            "atc4_code": "A01AD",
            "atc4": "OTHER AGENTS FOR LOCAL ORAL TREATMENT",
            "atc3_code": "A01A",
            "atc3": "STOMATOLOGICAL PREPARATIONS",
            "atc2_code": "A01",
            "atc2": "STOMATOLOGICAL PREPARATIONS",
            "atc1_code": "A",
            "atc1": "ALIMENTARY TRACT AND METABOLISM"
        },
```

## 11.2 Retrieve Coding Queries

<table>
<tr><td>Help Main</td><td>https://developer-cdms.veevavault.com/api/22.2/#retrieve-coding-queries</td></tr>
<tr><td>Style</td><td>GET</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/queries</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td>Required</td><td>Name of the study</td></tr>
<tr><td>coding_item_definition</td><td>Required</td><td>See the Study Design -> Retrieve Coding Form Definitions for retrieving these values. There, the <code>coding_item_definition_id</code> is the value to use into this endpoint.</td></tr>
<tr><td>limit</td><td>Optional</td><td>Pagination API - different limit from the default 1000 can be used</td></tr>
<tr><td>offset</td><td>Optional</td><td>Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Use this API to retrieve the existing queries of a specific ID (or IDs). The ID of the query is returned on add via the API, or from Retrieve Queries endpoint.</li>
<li>The return value manual is such that false = added by the system.</li>
<li>When the query was added via a specific design rule, the rule_definition indicates the name of the rule that originated the query.</li>
<li>Queries are added at the item or event level. For event level, the form ‘down’ information is of course omitted</li>
<li>For the coding_item_definition parameter, the appropriate value for the study / coding item definition, use the endpoint Study Design -> Retrieve Coding Form Definitions. Specifically the response parameter: <code>coding_item_definition_id</code></li>
<li>The return includes a messages array with each comment/message appended to the query. Reading top down in that array also reveals the progression of query status, and by whom</li>
<li>Additional filtering / retrieval of queries is discussed in other sections (By Filters or By Medical Coding Definition, see Queries chapter)</li>
</ul>
</td></tr>
</table>

### Examples

| Description / Request (URL) | Response |
|---|---|
| By specific coding definition ID | |
| .../app/cdm/queries?coding_item_definition=V0T000000002001&study_name=ABCP-2022-01_DEV1 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 1,<br>"total": 1<br>},<br>"queries": [<br>{<br>"id": "OPW00000000V001",<br>"query_name": "VV-000233",<br>"manual": true,<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 2,<br>"itemgroup_name": "ig-AE",<br>"itemgroup_sequence": 1,<br>"item_name": "AETERM",<br>"coding_item_definition": "V0T000000002001",<br>"created_date": "2022-06-08T20:09:48Z",<br>"created_by": "Eric Emerton",<br>"messages": [<br>{<br>"id": "OPY00000000W001",<br>"activity": "open__v",<br>"message": "Please check this value, this is a query from coding add via API",<br>"message_date": "2022-06-08T20:09:48Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>}<br>]<br>} |
| Queries across two coding definitions | |

| Description / Request (URL) | Response |
| --- | --- |
| .../app/cdm/queries?coding_item_definition=V0T000000002001,V0T000000004001&study_name=ABCP-2022-01_DEV1 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 2,<br>"total": 2<br>},<br>"queries": [<br>{<br>"id": "OPW00000000V001",<br>"query_name": "VV-000233",<br>"manual": true,<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001",<br>"eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "AE",<br>"form_sequence": 2,<br>"itemgroup_name": "ig-AE",<br>"itemgroup_sequence": 1,<br>"item_name": "AETERM",<br>"coding_item_definition": "V0T000000002001",<br>"created_date": "2022-06-08T20:09:48Z",<br>"created_by": "Eric Emerton",<br>"messages": [<br>{<br>"id": "OPY00000000W001",<br>"activity": "open__v",<br>"message": "Please check this value, this is a query from coding add via API",<br>"message_date": "2022-06-08T20:09:48Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>},<br>{<br>"id": "OPW00000000W001",<br>"query_name": "VV-000234",<br>"manual": true,<br>"query_status": "open__v",<br>"study_country": "United States",<br>"site": "101",<br>"subject": "SCR-0001", |

| Description / Request (URL) | Response |
| --- | --- |
| | "eventgroup_name": "LOGS",<br>"eventgroup_sequence": 1,<br>"event_name": "LOGS",<br>"event_sequence": 1,<br>"form_name": "Concomitant-Medication",<br>"form_sequence": 1,<br>"itemgroup_name": "ig-CM",<br>"itemgroup_sequence": 1,<br>"item_name": "CMTRT",<br>"coding_item_definition": "V0T000000004001",<br>"created_date": "2022-06-08T20:57:31Z",<br>"created_by": "Eric Emerton",<br>"messages": [<br>{<br>"id": "OPY00000000X002",<br>"activity": "open__v",<br>"message": "Please check this value, this is a query from coding add via API (3rd add to CM)",<br>"message_date": "2022-06-08T20:57:31Z",<br>"message_by": "Eric Emerton"<br>}<br>]<br>}<br>]<br>} |

## 11.3 Open Coding Queries

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#open-coding-query |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/coder/actions/openquery |
| Parameters | |

| Name | Array | Req/Opt | Notes |
|---|---|---|---|
| study_name | | Required | Name of the study |
| coding_request | queries | Required | The ID of the medical coding request. The query will be added against the item that the coding request applies to |
| message | queries | Required | The message of the query on add (255 character limit) |

**Notes**

- The action is used to add a **new** query to an item, specifically the item that is attached / referenced by a specific medical coding request. (e.g. AE verbatim or CM verbatim)
- **WARNING:** A locked form will not allow the add a new query, just at the EDC UI would disallow. Frozen forms will accept new queries added via the API.
- Additional abilities to open queries are available at:
  - **Open Queries - By Study Context**
  - **Open Queries - By Item ID** (different style of response)
  - **Open Queries - By Event ID** (different style of response)

### Examples

**Description / Request (URL)**

Multiple queries added

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "queries": [
            {
               "coding_request": "V0V000000004001",
                "message": "This is a query from coding
add via API"
            },
            {
               "coding_request": "V0V000000003002",
                "message": "This is a query from coding
add via API"
            }
            ]
  }
```

**Response**

```
{
    "responseStatus": "SUCCESS",
    "queries": [
        {
            "responseStatus": "SUCCESS",
            "id": "OPW00000000W001",
            "query_name": "VV-000234",
            "query_status": "open__v",
            "coding_request": "V0V000000004001"
        },
        {
            "responseStatus": "SUCCESS",
            "id": "OPW00000000V001",
            "query_name": "VV-000233",
            "query_status": "open__v",
            "coding_request": "V0V000000003002"
        }
    ]
```

| Description / Request (URL) | Response |
|---|---|
| | } |
| Multiple queries attempted, one succeeds, one fails | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"queries": [<br>{<br>"coding_request": "V0V000000004001",<br>"message": "This is a query from coding add via API"<br>},<br>{<br>"coding_request": "V0V00000000300233",<br>"message": "This is a query from coding add via API"<br>}<br>]<br>} | {<br>"responseStatus": "SUCCESS",<br>"queries": [<br>{<br>"responseStatus": "SUCCESS",<br>"id": "OPW00000000W001",<br>"query_name": "VV-000234",<br>"query_status": "open__v",<br>"coding_request": "V0V000000004001"<br>},<br>{<br>"responseStatus": "FAILURE",<br>"errorMessage": "[Coding Request] with id [V0V00000000300233] not found",<br>"coding_request": "V0V00000000300233"<br>}<br>]<br>} |

# 12 Jobs

## 12.1 Jobs / Data Extracts Overview

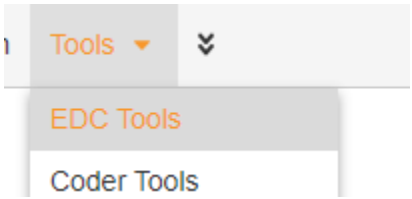
Currently the main full extract of the CDMS system (a study) is the **Study Data Extract**, itself a job inside the CDMS **EDC Tools** area -> **Jobs** tab. The job run yields a zip file that is cumulative of the entire study. This section will discuss the start / retrieve status / get file of this type of job, primarily, but also catalog other data export job types. There are additionally targeted data retrieval endpoints discussed in early sections of this document (**Retrieve Events / Forms** and **Retrieve Forms / Item Data**), but these are not meant for the retrieval of large amounts of data, i.e. across the entire study. Their purpose is for 'inspect before update' needs. Attempts to get wide slices of study data should work through the jobs, and their resulting zip files.

**NOTE:** There is no current incremental based export, it is a planned feature of CDMS, accompanied by API endpoints to drive.

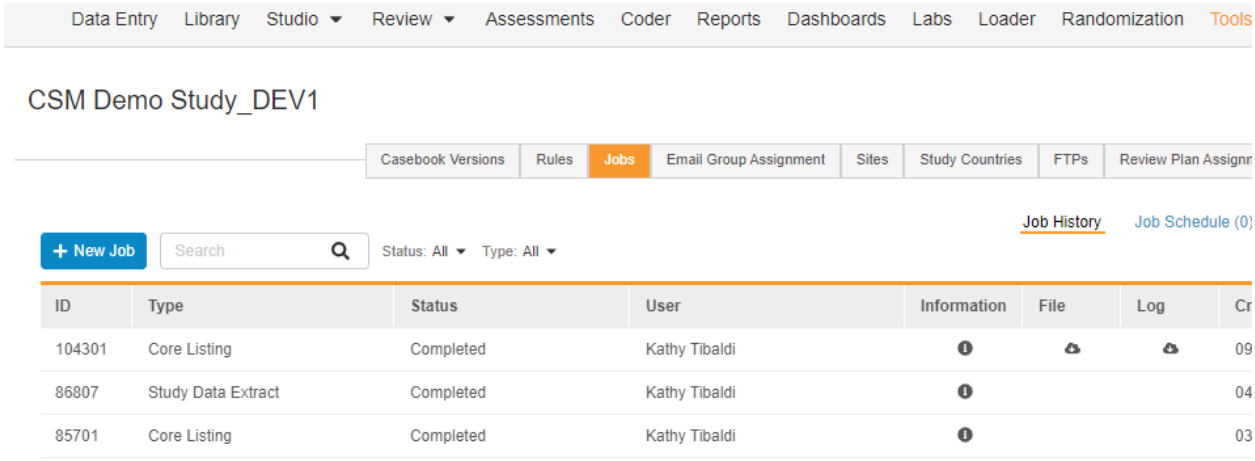
The current CDMS study job types for extracting data:

| Job Type | API Value | Notes |
|---|---|---|
| Study Data Export (SDE) | study_data_export__v | The full data export for a study. Includes the form data (e.g., AE, CM, DM) however that is configured, as well as operational datasets (SYS_* in name). The job **can** be run on a subset of datasets. Full details at: CDMS Help |
| Subject Progress Listing | subject_progress_listing__v | Operational listing of the study. Each row is a subject, with counts of progress of the subject. The job **cannot** be run in subsets. |
| Event Progress Listing | event_progress_listing__v | Operational listing of the study. Each row is an event, detailing that event status, date, and other information. The job **cannot** be run in subsets. |
| Form Progress Listing | form_progress_listing__v | Operational listing of the study. Each row is a form, detailing that form status, form properties, and other information. The job **cannot** be run in subsets. |
| Query Detail Listing | query_detail_listing__v | Operational listing of the study. Each row is a query in the study, detailing that status, cycle times, and other information. The job **cannot** be run in subsets. |
| Core Listings | core_listing__v | A predecessor of the SDE, only the form data listings, and of a different format than the SDE. The job **can** be run on a subset of data. |
| Data and Definitions | data_and_definition_export__v | A predecessor of the SDE, the data and definition file formats are different from the SDE. The job **cannot** be run in subsets. |

As background to study jobs, and in the context of the **Study Data Extract** (SDE), the tab for **EDC Tools** (for roles that have rights):

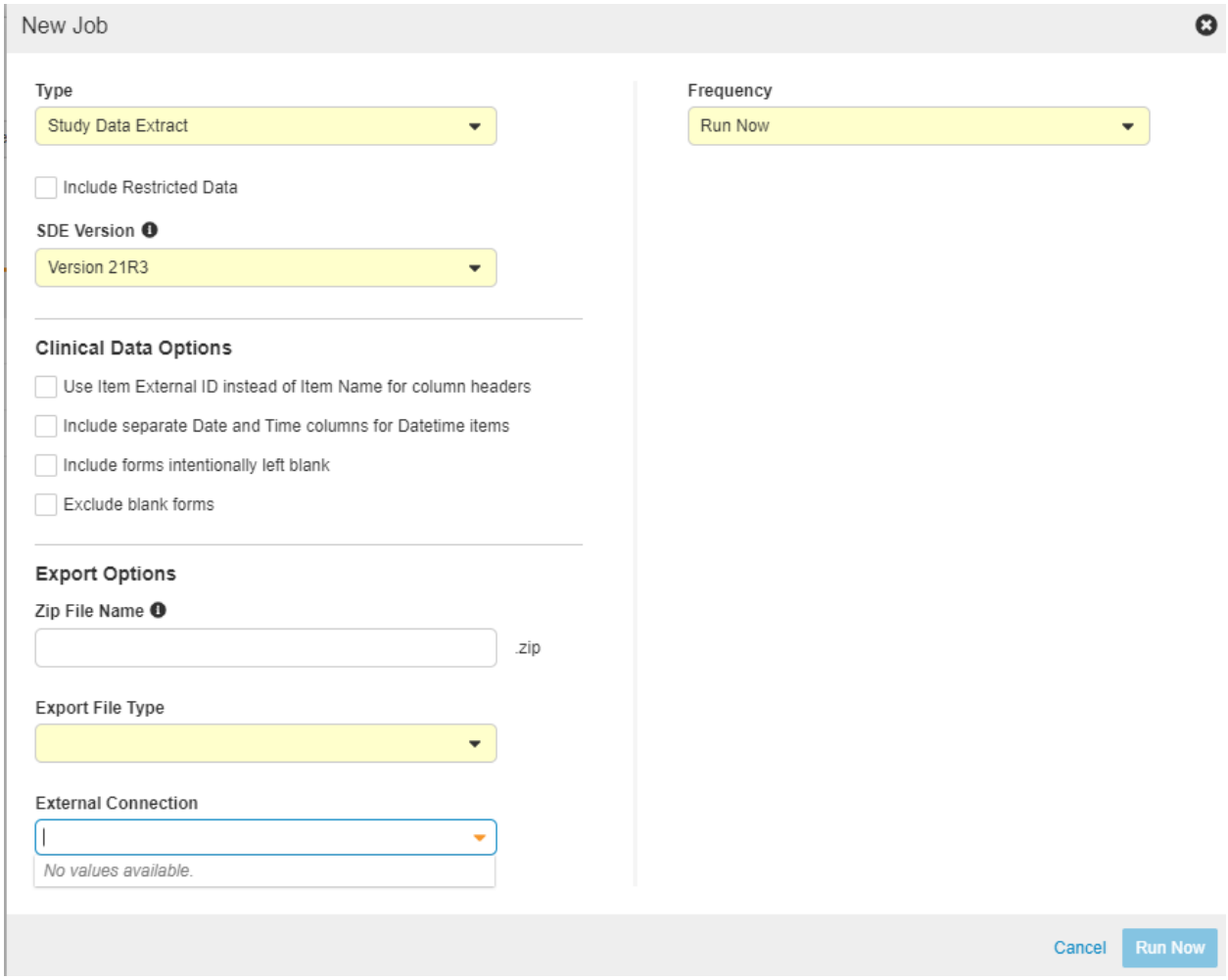



.. then selecting a study:

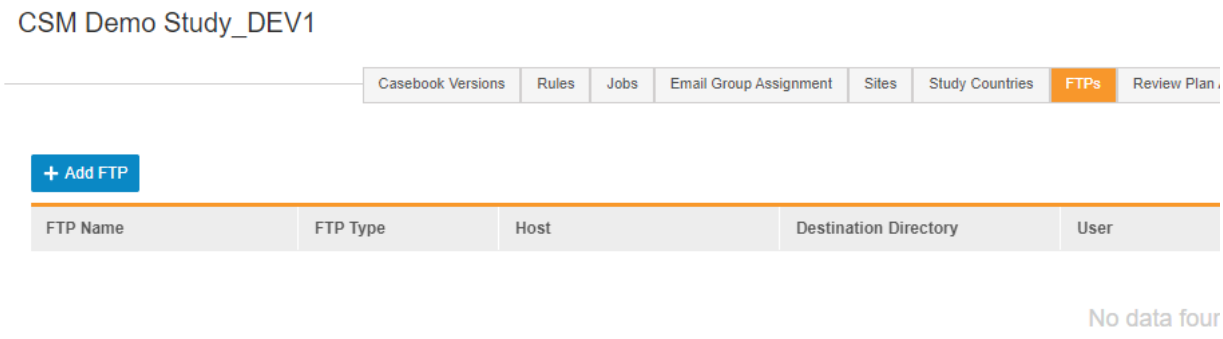



The Jobs tab of this section is where the Study Data Extract (SDE) and other export jobs can be run.

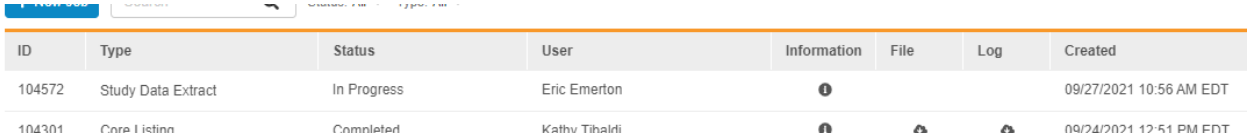
The job create screen:




… includes various options, each of which can be set if driving from the API start job endpoint. The **External Connection** - choices are a Vault FTP folder location, specifically set up at the sub tab (same screen):

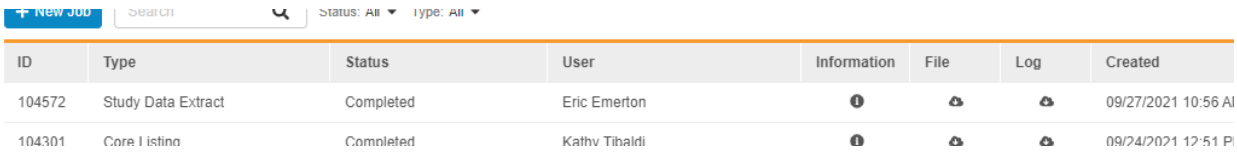



See this online help area for setup of the FTP connection: [https://cdmshelp.veeva.com/gr/study-administrators/connections/](https://cdmshelp.veeva.com/gr/study-administrators/connections/)
A running job is shown as:



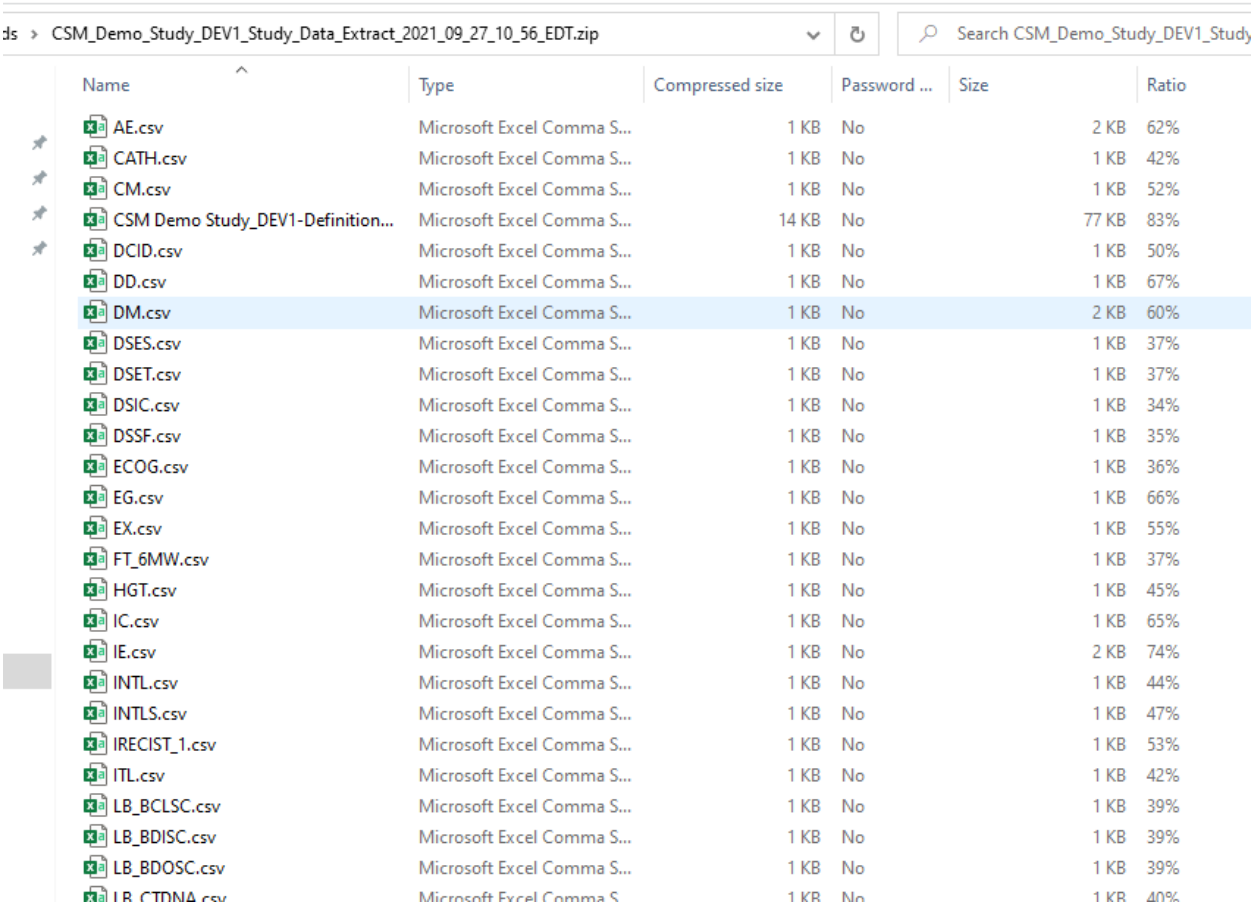
| ID | Type | Status | User | Information | File | Log | Created |
|---|---|---|---|---|---|---|---|
| 104572 | Study Data Extract | In Progress | Eric Emerton | | | | 09/27/2021 10:56 AM EDT |
| 104301 | Core Listing | Completed | Kathy Tibaldi | | | | 09/24/2021 12:51 PM EDT |

… and when finished:



| ID | Type | Status | User | Information | File | Log | Created |
|---|---|---|---|---|---|---|---|
| 104572 | Study Data Extract | Completed | Eric Emerton | | | | 09/27/2021 10:56 AI |
| 104301 | Core Listing | Completed | Kathy Tibaldi | | | | 09/24/2021 12:51 PI |

The File icon will download the resulting file:

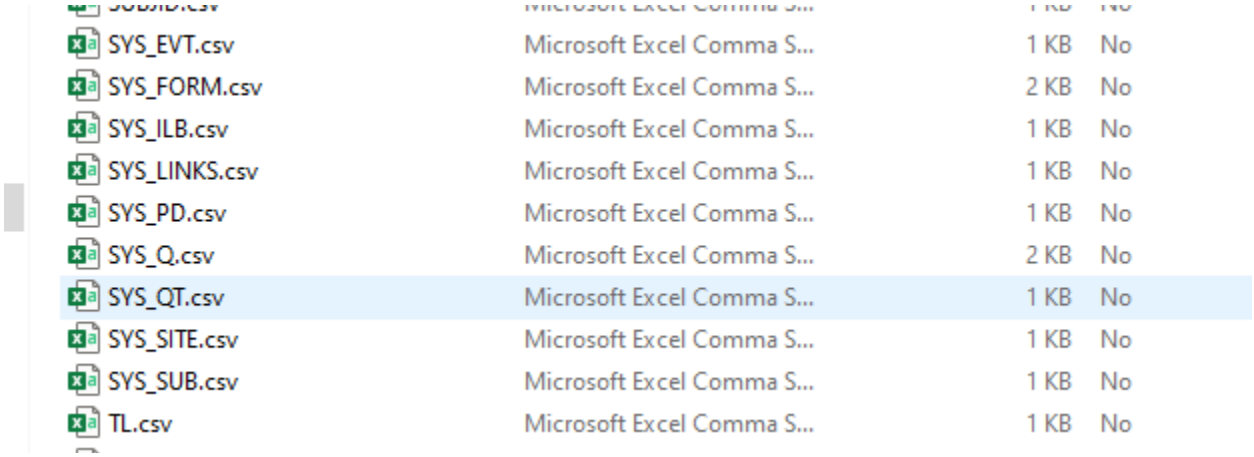


ds › CSM_Demo_Study_DEV1_Study_Data_Extract_2021_09_27_10_56_EDT.zip

Search CSM_Demo_Study_DEV1_Study

| Name | Type | Compressed size | Password ... | Size | Ratio |
|---|---|---|---|---|---|
| AE.csv | Microsoft Excel Comma S... | 1 KB | No | 2 KB | 62% |
| CATH.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 42% |
| CM.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 52% |
| CSM Demo Study_DEV1-Definition... | Microsoft Excel Comma S... | 14 KB | No | 77 KB | 83% |
| DCID.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 50% |
| DD.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 67% |
| DM.csv | Microsoft Excel Comma S... | 1 KB | No | 2 KB | 60% |
| DSES.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 37% |
| DSET.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 37% |
| DSIC.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 34% |
| DSSF.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 35% |
| ECOG.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 36% |
| EG.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 66% |
| EX.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 55% |
| FT_6MW.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 37% |
| HGT.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 45% |
| IC.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 65% |
| IE.csv | Microsoft Excel Comma S... | 1 KB | No | 2 KB | 74% |
| INTL.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 44% |
| INTLS.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 47% |
| IRECIST_1.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 53% |
| ITL.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 42% |
| LB_BCLSC.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 39% |
| LB_BDISC.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 39% |
| LB_BDOSC.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 39% |
| LB_CTDNA.csv | Microsoft Excel Comma S... | 1 KB | No | 1 KB | 40% |

The above zip file contains CSV, and often .sas7bdat files, as well as a definition file set to describe the export. It's the means of full export of all data in a study, not specific API calls. It does include various SYS_* named datasets, operational study data, that does not change per study design:



| | | | |
|---|---|---|---|
| SYS_EVT.csv | Microsoft Excel Comma S... | 1 KB | No |
| SYS_FORM.csv | Microsoft Excel Comma S... | 2 KB | No |
| SYS_ILB.csv | Microsoft Excel Comma S... | 1 KB | No |
| SYS_LINKS.csv | Microsoft Excel Comma S... | 1 KB | No |
| SYS_PD.csv | Microsoft Excel Comma S... | 1 KB | No |
| SYS_Q.csv | Microsoft Excel Comma S... | 2 KB | No |
| SYS_QT.csv | Microsoft Excel Comma S... | 1 KB | No |
| SYS_SITE.csv | Microsoft Excel Comma S... | 1 KB | No |
| SYS_SUB.csv | Microsoft Excel Comma S... | 1 KB | No |
| TL.csv | Microsoft Excel Comma S... | 1 KB | No |

More specific information on the format / files can be found at: https://cdmshelp.veeva.com/gr/study-administrators/study-data-extracts/

Retrieval options for the SDE zip file:

**Option A:** Vault FTP folder, through general FTPS connection. (Port 21, FTPS client software or other automation). The requires the setup of the FTP connector in EDC Tools, and selection at time of running the job (ad-hoc or scheduled)

**Option B:** Use of the Vault API for working with FTP folder, see also the next section of this document.

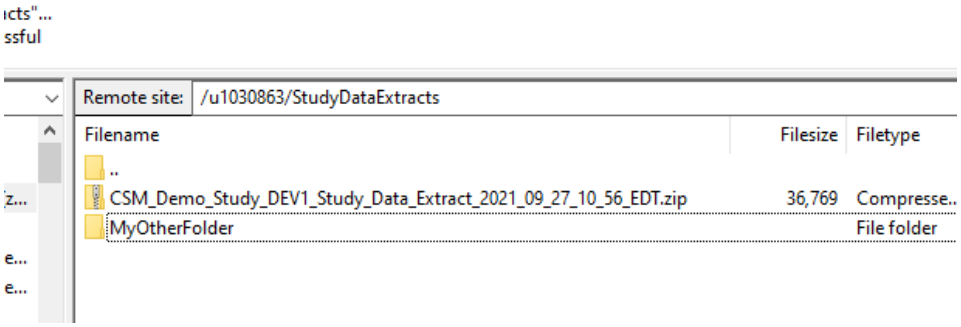
**Option C:** Use of the Get Job Output File endpoint, through the CDMS API endpoint.

## 12.2 Working with Vault FTP Folders Using Vault API

If the SDE or other similar extract jobs have a destination of the Veeva FTP location (also known as 'File Staging'), then one can use the SDE job zip delivered there through its otherwise recurring schedule. This alleviates the need to drive the start of the job / retrieve status, etc.

The main reference for Vault API, working with its FTP folders can be found here: https://developer.veevavault.com/api/22.2/#file-staging. The intent is to work with FTP folders, but via a REST API, instead of traditional FTPS / port 21 connections.

With the EDC Job configured to send the resulting SDE to an FTP, and viewing through an FTP tool like FIlezilla:




... we can use the API to retrieve files from that location (https://developer.veevavault.com/api/22.2/#list-items-at-a-path), using a REST call like:

.../api/v22.2/services/file_staging/items/**u1030863/StudyDataExtracts**?recursive=true

The return of the REST endpoint with recursive=true includes folders, and all files within:

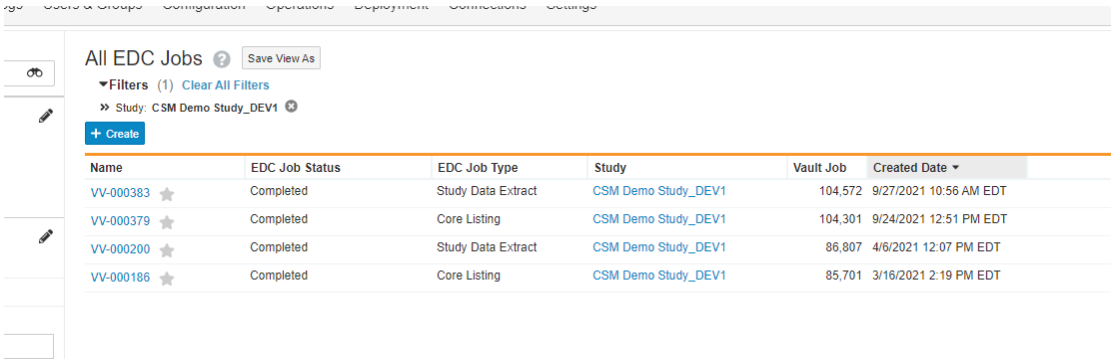

```
{
    "responseStatus": "SUCCESS",
    "data": [
        {
            "kind": "file",
            "path": "/u1030863/StudyDataExtracts/CSM_Demo_Study_DEV1_Study_Data_Extract_2021_09_27_10_56_EDT.zip",
            "name": "CSM_Demo_Study_DEV1_Study_Data_Extract_2021_09_27_10_56_EDT.zip",
            "size": 36769,
            "modified_date": "2021-09-27T15:31:04.000Z"
        },
        {
            "kind": "folder",
            "path": "/u1030863/StudyDataExtracts/MyOtherFolder",
            "name": "MyOtherFolder"
        },
        {
            "kind": "file",
            "path": "/u1030863/StudyDataExtracts/MyOtherFolder/CSM_Demo_Study_DEV1_Study_Data_Extract_2021_09_27_10_56_EDT.zip",
            "name": "CSM_Demo_Study_DEV1_Study_Data_Extract_2021_09_27_10_56_EDT.zip",
            "size": 36769,
            "modified_date": "2021-09-27T15:31:32.000Z"
        }
    ]
}
```


Then, use the REST endpoint to stream the file(s), see the reference: https://developer.veevavault.com/api/22.2/#get-item-content

## 12.3 Retrieve Recently Finished Jobs

If the API caller is **NOT** starting the SDE job, waiting, and retrieving that job's output zip file, then one needs to poll the study, its recently finished jobs. That is, one can schedule a nightly job through the **EDC Tools** UI, and an API caller can simply inspect those jobs to retrieve output files (zip) for these jobs.

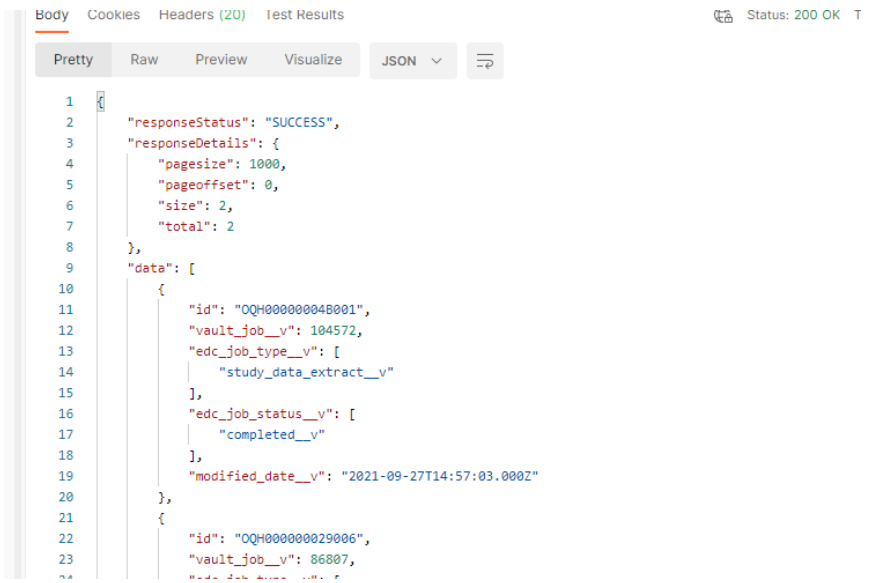
**IMPORTANT NOTE: THIS APPROACH IS TEMPORARY.** It will be replaced by a formal CDMS endpoint for getting study jobs, and across multiple studies. The records of the EDC Tools - > Jobs tab are in a vault object: (**edc_job__v**), essentially the **EDC Tools -> Jobs** tab




A Vault VQL statement can be run against a study and job type:

…/api/v22.2/query?q=SELECT id, vault_job__v,edc_job_type__v,edc_job_status__v, modified_date__v from edc_job__v where study__vr.name__v='**CSM Demo Study_DEV1**' and edc_job_status__v='**completed__v**' and edc_job_type__v='**study_data_extract__v**' order by modified_date__v desc

…yielding just the SDE type, and those now completed, and latest at beginning of the return:




The **vault_job__v** value in the return is the value to use in the endpoint discussed later in this document - **Retrieve Job Output File**

## 12.4 Start Study Job - SDE

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#start-job-sde |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/start_now |
| Parameters | |

| Name | Req/Opt | Default (when omitted) | Notes |
|---|---|---|---|
| study_name | Required | | Name of the study |
| job_type | Required | | For this job type, the value is always **study_data_extract__v** |
| include_restricted_data | Optional | false | Boolean - whether to include data that is otherwise on restricted forms. The API caller must have this permission to use **true**. |
| version | Required | | The version of the SDE, in vXX.Y format (XX = year, Y = 1, 2, or 3). That is the format at release of the SDE. More information on system datasets here in CDMS Help |
| export_file_type | Required | | Must be one of **CSV** or **SAS with XPT and CSV** |
| file_name | Optional | (see notes) | If omitted, Vault names the ZIP file with "{Study_Name}*Study_Data_Extract*{DATETIME}{TIMEZONE}". Some restrictions apply to the file naming |
| external_connections | Optional | | JSON array / list of the FTP connection names (setup in the study, via **EDC Tools**) to also forward the SDE zip file on to) |
| use_external_ids | Optional | false | Boolean - whether to use the item definition / form definition 'External ID' property, instead of design names |
| include_formilb | Optional | false | Boolean - include the Form ILB or not |
| split_datetime | Optional | false | Boolean - whether to split date time fields into 3 fields (main, date only, time only) |
| exclude_blank_forms | Optional | false | Boolean - whether to exclude rows/data from forms that are in status blank |
| include_rand_treatment | Optional | false | Boolean - only applicable if the study is using the Randomization module. Use of **true** includes the **SYS_RAND** dataset |
| all_clinical_datasets | Required | | Boolean - whether to include all non-system (CRF) datasets |
| clinical_datasets | Optional * | | If all_clinical_datasets = false, then this must be provided, a JSON array of dataset names to run |
| all_system_datasets | Required | | Boolean - whether to include all system datasets, e.g. SYS_FORM, SYS_SUB, etc. More information on system datasets: CDMS Help |
| system_datasets | Optional * | | Required if all_system_datasets = false. JSON array of SYS_* named datasets to run |
| include_custom_objects | Required | | Boolean - whether to include any custom objects. These objects must also be registered with the study |
| custom_objects | Optional * | | When include_custom_objects = true, the objects to include. These objects are setup in the object sde_customobject_config__v -> name__v value. That record refers to a custom object for export, i.e. records of that study. The custom object setup should be |

| | |
|---|---|
| | configured for deployments and have a study__c column |
| Notes | • When the study job starts, the vault job ID is returned to the caller. The caller then uses that to check job status, and eventually retrieve the file of the job<br>• For some system datasets (SYS_RAND, SYS_LAB*, SYS_ASM*), that module must be in use in the study, for the dataset to appear in the run. |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Common | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"request": {<br>"job_type": "study_data_extract__v",<br>"version": "v22.2",<br>"include_restricted_data": false,<br>"include_rand_treatment": false,<br>"export_file_type": "SAS with XPT and CSV",<br>"use_external_ids": false,<br>"include_formilb": false,<br>"split_datetime": false,<br>"exclude_blank_forms": false,<br>"all_clinical_datasets": true,<br>"all_system_datasets": true,<br>"include_custom_objects": false<br>}<br>} | {<br>"responseStatus": "SUCCESS",<br>"response": {<br>"job_type": "study_data_extract__v",<br>"job_id": 307484,<br>"created_by": "eric.emerton@eke.com",<br>"created_date": "2022-06-27T20:30:05Z",<br>"include_restricted_data": false,<br>"version": "v22.2",<br>"export_file_type": "SAS with XPT and CSV",<br>"file_name": null,<br>"external_connections": null,<br>"use_external_ids": false,<br>"include_formilb": false,<br>"split_datetime": false,<br>"exclude_blank_forms": false,<br>"include_rand_treatment": false,<br>"all_clinical_datasets": true,<br>"clinical_datasets": null,<br>"all_system_datasets": true,<br>"system_datasets": null,<br>"include_custom_objects": false,<br>"custom_objects": null<br>}<br>} |
| Bare required options (thus all the defaults) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"request": {<br>"job_type": "study_data_extract__v",<br>"version": "v22.2",<br>"export_file_type": "CSV",<br>"all_clinical_datasets": true,<br>"all_system_datasets": true,<br>"include_custom_objects": false | {<br>"responseStatus": "SUCCESS",<br>"response": {<br>"job_type": "study_data_extract__v",<br>"job_id": 307485,<br>"created_by": "eric.emerton@eke.com",<br>"created_date": "2022-06-27T20:32:01Z",<br>"include_restricted_data": false,<br>"version": "v22.2", |

| Description / Request (URL) | Response |
|---|---|
| }<br>} | "export_file_type": "CSV",<br>"file_name": null,<br>"external_connections": null,<br>"use_external_ids": false,<br>"include_formilb": false,<br>"split_datetime": false,<br>"exclude_blank_forms": false,<br>"include_rand_treatment": false,<br>"all_clinical_datasets": true,<br>"clinical_datasets": null,<br>"all_system_datasets": true,<br>"system_datasets": null,<br>"include_custom_objects": false,<br>"custom_objects": null<br>}<br>} |
| **Settings specific datasets, plus other options** | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"request": {<br>"job_type": "study_data_extract__v",<br>"version": "v22.2",<br>"include_restricted_data": true,<br>"include_rand_treatment": true,<br>"export_file_type": "SAS with XPT and CSV",<br>"file_name": "My_Zip_File",<br>"use_external_ids": true,<br>"include_formilb": true,<br>"split_datetime": true,<br>"exclude_blank_forms": true,<br>"all_clinical_datasets": false,<br>"clinical_datasets": [<br>"AE",<br>"DM"<br>],<br>"all_system_datasets": false,<br>"system_datasets": [<br>"SYS_FORM",<br>"SYS_Q"<br>],<br>"include_custom_objects": true,<br>"custom_objects": [<br>"doc_tracking__c",<br>"adjudication_tracking__c"<br>]<br>}<br>} | {<br>"responseStatus": "SUCCESS",<br>"response": {<br>"job_type": "study_data_extract__v",<br>"job_id": 307486,<br>"created_by": "eric.emerton@eke.com",<br>"created_date": "2022-06-27T20:47:35Z",<br>"include_restricted_data": true,<br>"include_rand_treatment": true,<br>"version": "v22.2",<br>"export_file_type": "SAS with XPT and CSV",<br>"file_name": "My_Zip_File",<br>"external_connections": null,<br>"use_external_ids": true,<br>"include_formilb": true,<br>"split_datetime": true,<br>"exclude_blank_forms": true,<br>"include_rand_treatment": false,<br>"all_clinical_datasets": false,<br>"clinical_datasets": [<br>"AE",<br>"DM"<br>],<br>"all_system_datasets": false,<br>"system_datasets": [<br>"SYS_Q",<br>"SYS_FORM"<br>],<br>"include_custom_objects": true,<br>"custom_objects": [ |

| Description / Request (URL) | Response |
|---|---|
| | "doc_tracking__c",<br>"adjudication_tracking__c"<br>]<br>}<br>} |
| Also Direct to the FTP Location(s) | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"request": {<br>"job_type": "study_data_extract__v",<br>"version": "v22.2",<br>"export_file_type": "CSV",<br>"all_clinical_datasets": true,<br>"all_system_datasets": true,<br>"include_custom_objects": false,<br>"external_connections": [<br>"My_FTP_Drop"<br>]<br>}<br>} | {<br>"responseStatus": "SUCCESS",<br>"response": {<br>"job_type": "study_data_extract__v",<br>"job_id": 307485,<br>"created_by": "eric.emerton@eke.com",<br>"created_date": "2022-06-27T20:32:01Z",<br>"include_restricted_data": false,<br>"version": "v22.2",<br>"export_file_type": "CSV",<br>"file_name": null,<br>"external_connections": [<br>"My_FTP_Drop"<br>],<br>"use_external_ids": false,<br>"include_formilb": false,<br>"split_datetime": false,<br>"exclude_blank_forms": false,<br>"include_rand_treatment": false,<br>"all_clinical_datasets": true,<br>"clinical_datasets": null,<br>"all_system_datasets": true,<br>"system_datasets": null,<br>"include_custom_objects": false,<br>"custom_objects": null<br>}<br>} |

## 12.5 Start Study Job - Subject Progress Listing

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#start-job-subject-progress |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/start_now |
| Parameters | |

| Name | Req/Opt | Default (when omitted) | Notes |
|---|---|---|---|
| study_name | Required | | Name of the study |
| job_type | Required | | For this job type, the value is always **subject_progress_listing__v** |
| include_restricted_data | Optional | false | Boolean - whether to include data that is otherwise on restricted forms. The API caller must have this permission to use **true**. |

**Notes**

- This listing generates data about subjects (row per subject) with various counts of their progress
- When the study job starts, the vault job ID is returned to the caller. The caller then uses that to check job status, and eventually retrieve the file of the job

### Examples

| Description / Request (URL) | Response |
|---|---|
| Run the Job | |

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "request": {
                "job_type": "subject_progress_listing__v",
                "include_restricted_data": false
              }
}
```

```
{
    "responseStatus": "SUCCESS",
    "response": {
        "job_type": "subject_progress_listing__v",
        "job_id": 307480,
        "created_by": "eric.emerton@eke.com",
        "created_date": "2022-06-27T20:26:11Z",
        "include_restricted_data": false
    }
}
```

## 12.6 Start Study Job - Event Progress Listing

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#start-job-event-progress |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/start_now |
| Parameters | |

| Name | Req/Opt | Default (when omitted) | Notes |
|---|---|---|---|
| study_name | Required | | Name of the study |
| job_type | Required | | For this job type, the value is always **event_progress_listing__v** |
| include_restricted_data | Optional | false | Boolean - whether to include data that is otherwise on restricted forms. The API caller must have this permission to use **true**. |

**Notes**

- This listing generates data about events (row per subject events) with current status and other metric information about events.
- When the study job starts, the vault job ID is returned to the caller. The caller then uses that to check job status, and eventually retrieve the file of the job

### Examples

**Description / Request (URL)**

Run the Job

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "request": {
                "job_type": "event_progress_listing__v",
                "include_restricted_data": false
              }
}
```

**Response**

```
{
    "responseStatus": "SUCCESS",
    "response": {
        "job_type": "event_progress_listing__v",
        "job_id": 307481,
        "created_by": "eric.emerton@eke.com",
        "created_date": "2022-06-27T20:26:39Z",
        "include_restricted_data": false
    }
}
```

## 12.7 Start Study Job - Form Progress Listing

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#start-job-form-progress |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/start_now |
| Parameters | |

| Name | Req/Opt | Default (when omitted) | Notes |
|---|---|---|---|
| study_name | Required | | Name of the study |
| job_type | Required | | For this job type, the value is always **form_progress_listing__v** |
| include_restricted_data | Optional | false | Boolean - whether to include data that is otherwise on restricted forms. The API caller must have this permission to use **true**. |
| include_item_counts | Optional | false | Whether to include an additional column about counts of items (data fields) on the forms |

**Notes**

- This listing generates data about events (row per subject forms) with current status and other metric information about each form.
- When the study job starts, the vault job ID is returned to the caller. The caller then uses that to check job status, and eventually retrieve the file of the job

### Examples

| Description / Request (URL) | Response |
|---|---|
| Run the Job | |

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "request": {
                "job_type": "form_progress_listing__v",
                "include_restricted_data": false,
                "include_item_counts": false
               }
}
```

```
{
    "responseStatus": "SUCCESS",
    "response": {
        "job_type": "form_progress_listing__v",
        "job_id": 307482,
        "created_by": "eric.emerton@eke.com",
        "created_date": "2022-06-27T20:27:00Z",
        "include_restricted_data": false,
        "include_item_counts": false
    }
}
```

## 12.8 Start Study Job - Query Detail Listing

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#start-job-query-detail-listing |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/start_now |
| Parameters | |

| Name | Req/Opt | Default (when omitted) | Notes |
|---|---|---|---|
| study_name | Required | | Name of the study |
| job_type | Required | | For this job type, the value is always **query_detail_listing__v** |
| include_restricted_data | Optional | false | Boolean - whether to include data that is otherwise on restricted forms. The API caller must have this permission to use **true**. |

**Notes**

- This listing generates data about queries (row per subject query) with current status and other metric information about queries.
- When the study job starts, the vault job ID is returned to the caller. The caller then uses that to check job status, and eventually retrieve the file of the job

### Examples

**Description / Request (URL)**

Run the Job

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "request": {
                "job_type": "query_detail_listing__v",
                "include_restricted_data": false
              }
}
```

**Response**

```
{
    "responseStatus": "SUCCESS",
    "response": {
        "job_type": "query_detail_listing__v",
        "job_id": 307483,
        "created_by": "eric.emerton@eke.com",
        "created_date": "2022-06-27T20:27:26Z",
        "include_restricted_data": false
    }
}
```

## 12.9 Start Study Job - Core Listings

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#start-job-core-listings |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/start_now |
| Parameters | |

| Name | Req/Opt | Notes |
|---|---|---|
| study_name | Required | Name of the study |
| job_type | Required | For this job type, the value is always **core_listing__v** |
| all_sites | Required | Boolean - whether to include data from all sites (or a subset) |
| sites | Required * | Required only when all_sites = false. JSON array of site numbers to include |
| all_forms | Required | Boolean - where to include all form data |
| forms | Required * | Required only when all_forms = false. JSON array of study design form names to include |

**Notes**

- This export is a predecessor to the SDE, of a different format. It only returns CSVs of form data
- When the study job starts, the vault job ID is returned to the caller. The caller then uses that to check job status, and eventually retrieve the file of the job

### Examples

**Core Listing - all sites**

Request:

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "request": {
                "job_type": "core_listing__v",
                "all_sites": true,
                "all_forms": true,
                "forms": []
              }
}
```

Response:

```
{
    "responseStatus": "SUCCESS",
    "response": {
        "job_type": "core_listing__v",
        "job_id": 305511,
        "created_by": "eric.emerton@eke.com",
        "created_date": "2022-06-17T19:39:51Z",
        "all_sites": true,
        "sites": null,
        "all_forms": true,
        "forms": null
    }
}
```

**Core Listing - all sites = false, then specific list**

Request:

```
{
```

Response:

```
{
```

| Description / Request (URL) | Response |
|---|---|
| "study_name": "ABCP-2022-01_DEV1",<br>"request": {<br>"job_type": "core_listing__v",<br>"all_sites": false,<br>"sites": ["101", "201", "102"],<br>"all_forms": true,<br>"forms": []<br>}<br>} | "responseStatus": "SUCCESS",<br>"response": {<br>"job_type": "core_listing__v",<br>"job_id": 305702,<br>"created_by": "eric.emerton@eke.com",<br>"created_date": "2022-06-17T19:55:14Z",<br>"all_sites": false,<br>"sites": [<br>"101",<br>"201",<br>"102"<br>],<br>"all_forms": true,<br>"forms": null<br>}<br>} |
| Core Listing - specific forms and sites | |
| {<br>"study_name": "ABCP-2022-01_DEV1",<br>"request": {<br>"job_type": "core_listing__v",<br>"all_sites": false,<br>"sites": ["101"],<br>"all_forms": false,<br>"forms": ["AE"]<br>}<br>} | {<br>"responseStatus": "SUCCESS",<br>"response": {<br>"job_type": "core_listing__v",<br>"job_id": 305706,<br>"created_by": "eric.emerton@eke.com",<br>"created_date": "2022-06-17T19:58:27Z",<br>"all_sites": false,<br>"sites": [<br>"101"<br>],<br>"all_forms": false,<br>"forms": [<br>"AE"<br>]<br>}<br>} |

## 12.10 Start Study Job - Data and Definitions Export

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#start-job-dde |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/start_now |
| Parameters | |

| Name | Req/Opt | Default (when omitted) | Notes |
|---|---|---|---|
| study_name | Required | | Name of the study |
| job_type | Required | | For this job type, the value is always **data_and_definition_export__v** |
| include_restricted_data | Optional | false | Boolean - whether to include data that is otherwise on restricted forms. The API caller must have this permission to use **true**. |

**Notes**

- This export is a predecessor to the SDE, of a different format. It only returns CSVs of form data
- When the study job starts, the vault job ID is returned to the caller. The caller then uses that to check job status, and eventually retrieve the file of the job

### Examples

**Description / Request (URL)**

Run the Job

```
{
    "study_name": "ABCP-2022-01_DEV1",
    "request": {
                "job_type": "data_and_definition_export__v",
                "include_restricted_data": false
              }
}
```

**Response**

```
{
    "responseStatus": "SUCCESS",
    "response": {
        "job_type": "data_and_definition_export__v",
        "job_id": 307579,
        "created_by": "eric.emerton@eke.com",
        "created_date": "2022-06-27T20:23:24Z",
        "include_restricted_data": false
    }
}
```

## 12.11 Retrieve Job Status

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#retrieve-job-status |
| Style | **GET** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/***{job_id}*** |
| Parameters | **Name** / **Req/Opt** / **Notes**<br>job_id / Required / The vault job ID. This value is embedded in the endpoint / URL. These job IDs are received on the start of a job, or via a query of study jobs. |
| Notes | • Use this endpoint to determine the current status of a CDMS study job, i.e. waiting for a job to finish.<br>• Only the job types mentioned in the **Jobs / Data Extracts Overview** section of this chapter can use this endpoint. |

### Possible Job Statuses

| Status | API Value | Notes |
|---|---|---|
| In Progress | in_progress__v | The job is currently running |
| Completed | completed__v | The overall job finished. WARNING: This does not mean the job was 100% successful. Once should parse / examine the logs and output files of the job to verify their content and indication of errors for that job type. |
| Failed | failed__v | The overall job failed. The finer details of the failure are likely in files / logs the job generates. |
| Canceled | canceled__v | The job was / is canceled |
| Canceling | canceling__v | An API call was done to cancel the job (see next section), and the job is still in the process of canceling. Once complete, the status will move to canceled__v |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Get status for a job | |
| ..api/{{version}}/app/cdm/jobs/305610 | {<br>"responseStatus": "SUCCESS",<br>"job_type": "study_data_extract__v",<br>"response": {<br>"job_id": 305610,<br>"study_name": "ABCP-2022-01_DEV1",<br>"status": "canceled__v",<br>"created_by": "eric.emerton@eke.com",<br>"created_date": "2022-06-17T18:32:15Z",<br>"last_modified_date": "2022-06-17T18:32:58Z"<br>}<br>} |

## 12.12 Cancel Job

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#cancel-job |
| Style | **POST** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/***{job_id}***/cancel_now |
| Parameters | **Name** / **Req/Opt** / **Notes**<br>job_id / Required / The vault job ID. This value is embedded in the endpoint / URL. These job IDs are received on the start of a job, or via a query of study jobs. |
| Notes | • Use this endpoint to cancel a CDMS study job that is currently running / in progress<br>• Once the job goes to completed (or canceled), an error will be returned on the attempt<br>• Only the job types mentioned in the **Jobs / Data Extracts Overview** section of this chapter can use this endpoint. |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Cancel successful | |
| ../api/{{version}}/app/cdm/jobs/305610/cancel_now | { "responseStatus": "SUCCESS", "response": { "job_type": "study_data_extract__v", "job_id": 305610, "study_name": "ABCP-2022-01_DEV1" } } |
| Cancel yields an error | |
| ../api/{{version}}/app/cdm/jobs/305610/cancel_now | { "responseStatus": "FAILURE", "errorMessage": "[Job] with status [canceled__v] is not cancellable" } |

## 12.13 Retrieve Job Output Log

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#retrieve-job-log |
| Style | **GET** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/***{job_id}***/file/log |
| Parameters | **Name** / **Req/Opt** / **Notes**<br>job_id / Required / The vault job ID. This value is embedded in the endpoint / URL. These job IDs are received on the start of a job, or via a query of study jobs. |
| Notes | • Use this endpoint to retrieve the Vault job log for a CDMS study job<br>• The output is text format and is high level information only. The job type will dictate more warnings / errors in log files it produces (contained in the output file, zip)<br>• Only the job types mentioned in the **Jobs / Data Extracts Overview** section of this chapter can use this endpoint. |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Job that has status completed | |
| ../api/{{version}}/app/cdm/jobs/305616/file/log | 2022-06-17T19:00:40.908Z Starting execution for job 305616<br>2022-06-17T19:00:44.972Z All chunks for job 305616 processed; beginning aggregation.<br>2022-06-17T19:00:45.956Z Successfully aggregated completed results for job 305616.<br>2022-06-17T19:00:46.908Z<br>2022-06-17T19:00:46.908Z Job Title: AsyncOperation<br>2022-06-17T19:00:46.909Z Job Type: ASYNC_OPERATION<br>2022-06-17T19:00:46.909Z Job Subtype: EVENT_PROGRESS_LISTING<br>2022-06-17T19:00:46.909Z Job Schedule Time: 2022-06-17T19:00:39.000Z<br>2022-06-17T19:00:46.909Z Job Queue Time: 2022-06-17T19:00:39.000Z<br>2022-06-17T19:00:46.909Z Job Execution Time: 2022-06-17T19:00:41.000Z<br>2022-06-17T19:00:46.909Z Job Finish Time: 2022-06-17T19:00:47.000Z<br>2022-06-17T19:00:46.909Z Job Completion Status: Success (COMPLETED_WITH_SUCCESS) |
| Job that was canceled | |
| ../api/{{version}}/app/cdm/jobs/305610/file/log | 2022-06-17T18:32:48.058Z All chunks for job 305610 processed; beginning aggregation.<br>2022-06-17T18:32:49.065Z Successfully aggregated completed results for job 305610.<br>2022-06-17T18:32:49.087Z Aggregation results are likely incomplete due to job 305610 having been canceled; 29 chunks were not processed due to cancellation.<br>2022-06-17T18:32:51.608Z<br>2022-06-17T18:32:51.609Z Job Title: AsyncOperation<br>2022-06-17T18:32:51.609Z Job Type: ASYNC_OPERATION<br>2022-06-17T18:32:51.609Z Job Subtype: STUDY_DATA_EXTRACT |

| Description / Request (URL) | Response |
| --- | --- |
| | 2022-06-17T18:32:51.609Z Job Schedule Time: 2022-06-17T18:32:16.000Z<br>2022-06-17T18:32:51.609Z Job Queue Time: 2022-06-17T18:32:16.000Z<br>2022-06-17T18:32:51.609Z Job Execution Time: 2022-06-17T18:32:20.000Z<br>2022-06-17T18:32:51.610Z Job Finish Time: 2022-06-17T18:32:52.000Z<br>2022-06-17T18:32:51.610Z Job Completion Status: Cancelled (CANCELLED)<br>2022-06-17T18:32:51.610Z Job Cancelled by: Eric Emerton (eric.emerton@eke.com)<br>2022-06-17T18:32:51.610Z Job Cancelled on: 2022-06-17T18:32:32.000Z |

## 12.14 Retrieve Job Output File

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#retrieve-job-output-file |
| Style | **GET** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/jobs/***{job_id}***/file/content |
| Parameters | **Name** / **Req/Opt** / **Notes**<br>job_id / Required / The vault job ID. This value is embedded in the endpoint / URL. These job IDs are received on the start of a job, or via a query of study jobs. |
| Notes | • Use this endpoint to retrieve the CDMS study job output file, e.g. a zip file of export content<br>• Only a job with status **completed__v** can run this endpoint<br>• **WARNING:** Depending on the job type, additional logs could be included in the output zip file, with indications of warnings and/or failures. Always read these files for true status of the job<br>• Only the job types mentioned in the **Jobs / Data Extracts Overview** section of this chapter can use this endpoint. |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Get of job output file | |
| `../api/{{version}}/app/cdm/jobs/305510/file/content` | (not shown, the stream of a zip file, typically) |

# 13 Users

## 13.1 Retrieve Users

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#retrieve-users |
| Style | |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/users |
| Parameters | |

| Name | Req/Opt | Notes |
|---|---|---|
| study_name | Optional | Name of the study. This can be omitted, and all users the API caller has rights to are returned |
| user_names | Optional | List specific users to return, the user name in <user>@<domain>.com format.. Separate multiple by comma |
| limit | Optional | Pagination API - different limit from the default 1000 can be used |
| offset | Optional | Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd |

Notes

- User to retrieve information about users in the vault as they relate to studies in the vault
- The top level of each user / entry returned is Vault account properties. An interior section of the return is itself a JSON array of study access that the user has rights for.
- The return **will** include all **vault owners** since those users are on every study.
- Refer to the **Return Parameter Notes** table after the examples for detailed information on each property.

### Examples

| Description / Request (URL) | Response |
|---|---|
| Many Users (multiple studies) | |
| ../app/cdm/users | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 17,<br>"total": 17<br>},<br>"vault_id": 1004329,<br>"users": [<br>{<br>"user_id": "164737",<br>"user_name": "alt_sys_user@eke.com", |

| Description / Request (URL) | Response |
|---|---|
| | "user_email": "eric.emerton@veeva.com",<br>"user_title": "",<br>"user_last_name": "Sys User",<br>"user_first_name": "Alt",<br>"company": "",<br>"federated_id": "",<br>"user_language": "en",<br>"user_locale": "en_US",<br>"user_timezone": "(GMT-05:00) Eastern Standard Time (America/New_York)",<br>"security_policy": "Basic",<br>"activation_date": "2020-09-29",<br>"created_date": "2020-09-29T17:26:41Z",<br>"last_modified_date": "2021-02-02T01:06:02Z",<br>"principal_investigator": false,<br>"active": true,<br>"vault_owner": false,<br>"all_studies_access": false,<br>"studies": [<br>{<br>"study_environment": "Labrinone-DM2_DEV1",<br>"study_role": "CDMS Lead Data Manager",<br>"study_access": "Enabled",<br>"all_sites_access": true,<br>"lms_training_status": "",<br>"ignore_lms_training_status": false<br>}<br>]<br>}<br><br>:<br>:<br>:<br>:<br>]<br>} |
| Users of one study | |
| ../app/cdm/users?study_name=ABCP-2022-01_DEV1 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 6,<br>"total": 6<br>},<br>"vault_id": 1004329,<br>"users": [<br>{ |

| Description / Request (URL) | Response |
| --- | --- |
| | "user_id": "96483",<br>"user_name": "dm_test_user@eke.com",<br>"user_email": "eric.emerton@veeva.com",<br>"user_title": "",<br>"user_last_name": "DM",<br>"user_first_name": "Test User",<br>"company": "",<br>"federated_id": "",<br>"user_language": "en",<br>"user_locale": "en_US",<br>"user_timezone": "(GMT-05:00) Eastern Standard Time (America/Indianapolis)",<br>"security_policy": "Basic",<br>"activation_date": "2022-06-28",<br>"created_date": "2018-07-23T19:28:20Z",<br>"last_modified_date": "2022-06-28T13:41:42Z",<br>"principal_investigator": false,<br>"active": true,<br>"vault_owner": false,<br>"all_studies_access": false,<br>"studies": [<br>{<br>"study_environment": "ABCP-2022-01_DEV1",<br>"study_role": "CDMS Lead Data Manager",<br>"study_access": "Enabled",<br>"all_sites_access": true,<br>"lms_training_status": "",<br>"ignore_lms_training_status": false<br>}<br>]<br>},<br>:<br>:<br>:<br>:<br>{<br>"user_id": "92677",<br>"user_name": "eric.emerton@eke.com",<br>"user_email": "eric.emerton@veeva.com",<br>"user_title": "",<br>"user_last_name": "Emerton",<br>"user_first_name": "Eric",<br>"company": "",<br>"federated_id": "",<br>"user_language": "en",<br>"user_locale": "en_US",<br>"user_timezone": "(GMT-05:00) Eastern Standard Time (America/New_York)",<br>"security_policy": "Basic",<br>"activation_date": "", |

| Description / Request (URL) | Response |
|---|---|
| | "created_date": "2018-02-12T21:07:25Z",<br>"last_modified_date": "2021-03-26T03:13:09Z",<br>"principal_investigator": false,<br>"active": true,<br>"vault_owner": true,<br>"all_studies_access": false<br>}<br>]<br>} |
| Single user who is a vault owner (thus access to all studies, studies array omitted) | |
| ../app/cdm/users?user_names=eric.emerton@eke.com | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 1,<br>"total": 1<br>},<br>"vault_id": 1004329,<br>"users": [<br>{<br>"user_id": "92677",<br>"user_name": "eric.emerton@eke.com",<br>"user_email": "eric.emerton@veeva.com",<br>"user_title": "",<br>"user_last_name": "Emerton",<br>"user_first_name": "Eric",<br>"company": "",<br>"federated_id": "",<br>"user_language": "en",<br>"user_locale": "en_US",<br>"user_timezone": "(GMT-05:00) Eastern Standard Time (America/New_York)",<br>"security_policy": "Basic",<br>"activation_date": "",<br>"created_date": "2018-02-12T21:07:25Z",<br>"last_modified_date": "2021-03-26T03:13:09Z",<br>"principal_investigator": false,<br>"active": true,<br>"vault_owner": true,<br>"all_studies_access": false<br>}<br>]<br>} |
| Single user with multiple study access | |
| ../app/cdm/users?user_names=mia.st | { |

| Description / Request (URL) | Response |
| --- | --- |
| one@eke.com | "responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 1,<br>"total": 1<br>},<br>"vault_id": 1004329,<br>"users": [<br>{<br>"user_id": "92876",<br>"user_name": "mia.stone@eke.com",<br>"user_email": "eric.emerton@veeva.com",<br>"user_title": "",<br>"user_last_name": "Stone",<br>"user_first_name": "Mia",<br>"company": "",<br>"federated_id": "",<br>"user_language": "en",<br>"user_locale": "en_US",<br>"user_timezone": "(GMT-05:00) Eastern Standard Time (America/New_York)",<br>"security_policy": "Basic",<br>"activation_date": "",<br>"created_date": "2018-02-12T23:21:15Z",<br>"last_modified_date": "2021-02-02T01:06:02Z",<br>"principal investigator": false,<br>"active": true,<br>"vault_owner": false,<br>"all_studies_access": false,<br>"studies": [<br>{<br>"study_environment": "Labrinone1_DEV1",<br>"study_role": "CDMS Clinical Research Associate",<br>"study_access": "Enabled",<br>"site_access": "",<br>"country_access": "United States",<br>"lms_training_status": "",<br>"ignore_lms_training_status": false<br>},<br>{<br>"study_environment": "Labrinone2_DEV1",<br>"study_role": "CDMS Lead Data Manager",<br>"study_access": "Enabled",<br>"site_access": "US-103,CA-101,CA-102,CA-103,UK-102,UK-103"<br>"country_access": "",<br>"lms_training_status": "",<br>"ignore_lms_training_status": false<br>}, |

| Description / Request (URL) | Response |
|---|---|
| | {<br>"study_environment": "Labrinone3_DEV1",<br>"study_role": "CDMS Lead Data Manager",<br>"study_access": "Enabled",<br>"site_access": ""<br>"country_access": "",<br>"lms_training_status": "",<br>"ignore_lms_training_status": false<br>},<br>{<br>"study_environment": "Labrinone4_DEV1",<br>"study_role": "CDMS Lead Data Manager",<br>"study_access": "Enabled",<br>"all_sites_access": true,<br>"lms_training_status": "",<br>"ignore_lms_training_status": false<br>}<br>]<br>}<br>]<br>} |

### Return Parameter Notes

| Location | Type | Notes |
|---|---|---|
| responseStatus | String | |
| responseDetails | Section | |
| responseStatus/limit | Integer | When paging involved, the limit of each return |
| responseStatus/offset | Integer | When paging involved, the offset of the current return |
| responseStatus/size | Integer | The size of the current return |
| responseStatus/total | Integer | The total (outside of any paging) of the query |
| vault_id | Integer | The Vault ID |
| users | Array | An entry per user of the return |
| users/user_id | String | Value is across all vaults / domains |
| users/user_name | String | E.g. eric.emerton@eke.com |
| users/user_email | String | |
| users/user_title | String | |
| users/user_last_name | String | |

| Location | Type | Notes |
|---|---|---|
| users/user_first_name | String | |
| users/company | String | |
| users/federated_id | String | This value is the user name in a system the user is using SSO for to login to the vault. The security policy must be set appropriately for this value to line up. (multiple SSO policies can reside in one vault) |
| users/user_language | String | E.g. en, fr, de, etc. |
| users/user_locale | String | E.g. en_US, en_AU, etc. |
| users/user_timezone | String | Full label of the user's timezone |
| users/security_policy | String | The security policy name value |
| users/activation_date | String (as Date) | If there is an activation date, shown as yyyy-MM-dd format. Otherwise **omitted** |
| users/created_date | String | Format yyyy-MM-ddThh:ss:mmZ, UTC date time |
| users/last_modified_date | String | Format yyyy-MM-ddThh:ss:mmZ, UTC date time |
| users/principal_investigator | Boolean | When choosing 'Add as Principal Investigator' from the UI, a user can also be a PI for tagging to a specific site. |
| users/active | Boolean | I.e. if the user is active in the vault (no specific study) |
| users/vault_owner | Boolean | If the user is a vault owner, they have access to all studies (need no specific study access). These users returned will not have a user/studies JSON array of entries |
| users/all_studies_access | Boolean | When it is true access to all existing and future studies of the vault |
| users/studies | Array node | Vault owner users have this **omitted**, i.e. since access to all studies is implied |
| users/studies/study_environment | String | The Study Instance label, e.g. ABC-2022-01_DEV1 |
| users/studies/study_role | String | The study role name, i.e. 'for that study' |
| users/studies/study_access | String | Values of **Enabled** or **Disabled** |
| users/studies/all_sites_access | Boolean | When **true,** the user has access to all sites. If **false**, **omitted** from return. |
| users/studies/site_access | String | Comma list of sites the user has access to |
| users/studies/country_access | String | Comma list of study counties for the user, if not specific site access. That is, the sites of those countries. |
| users/studies/lms_training_status | String | Values are empty, **Trained** or **Not Trained** |
| users/studies/ignore_lms_training_status | Boolean | When this value is **true**, the LMS training status on that study is ignored. That is, the user is allowed into the study per this flag |

## 13.2 Upload Users

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#upload-users |
| Style | **PUT** |
| Endpoint | <your Vault URL>/api/v22.2/app/cdm/users |
| Parameters | |

| Name | Req/Opt | Notes |
|---|---|---|
| import_file | Required | CSV file. Includes header line, and a row per user to action on. One line can be multiple studies (e.g UAT vault). See CDMS Help for full details on the structure of the file, plus latest template |
| append_study_country_access | Optional | |

**Notes**

- This endpoint is used to create or update users
- Each row is a user to make updates (or create)
- See CDMS Help for full details on the structure of the file, plus latest template
- For deactivation at a study level, the endpoint is used with specific **'Disabled'** value in the Study Access column.
- Vault or domain activation or deactivation - see the next few sections
- The CSV attempted is checked for:
  - Correct column header names
  - Values in each cell that comply with checks
  - If **ANY** of the rows fail, the return response will enumerate those rows with issues. No job starts in such cases
- **WARNING**: The CSV template can change from release to release as new properties are introduced. If some of the new properties are required, this could upend existing built integrations. The API version does **NOT** refer back and use older CSV templates. Consult the documentation and experiment in pre release vaults on coming releases
- Once a job successful starts, the job ID is returned, and this endpoint:

  <your Vault URL>/api/v22.2/services/jobs/***{job_id}***

  .. can be used to check for the status of the job. Once the job completes, consider using **Retrieve Users** to verify the state of each of the users created and/or updated.

## Examples

| Description / Request (URL) | Response |
|---|---|
| CSV attempted is valid, job starts | |
| `https://{{vaultDNS}}/api/v22.2/app/cdm/users?append_study_country_access=false`<br><br>(with content/type = text/csv and file passed in **import_file** parameter) | {<br>"responseStatus": "SUCCESS",<br>"response": {<br>"job_type": "EDC Import",<br>"job_id": 446208,<br>"status": "Queued",<br>"created_by": "user.admin@username.com",<br>"created_date": "2021-08-16T12:14:42Z" |

| Description / Request (URL) | Response |
|---|---|
| | }<br>} |
| Attempt where there are issues in the CSV | |
| https://{{vaultDNS}}/api/v22.2/app/cdm/users?append_study_country_access=false<br><br>(with content/type = text/csv and file passed in **import_file** parameter) | {<br>"responseStatus": "FAILURE",<br>"responseDetails": {<br>"rows": 1,<br>"ready": 0,<br>"warning": 0,<br>"error": 1,<br>"limit": 100<br>},<br>"errors": [<br>{<br>"errorMessage": "Error: Ignore LMS Status [Maybe] does not exist, Study Access [Enabledd] does not exist, Role [CDMS Principal Investigatorr] does not exist",<br>"User Name": "test_api_1@eke.com",<br>"Study Role": "CDMS Principal Investigatorr",<br>"Ignore LMS Status": "Maybe",<br>"Study Access": "Enabledd"<br>}<br>]<br>} |

## 13.3 Inactivate User – Vault or Domain Level

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#inactivate-user |
| Style | **DELETE** |
| Endpoint | <your Vault URL>/api/v22.2/objects/users/***{user_id}*** |
| Parameters | **Name** / **Req/Opt** / **Notes**<br>user_id / Required / The Vault user ID.<br>domain / Optional / URL parameter, true/false. When omitted the default is **false**. |
| Notes | • Use this endpoint **ONLY** to inactivate a user account, vault or domain wise<br>• For the deactivation of just a **specific** study, use the **Upload Users** endpoint, but with a line for the user indicating a **Disabled** status.<br>• **WARNING**: This endpoint with **domain=true** will inactivate the user across **ALL** vaults they otherwise have access into. The use of domain=false will inactivate the user in just the vault targeted. (all studies in that vault).<br>• **TIP:** Later a user can be reactivated. Consider leaving a user's study access intact, so that a simple reactivation later can return the user to all appropriate study access rights. |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Inactivate successful - just the target vault | |
| ..api/v22.2/objects/users/1234567 | { "responseStatus": "SUCCESS", "id": 1234567 } |
| Inactivate successful - **domain** level (ALL vaults on that domain) | |
| ..api/v22.2/objects/users/1234567?domain=true | { "responseStatus": "SUCCESS", "id": 1234567 } |
| Bad User ID | |
| ..api/v22.2/objects/users/1234599 | { "responseStatus": "FAILURE", "errors": [ { "type": "INVALID_DATA", "message": "User [1234599] not found" |

| Description / Request (URL) | Response |
| --- | --- |
| | }<br>]<br>} |

## 13.4 Activate User – Domain Level

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#activate-users<br><br>(from the main vault endpoint for update a user, at vault level) - https://developer.veevavault.com/api/22.2/#update-users ) |
| Style | **PUT** |
| Endpoint | <your Vault URL>/api/v22.2/objects/users/***{user_id}*** |
| Parameters | **Name** / **Req/Opt** / **Notes**<br>user_id / Required / The Vault user ID, as part of the endpoint<br>domain_active__v / Required / **Encoded** parameter on the action only. Value = **true** |
| Notes | • Use this endpoint **ONLY** to activate a user account that is currently inactive on the domain (all vaults in the domain)<br>• Only a domain admin can perform this action. The API caller must have that access on the domain to perform.<br>• This action must be performed first, prior to more actions to activate in specific vaults, then more to activate specific studies (where necessary) |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Active on domain successful | |
| ..api/v22.2/objects/users/1234567<br><br>(with encoded parameter parameter **domain_active__v = true**) | {<br>"responseStatus": "SUCCESS",<br>"id": 1234567<br>} |
| Bad User ID | |
| ..api/v22.2/objects/users/1234599 | {<br>"responseStatus": "FAILURE",<br>"errors": [<br>{<br>"type": "INVALID_DATA",<br>"message": "User [1234599] not found"<br>}<br>]<br>} |

## 13.5 Activate User – Vault Level

| | |
|---|---|
| Help Main | https://developer-cdms.veevavault.com/api/22.2/#activate-users<br><br>(from the main vault endpoint for update a user, at vault level) - https://developer.veevavault.com/api/22.2/#update-vault-membership) |
| Style | **PUT** |
| Endpoint | <your Vault URL>/api/v22.2/objects/users/***{user_id}***/vault_membership/***{vault_id}*** |
| Parameters | **Name** / **Req/Opt** / **Notes**<br>user_id / Required / The Vault user ID, as part of the endpoint<br>vault_id / Required / The Vault ID to be activated, as part of the endpoint<br>active__v / Required / **Encoded** parameter on the action only. Value = **true** |
| Notes | • Use this endpoint **ONLY** to activate a user account that is currently inactive on a specific vault, and the account is to be activated on that vault only.<br>• This action must be performed first, prior to specific study activations (where necessary) |

### Examples

| Description / Request (URL) | Response |
|---|---|
| Active on vault successful | |
| ..api/v22.2/objects/users/1234567/vault_membership/10043299<br><br>(with encoded parameter parameter **active__v = true**) | {<br>"responseStatus": "SUCCESS"<br>} |
| Bad User ID | |
| ..api/v22.2/objects/users/1234599/vault_membership/10043299 | {<br>"responseStatus": "FAILURE",<br>"errors": [<br>{<br>"type": "INVALID_DATA",<br>"message": "User [1234599] not found"<br>}<br>]<br>} |

# 14 Study Design

## 14.1 Design Overview

The study design endpoints started at release **22R2**, and additional will be released going forward. The primary use case is for programmatic parsing of a study's design, similar to CDISC ODM **xml** files. The plan for Vault CDMS is to release APIs that can retrieve that similar representation of a study's design, by casebook version (file per), but initially in JSON file format (Vault platform's primary format).

Studies are designed using design level components that mirror the data levels:

- **Casebook Version Definition**
- .. then.. -> **Event Group Definition**
- .. then.. -> **Event Definition**
- .. then.. -> **Form Definition**
- .. then.. -> **Item Group Definition**
- .. then.. -> **Item Definition**
- …then.. -> Codelist / Unit Codelist Definitions (used by dropdown Item definitions)

Each casebook version yields a schedule of these components, i.e. what is in use at the various levels. Properties like 'repeating', labels, short labels, etc. (seen in the Studio SDS) are included at all levels.

Currently, the Vault CDMS Studio module allows for export of a Study Design Specification (SDS), plus annotated workbooks PDFs - by casebook version. The SDS is an Excel workbook and could be parsed for all the same design information, but programmatic retrieval of these is not possible, and studies get amendments / new casebook versions often. Further, the programmatic parsing of Excel workbooks can be risky as compared to the same information in JSON (or XML) format.

### 14.2 Retrieve Casebook Versions

Similar to the online help, this is noted here only as a pointer to the existing **Studies ->** **Retrieve Studies** endpoint. As of 22R2 release, it includes the casebook definitions in the study..

## 14.3 Retrieve Coding Definitions

<table>
<tr><td>Help Main</td><td>https://developer-cdms.veevavault.com/api/22.2/#retrieve-coding-form-definitions</td></tr>
<tr><td>Style</td><td>GET</td></tr>
<tr><td>Endpoint</td><td><your Vault URL>/api/v22.2/app/cdm/coder/forms</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Req/Opt</th><th>Notes</th></tr>
<tr><td>study_name</td><td>Optional</td><td>Name of the study. This can be omitted, and all users the API caller has rights to are returned</td></tr>
<tr><td>limit</td><td>Optional</td><td>Pagination API - different limit from the default 1000 can be used</td></tr>
<tr><td>offset</td><td>Optional</td><td>Pagination API - the offset of the current page, if the response is to cross multiple pages. Example: 1750 records in query, 0 thru 999 (zero based index) in 1st 'page', then 1000 to 1749 in the 2nd</td></tr>
</table>
</td></tr>
<tr><td>Notes</td><td>
<ul>
<li>Used to retrieve the list of medical coding forms, i.e. those tagged to use Vault CDMS Coder. Medical coding requests are added on the submit of such forms</li>
<li>The coding definition includes the item definition being coded (e.g. AE verbatim, DM drug name), plus other properties from the form (if the study design maps them)</li>
<li>With this information, one can then use the response information to obtain information from the Medical Coding requests and queries of the study, that is, ‘of’ that definition.</li>
<li>WARNING: Regarding parameter names from definition response vs. usage in other endpoints:
<ul>
<li>The value of the definition response (this endpoint) for:<br><code>coding_item_definition_name</code><br>.. is in turn used as the parameter into the Medical Coding -> Retrieve Coding Requests endpoint to get requests of that definition, specifically, its parameter:<br><code>coding_item_definition</code></li>
<li>The value of this definition response (this endpoint) for:<br><code>coding_item_definition_id</code><br>… is in turn used to as the parameter into the Medical Coding -> Retrieve Coding Queries endpoint to get queries added against medical coding requests of that definition, specifically:<br><code>coding_item_definition</code></li>
<li>A future release is likely to change these parameter names (retrieve coding requests / queries) to be consistent, but still honor old/new naming.</li>
</ul>
</li>
</ul>
</td></tr>
</table>

## Examples

| Description / Request (URL) | Response |
|---|---|
| Get Coding Form Definitions of a Study | |
| …/api/v22.2/app/cdm/coder/forms?study_name=ABCP-2022-01_DEV1 | {<br>"responseStatus": "SUCCESS",<br>"responseDetails": {<br>"limit": 1000,<br>"offset": 0,<br>"size": 3,<br>"total": 3<br>},<br>"forms": [<br>{<br>"form_name": "AE",<br>"form_type": "ae__v",<br>"form_definition_status": "active__v",<br>"form_coding_status": "active__v",<br>"dictionary_definition": "MedDRA",<br>"coding_item_definition_id": "V0T000000002001",<br>"coding_item_definition_name": "VV-000004"<br>},<br>{<br>"form_name": "MULTI_CODING",<br>"form_type": "other__v",<br>"form_type_other_label": "My Other Label Here",<br>"form_definition_status": "active__v",<br>"form_coding_status": "active__v",<br>"dictionary_definition": "MedDRA",<br>"coding_item_definition_id": "V0T000000003001",<br>"coding_item_definition_name": "VV-000005"<br>},<br>{<br>"form_name": "Concomitant-Medication",<br>"form_type": "conmed__v",<br>"form_definition_status": "active__v",<br>"form_coding_status": "active__v",<br>"dictionary_definition": "WHODrug C3",<br>"coding_item_definition_id": "V0T000000004001",<br>"coding_item_definition_name": "VV-000006"<br>}<br>]<br>} |

## 15 Best Practices / Tips

| Area | Best Practice |
|---|---|
| **Error Detection** | Watch the 'inner' vs. 'outer' success/fail statuses in API response. The Vault API is designed such that most API endpoints all multiple actions of that type in the same call. As such, the status of each entry in the request is embedded at the inner level. The outer **responseStatus** is largely an indication that 'yes, you reached the server, and it performed an action'. The example below is the attempt to add two event groups, where one succeeds, one fails. Note the inner **responseStatus** for each, with overall outer **responseStatus**<br><br>`{`<br>`"responseStatus": "SUCCESS",`<br>`"eventgroups": [`<br>`{`<br>`"responseStatus": "SUCCESS",`<br>`"study_country": "United States",`<br>`"site": "101",`<br>`"subject": "101-001",`<br>`"eventgroup_name": "egIRT_INFO",`<br>`"eventgroup_sequence": 1`<br>`},`<br>`{`<br>`"responseStatus": "FAILURE",`<br>`"errorMessage": "[Subject] with name [101-021] not found"`<br>`"study_country": "United States",`<br>`"site": "101",`<br>`"subject": "101-021",`<br>`"eventgroup_name": "egIRT_INFO"`<br>`}`<br>`]`<br>`}` |
| **Error Detection** | If creating an integration into Vault CDMS, what is your error detection overall strategy?<br>● Do you examine each response for success before moving on to a sequential / next call<br>● What sort of alerting / process kicks off when there is an error? How do you avoid things falling through the cracks?<br>● What about the mistake / update scenarios? What happens when your source data changes, mistake or otherwise, how do you push the data in, where an update might have a different course of action from the initial add? |

| Area | Best Practice |
|---|---|
| **Study Design** | For an inbound integration to Vault CDMS:<br>● If possible, **avoid** form designs that involve **both** user entered data and integrated data.<br>● If the data is semi-consistent from the upstream system from Vault CDMS, consider a template of forms, 100% read only that can be reused study to study. |
| **Study Design** | A repeating visit in Vault CDMS increments the **eventgroup_sequence** values (1, 2, .. etc). Although there is an event_sequence property, current system design yields that as a **1**, always. The event group level is used for both single repeating events (e.g. Unscheduled), or series of events (e.g. Day 1, Day 8, Day 15.. but by Cycle) |
| **API Account** | Try to follow a path for the API account with a Vault Security Policy with setting 'No Password Expiration'. In this way, a recurring integration is not at risk of having an expired password upend the push or pull of important information. |
| **API Account** | An API user is no different from any other study user. Typically, users must go through / pass a training before their study access is truly enabled. For an API / system user, ensure to check the **Ignore LMS Status, Assume Trained** attribute on the account (typically). Otherwise the API user won't find the study (or studies) intended. The action is not necessary if the specific Vault is not activated to required LMS training before study access is granted. |
| **Inspecting Data** | Strive to inspect the existing forms / events before attempting an action to push data into the study. Does the subject even exist? Does the target area of data (form, event group, etc) exist? |
| **Inspecting Data** | If a form is submitted, it must first be opened for edit via the **Edit Submitted Form** API endpoint, Edit Form (later section), before proceeding with the update of data. Currently there is no parameter to allow for conditional open for edit, during the call to update data on a form, not a parameter to additionally submit the form once data is updated/added. |
| **Create vs Upsert** | Various API endpoints started as specific create actions, with later **PUT** versions performing an upsert of the entity (event group, form, item group, etc). The upsert option is a better option, since specific sequences can be indicated at the various design levels, with the result being either an add (if does not exist) or a skipped action. Update '2nd time in' scenarios benefit from this approach as the same sequence of API calls can typically be used. |
| **Sequences** | The various design levels of a study - event group, form, item group - can repeat, or not repeat. When the do not repeat the Create / Upsert endpoints do not require the specific indication of, say:<br><br>:<br>"eventgroup._sequence": 1<br>:<br><br>in the request body. That said, consider always including these regardless, as it makes for a more consistent request/response match |

<table>
<tr><th>Area</th><th>Best Practice</th></tr>
<tr><td>Locked / Frozen</td><td>Various levels of locked and frozen statuses in turn affect ability to push data into Vault CDMS:<br><br>
<table>
<tr><th>Area</th><th>Status</th><th>Add/Update Data</th><th>Add/Update Queries</th></tr>
<tr><td>Study</td><td>Locked</td><td>Disallowed</td><td>Disallowed</td></tr>
<tr><td>Site</td><td>Locked</td><td>Disallowed</td><td>Disallowed</td></tr>
<tr><td>Subject</td><td>Locked</td><td>Disallowed</td><td>Disallowed</td></tr>
<tr><td>Event</td><td>Locked</td><td>Disallowed</td><td>Disallowed</td></tr>
<tr><td>Event</td><td>Frozen</td><td>Disallowed (event date changes)</td><td>Allowed (on event date)</td></tr>
<tr><td>Form</td><td>Locked</td><td>Disallowed</td><td>Disallowed</td></tr>
<tr><td>Form</td><td>Frozen</td><td>Disallowed</td><td>Allowed</td></tr>
</table>
<br>Frozen and Locked are not automatable through the API, by design. These were/are human decisions to stop actions from happening.</td></tr>
<tr><td>Change Reasons</td><td>For the update of a form data, a change reason is only used when there is an update/change to existing values, after there has been at least one submit of the form. Change reasons can be included for the initial set of data, but they are ignored, instead the default ‘changes before submission’ is used. That said, it is easier to send your change reason with all form updates. In this way you don’t have to track necessarily that it's the 1st or later touch of the form data.</td></tr>
<tr><td>Jobs</td><td>The Vault CDMS job types that allow start / inspect / retrieve output follow the Vault platform job flow. That is, a job of Completed status might not necessarily not have had any issues. Depending on the job type, more detailed log style information is usually included in the output. Both the Vault job log and the content of the job’s output should be inspected.</td></tr>
<tr><td>User Administration</td><td>For updating of users with a CDMS vault, ensure the proper usage depending on the need:<br>(1) For full deactivation of an account, this will remove the user from all vaults (thus all studies in those vaults) on that domain. Reactivation of an account is later (if needed) a two step process. (update of domain status, then update of vault memberships)<br>(2) For deactivation of an account at the vault level, this will remove the user from any studies they otherwise would have had access to, in that vault<br>(3) For study access deactivation -> use the upload users endpoint, with a line indicating Study Status = Disabled, to remove access to a single study. Specific site / country access does not need to be removed, as later a re-activation in the study might happen, returning the user to the state they were at deactivation in the study.</td></tr>
<tr><td>User Administration</td><td>The add/update of users in the vault involves the pass of a CSV of user information in the same way as the Vault CDMS Users UI. The CSV template can change from release to release as new properties are introduced. If some of the new properties are required, this could upend existing built integrations. The API version does NOT refer back and use older CSV templates. Consult the CDMS Help and experiment in pre release vaults on coming releases when writing user administration integrations.</td></tr>
</table>