

EDT Core API - Notes for Maple

EDT 11

Contents

Contents	1
Revisions	3
EDT 11.6.2	9
Changes	9
Additions.....	10
EDT 11.6.3	10
EDT 11.6.4	11
EDT 11.6.5	11
EDT 11.6.6	11
EDT 11.6.7	11
EDT 11.6.8 Beta	11
EDT 11.6.8	13
EDT 11.8 Beta.....	13
EDT 11.8 (pre) Beta 2.....	14
EDT 11.8 Release Candidate.....	15
Terminology	16
Introduction	17
Environment Names	18
Proof of Concept server	18
Testing the API	18
Detailed Documentation.....	19
Guide to using the API.....	19
Identifying Resources.....	19
EDT IDs	19
Using Keys to identify resources	20
Summary of API endpoints.....	21
Endpoint Details.....	23
Cases	23
Getting a list of Cases	23
Case Custom Fields.....	23
GetIdByKeyOrField - Looking up a Case ID using the key or a custom field	23

CaseldsLookup	24
5.2.1 Creating a Case	24
5.2.2 Updating the metadata of a Case	25
5.2.3 Merging an Agency File or a Court File – DRAFT	26
Case Users	26
5.3.1 Adding & Removing Users to a Case	26
Extra scenario – Add a user to multiple cases - SyncUserCases	27
Extra scenario – GetUserCases – Get a list of the cases a specified user has access to	28
Case Groups	28
Supporting endpoint: Get the Groups in a Case	28
Supporting endpoint: Get a Group's details	29
Updating a group's users using the SyncGroupUsers endpoint (ISL)	29
Updating a group's users using AddUserToCaseGroup and RemoveUserFromCaseGroup (DIAM)	29
Getting the groups of a user	30
Participants/Persons	30
5.4.1 Adding a Person (previously called a Participant)	30
Supporting endpoint: GetPersons (v1) - Deprecated	32
Supporting endpoint: GetPersons (v2)	32
Supporting endpoint: Get the details of a Person resource using an ID and using a Key	33
5.4.2 Updating a Person (previously called a Participant)	34
5.4.3 Add a Person (Participant) to a Case	36
5.4.4 Remove a Participant from a Case	37
Supporting endpoint: GetCaseParticipants	37
Supporting endpoint: GetParticipantTypes	38
5.4.5 Merging Person (previously called Participant) Details	38
Supporting endpoints – Identifiers	38
Records	40
Searching for Records	40
Using Keys with the Record APIs	41
5.5.1 Adding a Record	41
5.5.2 Updating a Record	43
5.5.4 Deleting a Record	44
Extra - Deleting all records in a Case	44
Extra - Deleting all of a record's renditions	44
User Accounts	44
Extra Scenario - Creating a User	44
Supporting User endpoints	45
Extra scenario: UpdateUser	45
Extra scenario: Add and remove user to/from OU Group	45
Extra scenario: GetUserOuGroups	46
Platform Events	46
Get Platform Events	46
Notifications	48
Creating a new Notification	48
Other	49
Get the Version	49
Appendix – TimeZones in EDT	49

Revisions

May 2022 – Initial draft of document

Updates below:

Date	Exposed endpoints and other changes
28 Apr 2022	<p>Exposed API routes:</p> <ul style="list-style-type: none"> POST /api/v1/cases [Update: this route is no longer valid. See 5.2.1 <i>Creating a Case</i>]
13 May 11.3.365	<p>Exposed API routes:</p> <ul style="list-style-type: none"> POST /api/v1/cases/{caseId}/case-users/sync POST /api/v1/cases/{caseId}/groups/{groupId}/sync GET /api/v1/version GET /api/v1/cases/{caseId}/groups GET /api/v1/cases/{caseId}/groups/{groupId} <p>The Swagger endpoints have changed to:</p> <ul style="list-style-type: none"> auth: https://346n1v5oeh.execute-api.ca-central-1.amazonaws.com/swagger/index.html open: https://346n1v5oeh.execute-api.ca-central-1.amazonaws.com/swagger-ui/index.html <p>The start of other routes has changed from "/v1" to "/api/v1" to be consistent with the documentation.</p>
20 May 11.3.403	<p>Model changed for the CreateCase API:</p> <ul style="list-style-type: none"> Properties removed: Id, IsS3, IsThumbnailsEnabled, ExportTempPath, StagingAreaPath Properties changed: int TemplateId is now string TemplateCase. This now accepts either a Case Name or a Case Id TimeZone is now TimeZoneId Suffix is now DatabaseFileSuffix <p>Documentation below updated to reflect the changes above and add additional information.</p> <p>The implementation of Case creation using a template has been completed.</p> <p>The <i>Add Users to Case</i> scenario now only adds non-administrators to the specified Case. This is now consistent with the EDT UI. (Administrators have access to all cases without being explicitly added)</p> <p>Improvements have been made to the error handling. The most visible change is more meaning error messages and response codes being returned on more occasions.</p> <p>The previously reported technical issue that limited the content available via the Swagger UI has been resolved. Going forward, more detailed information about the individual API routes will be provided via the Swagger UI instead of in this document.</p>
30 May 11.3.469	<p>This build exposes the primary API routes used for the 5.2.2 <i>Updating the metadata of a Case</i> scenario:</p> <p>Exposed API routes:</p> <ul style="list-style-type: none"> GET /api/v1/cases/{caseId} PUT /api/v1/cases/{caseId} <p>Some of the more detailed documentation that was in this document has been removed as it is now available from the Swagger UI. A new section below called <i>Detailed Documentation</i> contains some further information.</p>

Date	Exposed endpoints and other changes
8 Jun 11.5.54	<p>Scenario: 5.4.1 Adding a Participant [Update: these Person routes have been changed to include OrgUnit]</p> <ul style="list-style-type: none"> • POST /api/v1/persons • GET /api/v1/persons • GET /api/v1/persons/{personId} <p>Scenario: 5.5.1 Adding a Record</p> <ul style="list-style-type: none"> • POST /api/v1/cases/{caseId}/records • PUT /api/v1/cases/{caseId}/records/{docId}/{fileType}
23 Jun 11.5.238	<p>Added some notes about some functionality under consideration. See "[Proposed] Using Keys to as an alternative to EDT IDs"</p> <p>Scenario: 5.4.1 Adding a Participant [Update: this route has changed to include OrgUnit]</p> <ul style="list-style-type: none"> • POST /api/v1/persons. Updated to support Title and Custom Fields <p>Scenario: 5.4.2 Updating a Participant [Update: this route has changed to include OrgUnit]</p> <ul style="list-style-type: none"> • PUT /api/v1/persons
1 Jul 11.5.307	<p>The section describing the proposed introduction of Keys has been revised with a new approach.</p> <p>The following endpoint can be used to lookup a Case ID using a Key:</p> <ul style="list-style-type: none"> • GetByIdByKey /api/v1/cases/{key}/id <p>Scenario: 5.5.2 Updating a Record</p> <ul style="list-style-type: none"> • PUT /api/v1/cases/{caseId}/records <p>Scenario: 5.5.4 Deleting a Record</p> <ul style="list-style-type: none"> • DELETE /api/v1/cases/{caseId}/records/{idOrKey} <p>Added Users endpoints (additional to the scenarios in the DEMS specification):</p> <ul style="list-style-type: none"> • CreateUser: POST /api/v1/users • GetUsers: GET /api/v1/users • GetUser: GET /api/v1/users/{userIdOrKey}
8 Jul 11.5.342	<p>The proposed introduction of Keys has moved out of the proposal stage. Keys will be introduced to routes and resources in the coming weeks.</p> <p>Participant (Person) APIs updated to support specifying an ID or Key: [Update: these routes have changed to include OrgUnit]</p> <ul style="list-style-type: none"> • GET /api/v1/persons/{idOrKey} • PUT /api/v1/persons/{idOrKey}

Date	Exposed endpoints and other changes
4 Aug 11.5.558	<p>OrgUnits We have started to make the APIs <i>OrgUnit</i> aware. EDT is currently introducing OrgUnits, whose primary intention is to provide multi-tenancy in EDT. We assume they are of little interest to Maple, so these notes assume there is one OrgUnit, the default one, which has an ID of 1.</p> <p>The following routes have been updated to require/support specifying the OrgUnit.</p> <pre> GetPersons GET /api/v1/org-units/{orgUnitId}/persons GetPerson GET /api/v1/org-units/{orgUnitId}/persons/{personIdOrKey} CreatePerson POST /api/v1/org-units/{orgUnitId}/persons UpdatePerson PUT /api/v1/org-units/{orgUnitId}/persons/{personIdOrKey} DeletePerson DELETE /api/v1/org-units/{orgUnitId}/persons/{personIdOrKey} </pre> <p>Case Endpoints: change to the routes for GetCases and CreateCase to include specifying of the OrgUnit:</p> <pre> CreateCase POST /api/v1/org-units/{orgUnitId}/cases GetCases GET /api/v1/org-units/{orgUnitId}/cases </pre> <p>Scenario 5.3.1 <i>Adding & Removing Users to a Case</i> updated:</p> <ul style="list-style-type: none"> The CaseUsersSync endpoint now uses arrays of Ids and/or Keys. <p>Records.</p> <ul style="list-style-type: none"> The way records are identified has changed. See <i>Using Keys with the Record APIs</i> for details. <p>The CreateCase and UpdateCase endpoints have been updated in the following ways:</p> <ul style="list-style-type: none"> The Key field has been added so the Key value can be set/changed. The way that custom fields are specified has changed.
19 Aug 11.5.633	<p>Case ID Lookup. The topic "Looking up Case ID using a Key" has been changed to "Looking up a Case ID using a key or custom field". The endpoint's route has changed to include the Org Unit.</p> <p>DeleteRecords endpoint added to this document. This will be used by the <i>Return Case to Police</i> and <i>Merging an Agency or Court File</i> scenarios.</p> <pre> DELETE /api/v1/cases/{caseId}/records </pre> <p>SyncUserCases endpoint added. (Sometimes referred to as 5.3.1c:)</p> <pre> POST /api/v1/org-units/1/users/{userIdOrKey}/cases/sync </pre> <p>Persons The naming requirements for Persons have been relaxed. Previously a Name, First Name, Last Name, and Email Address were required, and these had to be unique. Now none of these fields are required and the names fields do not have to be unique. If the Email Address is provided, it must be unique within the Org Unit.</p> <p>Corrections:</p> <ul style="list-style-type: none"> Removed the notes about "Document+ID" from the RecordUpdate and RecordDelete APIs. In the examples in 5.2.1 <i>Creating a Case</i> and 5.2.2 <i>Updating..a Case</i>, the name of the "timeZone" parameter was changed to "timeZoned".

Date	Exposed endpoints and other changes
23 Sep 11.5.887	<ul style="list-style-type: none"> CaseldLookup <ul style="list-style-type: none"> Added support for "Key:" prefix. <ul style="list-style-type: none"> If specifying a Key, the endpoint now ensures the returned Case is in the specified OrgUnit. Implemented the GetCaseParticipants API. Example: <div>GET /api/v1/cases/{caseId}/participants</div> <p>The results can also be filtered by person (ID or Key) and/or participant type. See <i>"Supporting endpoint: GetCaseParticipants"</i> for more information and examples.</p> Implemented the GetParticipantTypes API: <div>GET /api/v1/org-units/{orgUnitId}/participant-types</div> Added the "ParticipantTypeFilter" option to the SyncCaseParticipants API to support syncing only the persons of a specified Participant Type. For examples, see the updated notes under <i>"5.4.3 Add a Participant to a Case"</i>. [Do not use this API without discussing with Phil H first, as the API may be removed.] Implemented the AddCaseParticipant endpoint as an alternative to SyncCaseParticipants. See the notes for <i>"5.4.3 Add a Participant to a Case"</i> Draft notes for <i>5.2.3 Merging an Agency File or a Court File</i>
29 Sep 11.5.912	<ul style="list-style-type: none"> A new DateOnly custom field type is available. Primarily for Cases at this point. Some UI references may now refer to the previous/existing "Date" field type as "Date Time" or similar. Person resources. The ability to get and set custom field values during an Update has been restored. Currently works correctly with Text and DateTime custom fields. Also see additional notes in <i>"5.4.1 Adding a Participant"</i> section.
7 Oct 11.5.987	<ul style="list-style-type: none"> Case custom fields. Added the ability to refer to fields and list values by Name. Specifying a list value that doesn't exist automatically adds it. Ability to add and remove a user to/from an Org Unit (site) group. Person resources. Fixed: CreatePerson does not set custom fields. (Only Text and DateTime custom fields are currently supported for Person resources)
24 Oct 11.5.1051	<ul style="list-style-type: none"> Duplicate case names allowed in CreateCase and UpdateCase UpdateUser endpoint added. GetOrgUnitFields added: <div>GET /api/v1/org-units/{orgUnitId}/fields</div> <p>This replaces the previous route which was:</p> <div>GET /api/v1/site/fields?organisationalUnitId=1 // DO NOT USE!</div> The GetVersion API now returns a different version number than what is displayed in the EDT Web UI. This is because the API and UI have been split into separate services with their own development roadmaps. The GetVersion API returns the version of the API.

Date	Exposed endpoints and other changes
16 Nov 11.5.1255 API: 11.5.234	<ul style="list-style-type: none"> Added the ability to use custom field names in the CaseldLookup endpoint Breaking change: Changed the result of the CaseldLookup endpoint to return a list of case ID/Key values. Added a new GetUserCases endpoint to get a list of the cases a specified user is assigned to. Added the ability to find records using a multiple field values. See the <i>Records > Searching for Records</i> section for more information. The GetUserCases endpoint was returning cases with all statuses. Eg including removed cases. Now it returns Active and Inactive cases only by default, but an optional parameter can be included to return cases with other statuses.
1 Dec 11.5.1375 API: 11.5.309	<ul style="list-style-type: none"> Added GetUserOuGroups endpoint to return a list of the OU (site) Groups that a user belongs to. See <i>Extra scenario: GetUserOuGroups</i>.
2023	
9 Feb 11.5.1762/ 11.5.470	<ul style="list-style-type: none"> The limitation on Participant custom field types has been removed. This change was in 11.5.1762/11.5.470 deployed to POC and Vanilla on 1 Feb
8 Mar	Document update: Clarified the previous note and added an Environment Names section to the Introduction
23 Mar Web 11.5.2146 API 11.5.628	<p>Document update:</p> <ul style="list-style-type: none"> Terminology section added Introduction updated Summary of API endpoints section added <p>New endpoints:</p> <ul style="list-style-type: none"> AddUserToCaseGroup (DIAM, NED-232) <pre>POST /api/v1/cases/{caseId}/groups/{groupId}/users/{userId}</pre> <ul style="list-style-type: none"> RemoveUserFromCaseGroup (DIAM, NED-232) <pre>DELETE /api/v1/cases/{caseId}/groups/{groupId}/users/{userId}</pre> <p>Note: the endpoint for adding users to groups that Maple reported to EDT as not working is not a published endpoint and should not be used. We have not included the full route of that endpoint here because this document is only for published endpoints. However, to help ensure the correct endpoint is used note that the unpublished/incorrect endpoint includes "case-users" in the route whereas the correct endpoints (shown above) do not.</p>

13 Apr

Web

11.5.2243

API

11.5.683

Added an example of Updating a record's native file using C#.

Minor endpoint changes

- **SaveCaseSettings endpoint.** In line with removing the option from the UI, the UseDocIdForDownloads property of the CaseUpdateRequest has been deprecated and any value passed for this property passed in the request will be interpreted as True.
- **UpdateRecordFile endpoint.** The fileType parameter of the UpdateRecordFile endpoint is now completely case-insensitive. Eg when updating a native file, previously the fileType had to be "Native" or "native". The extension of the file being uploaded previously had to be lowercase.

Changed Routes

As part of converting the API from a private/internal API to a published API, work is being done to ensure any routes published are consistent, predictable, and hierarchical.

To this end, some of the unpublished routes that are visible in the Swagger UI may become deprecated and will eventually be removed. Swagger UI shows these are deprecated. These deprecated routes still work but should not be used.

All of the routes that have been deprecated were previously unpublished. There is one exception.

The SyncCaseUsers was previously published for use by the ISL. Its route has been changed and the old endpoint has been renamed SyncCaseUsersOld. Typically, a change like this would result in a new version of the API, but since this endpoint is only used by Maple we are requesting that the ISL be updated to use the new route. If this is not acceptable, please let Phil @ EDT know. The details and examples of this endpoint have been updated to indicate the change.

Details about SyncCaseUsers change, and the changes to the unpublished endpoints are below.

SyncCaseUsers

```
Deprecated: POST /api/v1/cases/{caseId}/case-users/sync
New route:  POST /api/v1/cases/{caseId}/users/sync
```

This route has been changed for consistency. There have been no changes to behaviour.

GetCaseUsers

```
Deprecated: GET /api/v1/cases/{caseId}/case-users
New route:  GET /api/v1/cases/{caseId}/users
```

The implementation is currently mostly the same, except the new route returns less info about each case user (i.e. CaseId, UserId, Username) than the old route. If the information provided is inadequate, please contact Phil @ EDT with details about the additional data to be included.

AddUserToCase

```
Deprecated: POST /api/v1/cases/{caseId}/case-users/{userId}
New route:  POST /api/v1/cases/{caseId}/users/{userId}
```

The unpublished/deprecated endpoint does not work correctly. The new endpoint does.

RemoveUserFromCase

```
Deprecated: DELETE /api/v1/cases/{caseId}/case-users/remove/{userId}
New route:  DELETE /api/v1/cases/{caseId}/users/{userId}
```


Date	Exposed endpoints and other changes
	<p>The new route has an Improved implementation including better validation etc.</p> <p>GetCaseGroupsOfUser (was GetCaseUserGroups)</p> <div> <p>Deprecated: GET /api/v1/cases/{caseId}/case-users/{userId}/groups</p> <p>New Route: GET /api/v1/cases/{caseId}/users/{userId}/groups</p> </div> <p>The new route has an Improved implementation including better validation etc.</p>

EDT 11.6.2

Web version 11.6.2.89. API version 11.6.2.87

Changes

Changed endpoint names

Some of the endpoints have been renamed.

Old name and route		New Name
GetCase	GET /api/v1/cases/{caseId}	GetCaseDetail
UpdateCase	PUT /api/v1/cases/{caseId}	SaveCaseSettings
AddUserToCase	POST /api/v1/cases/{caseId}/users/{userId}	AddCaseUser
RemoveUserFromCase	DELETE /api/v1/cases/{caseId}/users/{userId}	RemoveCaseUser
GetCaseGroups	GET /api/v1/cases/{caseId}/groups	GetGroups
GetCaseGroup	GET /api/v1/cases/{caseId}/groups/{groupId}	GetGroup
SyncCaseGroupMembers	POST /api/v1/cases/{caseId}/groups/{groupId}/sync	SyncGroupUsers
GetActivePersons	GET /api/v1/org-units/{orgUnitId}/persons	GetPersons
UploadRecordFile	PUT /api/v1/cases/{caseId}/records/{recordId}/{fileType}	UpdateRecordFile

Updated endpoint: UpdateRecordFile

Non-breaking change.

Added a `renditionAction` query string parameter. This is used to control what happens to related renditions of a record when a file is updated.

Other endpoint changes

- CreateNewCase endpoint. A new `IsTemplate` property has been added to the `CaseCreateRequest` schema.
- SaveCaseSettings endpoint. A new `IsTemplate` property has been added to the `CaseUpdateRequest` schema.
- RemoveUserFromCaseGroup endpoint. If the request returns an error (e.g. a 404) the response content type is now `application/problem+json`.

Additions

New endpoints for (Person) Identifiers

GetIdentifiers

```
GET /api/v1/org-units/{orgUnitId}/identifiers
```

GetIdentifierById

```
GET /api/v1/identifiers/{identifierId}
```

GetIdentifiersForEntity

```
GET /api/v1/org-units/{orgUnitId}/identifiers/{entityType}/{entityId}/{identifierType}
```

CreateIdentifier

```
POST /api/v1/org-units/{orgUnitId}/identifiers
```

UpdateIdentifier

```
PUT /api/v1/identifiers
```

DeleteIdentifier

```
DELETE /api/v1/identifiers/{identifierId}
```

New endpoint: RemoveCaseParticipant

```
DELETE /api/v1/cases/{caseId}/participants/{personIdOrKey}
```

New endpoint: CaseldsLookup

```
POST /api/v1/org-units/{orgUnitId}/cases/lookup-ids
```

New endpoint: ReassignParticipantCases

```
POST /api/v1/org-units/{orgUnitId}/persons/{fromPersonIdOrKey}/reassign-cases/{toPersonIdOrKey}
```

Section "5.4.5 Merging Participants Details" has been added to this document which describes this endpoint in more detail.

EDT 11.6.3

No API enhancements. Some fixes.

While not part of the 11.6.3 release, on 10 Nov 2023 Elaine was notified that they can use the unpublished endpoint `GetSiteCaseInfo` to get the status of cases being created.

EDT 11.6.4

Web version 11.6.4.89. API version 11.6.4.87

Enhancements to the GetIdentifiers endpoint

EDT 11.6.5

New endpoint: DeleteRenditions

```
DELETE /api/v1/cases/{caseId}/records/{idOrKey}/renditions
```

Provides the ability to delete a record's renditions. NED-1554

EDT 11.6.6

Fix for NED-1953 – Issue with adding a merged participant to a case.

EDT 11.6.7

Changed endpoint: GetCases endpoint

This is a breaking change. Breaking changes are typically avoided. However, in this instance, it was deemed acceptable after consultation and coordination with Maple.

The GetCases endpoint now returns a paged response. The default Page is 1 and the default PageSize is 50.

The response model containing details of each case has also changed. It no longer returns irrelevant or per-user data, such as LastAccessedUtc.

NED-2083

EDT 11.6.8 Beta

2 Jul 2024. Web: 11.6.8.83. API: 11.6.8.59.

Changed endpoint names

The following endpoint names in this document have been changed to be consistent with Swagger.

Previous	New
GetVersion	GetVersionInfo
GetCustomFields	GetOrgUnitFields
CaseldLookup	GetIdByKeyOrField
CreateCase	CreateNewCase
SaveCaseSettings	UpdateCase
ExportToCase	MergeCase
GetOuGroups	GetOrgUnitGroups

AddUserToOuGroup	AddUserToOrgUnitGroup2
RemoveUserFromOuGroup	RemoveUserFromOrgUnitGroup2

New endpoint: GetPersons version 2

The full operationId in Swagger for this endpoint is "PersonApiControllerV2/GetPersons"

```
GET api/v2/org-units/{orgUnitId}/persons
```

Supports filtering and paging.

Ticket: NED-3012.

Deprecated endpoint: GetPersons version 1

Use the GetPersons version 2 endpoint instead.

Changed request and response models

Model	Details
EDT.SiteCustomFieldServiceCore.Models.SiteField	Response model for endpoint: GetOrgUnitFields New property: <code>DisplayInAboutCase</code> (boolean)
EDT.CaseServiceCore.Models.CaseldAndKey	Response model for endpoints: <ul style="list-style-type: none"> <code>GetIdByKeyOrField</code>. (Previous name <code>CaseldLookup</code>) <code>GetUserCases</code> New property: <code>Status</code> (string). This contains the Status of the Case. Ticket: NED-1990.
EDT.CaseServiceCore.Models.CaseModel	Response model for endpoint: CreateNewCase New property: <code>createdViaUi</code> (boolean)
EDT.EdtApiCore.ApiModels.CaseUpdateRequest	Request model for endpoint: UpdateCase New properties: <ul style="list-style-type: none"> <code>exportShowMarkupWarning</code> (boolean). Enables a warning when attempting to export without markup sets selected. Note: setting this property does not currently do anything in the beta. <code>pdfPresenter</code> (boolean, nullable). Show UI controls to display records in the PDF Presenter. If not set or null, the value will not be updated.
EDT.SiteServiceCore.Models.SiteCaseSettings	Response model for endpoints: <ul style="list-style-type: none"> <code>GetCaseDetail</code> <code>UpdateCase</code> New property: <code>exportShowMarkupWarning</code> (boolean)
People.ParticipantService.Models.ParticipantsInCase	Response model for endpoint: GetCaseParticipants New property: <code>isActiveCase</code> (boolean)

EDT.SiteServiceCore.Models.SiteCaseDetails	Response model for endpoint: GetSiteCaseInfo New property: totalCases (integer)
--	--

EDT 11.6.8

22 Jul 2024 - Build 11.6.8.67

Updated endpoint: GetPersons version 2

Note: The full operationId (name) in Swagger for this endpoint is "PersonApiControllerV2/GetPersons"

Changes:

- Optionally include inactive Persons in results. NED-4149.
- Add the ability to filter by email address. NED-2287.

Updated endpoints: GetGroups and GetGroup

New optional query string parameter:

- includeCounts. Boolean. Optional. Defaults to False. If True, includes the number of users and claims (permissions) of each group.

Updated request model: EDT.EdtApiCore.ApiModels.CaseUpdateRequest

Request model for endpoint: UpdateCase

Changed property:

- exportShowMarkupWarning. Boolean. Optional/nullable. Controls whether a warning is displayed to the user when they attempt to export without markup sets selected.
This property was included in the beta but was not functional. In the release, this property is now functional and optional.
If not set or null, the underlying setting will not be updated.

EDT 11.8 Beta

10 Dec 2024

New endpoint: GetPlatformEvents

EDT 11.8 includes a new feature called Platform Events. Platform Events enable external systems to monitor events occurring within the platform. There are currently two event types available related to Export-to-Participant events.

For information on how to get platform events, see the Platform Events section.

New endpoint: CreateNotification

A new feature called Notifications has been added in 11.8. Notifications are shown on the Notifications tab of the Activity Log view. For information on how to create a Notification via the API, see the Notifications section.

Changed: The default value of the CreatedViaUi request model property in the CreateNewCase endpoint

On 5 Sep 2024, in response to a support issue (FD 21173), Maple was informed that they could add CreatedViaUi to the request model and set it to True in the CreateNewCase endpoint to avoid the issue.

In EDT 11.8 this property defaults to True if not provided, so it is no longer necessary to explicitly set it.

“Deprecated” flag removed from endpoint: SyncCaseParticipants

The SyncCaseParticipants endpoint was deprecated in 11.6.2. Maple is still using this endpoint extensively and there isn't a more suitable alternative endpoint at present, so the Deprecated flag has been removed in this documentation to avoid confusion.

Breaking changes to Participant-related endpoints

Endpoint	Breaking Change
CreatePerson and UpdatePerson	<code>Address.IsPrivate</code> is no longer supported. A value can still be supplied but is not saved. <code>Address.PostalAddressAsJson</code> is no longer supported. A value can still be supplied but is not saved.
AddCaseParticipant	The endpoint now returns a 400 response code if the person is already assigned to the case.
GetCaseParticipants	<code>OrganisationMemberId</code> is always 0 <code>Title</code> is always empty <code>Parents[.OrganisationMemberId]</code> is always 0c

Changed response model: EDT.SiteCustomFieldServiceCore.Models.SiteField

This is the response model for endpoint: GetOrgUnitFields

New property: `IsRichText` (boolean)

Note: Identifiers endpoints

The Identifiers endpoints continue to work but the identifiers managed by these endpoints are no longer visible in the UI. The new Persons/Orgs UI has areas to manage Identifiers, but these *Entity Identifiers* differ from those set via the published API endpoints.

EDT 11.8 (pre) Beta 2

Person Names and the Person-related API endpoints - NED-5315

The request/response models for the Person-related endpoints include a Firstname property. The Person details dialog in the EDT 11.8 UI displays two name fields: Given Names and Preferred First Name.

In previous 11.8 builds, specifying Firstname in CreatePerson and UpdatePerson would update both the Given Names and Preferred First Name fields. The Firstname property now updates just the Given Names field.

Note: A consequence of this change is that the Preferred First Name field in the UI cannot be updated using the existing API endpoints.

Documentation update: UpdatePerson endpoint

The behaviour of the UpdatePerson method changed in EDT 11.8. The section *5.4.2 Updating a Person* (previously called a Participant in this document compares the pre-11.8 and 11.8 behaviour. This is part of the resolution to NED-5259/FD 22128.

Updated endpoint: CaseldsLookup - NED-5273/BCPSDEMS-2187

The response model has been updated to also return the Status of each case.

Fixed endpoints:

- SyncCaseParticipants errors. NED-5280/FD 22110.
- GetPerson/GetPersons(v2) not returning custom fields. NED-5300/FD 22212.
- MergeCase is placing records in the Default folder. NED-5337/FD 21541.
- CreatePerson/UpdatePerson issues. Resolution includes code changes and updates to this documentation (mentioned above). NED-5259/FD 22128.

Person DOB and Category properties

Coming in a future EDT 11.8 build: A `dob` property and a `category` property will be added to the Person request/response models used by the GetPerson, GetPersons, CreatePerson, and UpdatePerson endpoints. NED-5324.

The GetPersons endpoint already supports filtering on DOB and Category fields. When filtering on Category, use the Category ID of the Person.CategoryType field, not the Description. In the UI, see Site Settings > Fields > Person > Category Type for the IDs and Descriptions. The following is an example of filtering on these fields:

```
GET {{host}}/api/v2/org-units/1/persons?filter=category:other,dob:2000-04-01
```

EDT 11.8 Release Candidate

12 Mar 2025 - 11.8.0.235 (API 11.8.0.203)

Added DOB and Category standard fields to Person endpoints

Standard fields named `dob` and `category` have been added to the CreatePerson, UpdatePerson, GetPersons (v2), and GetPerson request/response models.

When using the UpdatePerson endpoint, if the `dob` or `category` fields are not specified, the respective values will not be updated.

There is currently no way to clear/remove the values from these fields. That is, they can only be set to a different value.

When setting the `dob` field:

- The value must be a valid date.
- The value must be a date or datetime using the format "2025-03-12" or "2025-03-12T00:00:00". If a time is included, it will be ignored.
- If the value provided is not valid, a 400 response code is returned.

When setting the `category` field:

- The value must be the relevant CategoryType.ID, not the CategoryType.Description. (See Site Settings > Fields > Person > CategoryType)
- The value is case-insensitive. For example, specifying a value of "other" will match the "Other" CategoryType.ID.
- If the value provided is not valid, a 400 response code is returned.

The following example illustrates how to update these 2 fields for a person with the ID of 3:

```
PUT /api/v1/org-units/1/persons/3
content-type: application/json

{
  "category": "Other",
  "dob": "2000-01-14"
}
```

EDT 11.8 Release

9 Jun 2025: Toolbox: 11.8.0.263, API: 11.8.0.219

No additional changes to this document.

EDT 11.9

EDT Web Version: 11.9.0.125 (API Version: 11.9.0.111)

1 Jul 2025

PDEMS project

The first 3 headings below are related to the PDEMS project (NED-3791)

- New Endpoint: CreateLoadFileImport
- New Endpoint: GetImport
- Platform Events

See *PDEMS Design - Automating Imports - v1.0 2025-03-12.pdf* for further information about this project.

New Endpoint: CreateLoadFileImport

```
POST {{host}}/api/v1/cases/85489/create-loadfile-import
content-type: application/json

{
  "importName": "PDEMS - Phils Initial import",
  "loadFilePath": "Files/2024-12-03 12-47-40 Marnie/Delimited_Data.csv",
  "importTemplate": "CSV"
}
```

New Endpoint: GetImport

```
GET {{host}}/api/v1/cases/{caseId}/imports/{importId}
```

Platform Events

Two new event types are created

- LoadFileImportAPI. This event type is raised as a result of a call to CreateLoadFileImport (PDEMS)
- LoadFileImportUI. This event type is not related directly to the PDEMS project.

To get import events related to PDEMS, use the following:

```
GET {{host}}/api/v1/platform-events?filter=eventtype:LoadFileImportAPI
```

To get these import events specific to a case:


```
GET {{host}}/api/v1/platform-events?filter=eventtype:LoadFileImportAPI,caseid:1234
```

Further clarification on Person names and documentation update to *Supporting endpoint: GetPersons (v2)*

The transition from the Participants 1.0 schema (in 11.6.8) to the Participants 2.0 schema (in 11.8) aimed to avoid breaking changes to the published API.

One of the factors during this transition was that the 1.0 schema has a `FirstName` field and the Participants 2.0 schema does not. Instead, the Participants 2.0 schema has `Given Names` and `Preferred First Name` fields (mostly visible in the UI). To address this, the `FirstName` field in the existing API endpoints was mapped to the `GivenNames` field in Participants 2.0. The pre-11.8 published endpoints should always read/write the new `Given Names` field when `FirstName` is used in the API.

One of the consequences of retaining the use of the pre-11.8 endpoints, is that `Preferred First Name` cannot be updated/returned. We have not published an endpoint to read/write that field at this point.

There is an issue/inconsistency when filtering with the `GetPersons (v2)` endpoint (introduced in 11.6.8):

- Filtering on `FirstName` in the filter parameter filters on `Preferred First Name`, and
- Filtering on `GivenNames` in the filter parameter filters on `Given Names`.

See the new Important Note in the “Supporting endpoint: `GetPersons (v2)`” chapter for more information.

Documentation Update: 5.5.2 Updating a Record

Updated to state that `PrimaryDateUtc (Date)` will be cleared if not provided. This is a clarification in the documentation, not a change in the endpoint’s behaviour.

Terminology

Term	Definition
Published	<p>Published endpoints are those that can be used by 3rd party clients.</p> <p>Published endpoints are supported by EDT.</p> <p>In general, published endpoints will not receive breaking changes. However, there may be scenarios where a breaking change is desirable or necessary and, in these situations, EDT will discuss the change with any parties using the endpoint.</p> <p>All endpoints in this document are published. All other endpoints in the EDT Core API are not published and not to be used.</p>

Introduction

The EDT Core API is in the process of being made available (“published”) to external parties. While extensive, many of the endpoints within the API are not ready for use by external parties. This document contains information about the endpoints that are ready to be used. That is, the endpoints in this document are “published”. As such, EDT will provide support for these endpoints and will not change these endpoints in a way that would break any clients, at least not without first discussing the proposed change with those using these endpoints.

This document contains notes about the API as it is being developed and progressively being exposed to Maple.

The purpose of this document is to communicate the API endpoints that are ready to be used and to provide information on how to use them.

Much of the detailed content of this document will eventually be available via Swagger. As that happens, the duplicated information will be removed from this document.

Important: Any endpoint that appears in the Swagger UI that is not published in this document SHOULD NOT be used.

Environment Names

There are 4 EDT environments. Some are referred to using different names

EDT Term	BCPS
POC (Proof of Concept)	Dev
PreProd	Test
Production	Prod/Production
Vanilla	Vanilla

Proof of Concept server

The Proof of Concept (POC) server is an EDT server that exposes the EDT API for the development and testing of the Maple ISL to EDT (DEMS) integration. This server will be updated regularly as more API endpoints relevant to Maple are exposed¹ and ready for testing.

The address of the POC is: <https://346n1v5oeh.execute-api.ca-central-1.amazonaws.com/>

At present the POC server uses a token to restrict access. This token must be included in the Authorization header of requests like this:

```
Authorization: Bearer <secret>
```

The "<secret>" shown above is a placeholder for the actual value. This token is available on request from EDT made by relevant BCPS personnel.

Related to this, most of the routes listed by our Swagger page are not currently callable and will return a 404 result. Only the routes we list in the API Notes document have been made callable at this point.

Testing the API

For testing purposes, the API can be called using custom code or a REST client such as Postman or Visual Studio Code with the "REST Client" extension.

To make a request, include the host at the beginning of the route. For example, to call the route documented as:

```
GET /api/v1/version
```

The request should be:

¹ Only endpoints/routes listed in this document are callable (exposed). Other routes listed in the Swagger UI will return a 404 (Not found) if called.

```
GET https://346n1v5oeh.execute-api.ca-central-1.amazonaws.com/api/v1/version
```

Detailed Documentation

As we make endpoints available, we will also make detailed documentation available via the Swagger UI. The following graphic shows some of the places where this documentation appears. For example, click on the Schema tab shown below to see notes about the properties of the model to be passed in the request body.

POST `/api/v1/cases` Create a new Case

Sample request:

```
POST /api/v1/cases
Authorization: Bearer <secret>
{
  "name": "Case name goes here",
  "timeZoneID": "Pacific Standard Time"
}
```

Parameters

No parameters

Request body *required*

Example Value **Schema**

CaseCreateModel ▾ {

- `waitForCaseCompletion` `boolean`
- `name` `string`
`nullable: true`
- `description` `string`
`nullable: true`

Whether to wait for the new case to be created or return immediately. Default: True.

The name of the Case. Must be unique.

Guide to using the API

Identifying Resources

EDT IDs

The routes to specific resources require the IDs of the resources to be included. For example, to get the details of a Case, you would use a route like this:

```
GET /api/v1/cases/{caseId}
```

In the example above, the `caseId` is required in order to get the details of the specified Case.

Typically, these IDs values are the EDT IDs of the resource. EDT IDs are allocated automatically by EDT when the resource is created. These IDs are almost always an integer value such as 12345. The exception are User resources that have string IDs such as "1eb14bb6-7c0f-41d0-ba52-17bb34286aff".

Using Keys to identify resources

Some routes in the EDT API support the use of Keys as an alternative to using EDT IDs to identify resources. A Key is a standard field on the resource where the value of the key uniquely identifies the resource in a similar way to how the EDT ID uniquely identifies the resource. However, unlike EDT IDs, the Key is set by client code via the API. Setting the Key value is optional.

Since (for most resources) the internal EDT ID is an integer value, the API can determine whether you are passing an ID or a Key by checking to see if the value represents a number or not. For example, given the API endpoint:

```
GET /api/v1/org-units/{orgUnitId}/persons/{personIdOrKey}
```

If you pass:

```
GET /api/v1/org-units/1/persons/1234
```

EDT will interpret "1234" on the end as being the EDT ID of the person, but if you pass:

```
GET /api/v1/org-units/1/persons/abc
```

EDT will interpret "abc" as being a Key value.

For records, specifying a Key would look up the record by Document ID. So, for example:

```
GET /api/v1/cases/{caseId}/records/ABC-123
```

Would mean "Get the record that has the Document ID of "ABC-123".

Key values used to identify resources are not case-sensitive. Specifying a key value of "abc" will match a resource with a Key value of "Abc" or "ABC".

When using Key values in URLs, any spaces within the key value must be replaced with the + (plus) character.

Issues differentiating IDs and Keys

There are some situations where EDT cannot tell whether the value is an ID or a Key:

- Where the resource type uses strings for IDs. There aren't very many of these, but the User resource is one such example.
- Where your Key value is numeric. In this scenario EDT cannot tell if the value is an ID or Key and will assume it is an ID.
- Where your Key value contains a colon. (We suggest you keep non-alphanumeric characters out of your key values for the time being)

In these situations, you can explicitly tell EDT that the value in the route is a Key value by adding the "key:" prefix to the value. For example, if the key value for a Person is the numeric value 99, then to ensure EDT interprets the value as a key use the prefix:

```
GET /api/v1/org-units/1/persons/key:99
```

In the same way, all User lookups using a key must always include the "key:" prefix since both the ID and Key values are strings.

```
GET /api/v1/users/key:PID1234
```

Where can Keys currently be used?

It will take some time for all EDT APIs to be updated to support Keys. For resources that can be specified using a key in the URL, the documentation of the endpoint will say "{idOrKey}" rather than just "{id}". The resource's type might also be added as a prefix in situations where the route has multiple ids/keys. Examples:

```
GET /api/v1/cases/{caseId}/records/{idOrKey}
GET /api/v1/cases/{caseId}/records/{recordIdOrKey}
```

Updating a resource using ID or Key

In some EDT APIs a resource is updated using a route that does not include the resource's identifier. For example, updating a Person previously used the following route and relied on the model in the request body containing the ID of the person resource to be updated.

```
PUT /api/v1/persons
```

This has now been changed so the ID or Key is specified in the route:

```
PUT /api/v1/persons/{idOrKey}
```

This will allow us to:

- Explicitly state whether we want to locate the resource by ID or Key.
- Update the resource's Key by specifying a new Key value in the model/body.
- Error if the ID in the route is different from the one in the model/body.

Over time other update (PUT) routes will be updated in a similar way.

Supported Characters in Keys

Certain characters cannot be used in URL paths so should not be used in Key values.

Over time we will define and enforce the set of supported characters and also provide approaches on how to include unsupported characters.

For the time being, please limit your key values to alphanumeric characters, space, hyphen, and underscore. When using a key value containing spaces in URL paths, replace the spaces with a plus (+) character.

Summary of API endpoints

Important: All endpoints in the EDT Core API, apart from those listed in this document, are not "published" and should not be used.

GetVersionInfo	GET /api/v1/version
GetCases	GET /api/v1/org-units/{orgUnitId}/cases
GetOrgUnitFields	GET /api/v1/org-units/{orgUnitId}/fields
GetIdByKeyOrField	GET /api/v1/org-units/{orgUnitId}/cases/{keyOrField}/id
CreateNewCase	POST /api/v1/org-units/{orgUnitId}/cases
GetCaseDetail	GET /api/v1/cases/{caseId}
UpdateCase	PUT /api/v1/cases/{caseId}
MergeCase	POST /api/v1/cases/{caseId}/export-to-case/merge-case/{destCaseId}
SyncCaseUsers	POST /api/v1/cases/{caseId}/users/sync
SyncUserCases	POST /api/v1/org-units/{orgUnitId}/users/{userIdOrKey}/cases/sync
GetCaseUsers	GET /api/v1/cases/{caseId}/users
GetUserCases	GET /api/v1/org-units/{orgUnitId}/users/{userIdOrKey}/cases
AddCaseUser	POST /api/v1/cases/{caseId}/users/{userId}
RemoveCaseUser	DELETE /api/v1/cases/{caseId}/users/{userId}
GetGroups	GET /api/v1/cases/{caseId}/groups
GetGroup	GET /api/v1/cases/{caseId}/groups/{groupId}
SyncGroupUsers	POST /api/v1/cases/{caseId}/groups/{groupId}/sync
GetCaseGroupsOfUser	GET /api/v1/cases/{caseId}/users/{userId}/groups
AddUserToCaseGroup	POST /api/v1/cases/{caseId}/groups/{groupId}/users/{userId}
RemoveUserFromCaseGroup	DELETE /api/v1/cases/{caseId}/groups/{groupId}/users/{userId}
CreatePerson	POST /api/v1/org-units/{orgUnitId}/persons
GetPersons	GET /api/v1/org-units/{orgUnitId}/persons <i>[Deprecated in 11.6.8]</i>
UpdatePerson	PUT /api/v1/org-units/{orgUnitId}/persons/{personIdOrKey}
AddCaseParticipant	POST /api/v1/cases/{caseId}/participants
<i>Added May 2023</i>	
SyncCaseParticipants	POST /api/v1/cases/{caseId}/participants/sync
GetCaseParticipants	GET /api/v1/cases/{caseId}/participants
GetParticipantTypes	GET /api/v1/org-units/{orgUnitId}/participant-types
GetRecords	GET /api/v1/cases/{caseId}/records
CreateRecord	POST /api/v1/cases/{caseId}/records
UpdateRecordFile	PUT /api/v1/cases/{caseId}/records/{recordId}/{fileType}
UpdateRecord	PUT /api/v1/cases/{caseId}/records/{idOrKey}
DeleteRecord	DELETE /api/v1/cases/{caseId}/records/{idOrKey}
DeleteAllRecords	DELETE /api/v1/cases/{caseId}/records
GetUsers	GET /api/v1/users
CreateUser	POST /api/v1/users
GetUser	GET /api/v1/users/{userId}
	GET /api/v1/users/key:{key}
UpdateUser	PUT /api/v1/users
GetOrgUnitGroups	GET /api/v1/org-units/{orgUnitId}/groups
AddUserToOrgUnitGroup2	POST /api/v1/org-units/{orgUnitId}/groups/{ouGroupId}/users
RemoveUserFromOrgUnitGroup2	DELETE /api/v1/org-units/{orgUnitId}/groups/{ouGroupId}/users/{userId}
	DELETE /api/v1/org-units/{orgUnitId}/groups/{ouGroupId}/users/key:{key}
GetUserOuGroups	GET /api/v1/org-units/{orgUnitId}/users/{userIdOrKey}/groups
<i>Added in 11.6.2:</i>	
GetIdentifiers	GET /api/v1/org-units/{orgUnitId}/identifiers
GetIdentifierById	GET /api/v1/identifiers/{identifierId}
GetIdentifiersForEntity	GET /api/v1/org-units/{orgUnitId}/identifiers/{entityType}/{entityId}/{identifierType}
CreateIdentifier	POST /api/v1/org-units/{orgUnitId}/identifiers
UpdateIdentifier	PUT /api/v1/identifiers
DeleteIdentifier	DELETE /api/v1/identifiers/{identifierId}
RemoveCaseParticipant	DELETE /api/v1/cases/{caseId}/participants/{personIdOrKey}
CaseIdsLookup	POST /api/v1/org-units/{orgUnitId}/cases/lookup-ids

```
ReassignParticipantCases POST /api/v1/org-units/{orgUnitId}/persons/{fromPersonIdOrKey}/
                                reassign-cases/{toPersonIdOrKey}
```

11.6.3:

```
GetSiteCaseInfo      GET /api/v1/org-units/{orgUnitId}/cases/info/{caseId}
    Not new in 11.6.3, but on 10 Nov 2023 Elaine at Maple was told
                        that they can use this to safely get the status of cases being
                        created.
```

Added in 11.6.5:

```
DeleteRenditions      DELETE /api/v1/cases/{caseId}/records/{idOrKey}/renditions
```

Added in 11.6.8

```
GetPersons (v2)      GET /api/v2/org-units/{orgUnitId}/persons
```

Added in 11.8

```
CreateNotification    POST /api/v1/org-units/{orgUnitId}/notifications
GetPlatformEvents     GET /api/v1/platform-events
```

Added in 11.9:

```
CreateLoadFileImport  POST /api/v1/cases/{caseId}/create-loadfile-import
GetImport              GET /api/v1/cases/{caseId}/imports/{importId}
```

Endpoint Details

The following sections contain notes about the APIs that have been made available to Maple. This information will tend to be high level information, with more detailed information available via the Swagger UI.

Cases

Getting a list of Cases

The following gets a list of the cases in Org Unit 1 (the default OU).

```
GET /api/v1/org-units/1/cases
```

Case Custom Fields

To get information about the custom fields defined for Cases, call the following endpoint and then filter the results by objectType = "Case":

```
GET /api/v1/org-units/{orgUnitId}/fields
```

Note: the previous route, `GET /api/v1/site/fields?organisationalUnitId=1` has been deprecated. Do not use. This will be removed.

GetIdByKeyOrField - Looking up a Case ID using the key or a custom field

Case IDs are used extensively in EDT APIs. It will take some time for all routes to be updated to support specifying a Case Key instead of an ID. In the meantime, an endpoint has been added that supports efficiently looking up a Case's using its Key or the value of a custom field:

```
GET /api/v1/org-units/1/cases/{keyOrField}/id
```

To search by:

- Key. Use "key:abc" or "key:123" as the keyOrField value. Specifying the "Key:" prefix is optional, so the following are equivalent to the first 2 examples respectively: "abc" or "23"
- Custom field. Specify the field's id or name, a colon, and the value. E.g. to find the Case where the Colour custom field with ID 123 has the value blue, use: "Colour:blue" or "123:blue" as the keyOrField value.

The endpoint returns an array of Case ID, Key, and Status values. So, for example, if a single Case is found, the response body is something like:

```
[
  {
    "id": 3,
    "key": "ABC",
    "status": "Active"
  }
]
```

The Status field was added to the response model in EDT 11.6.8.

If no cases are found that match, an empty array is returned.

The lookup using a custom field also works for multi-valued custom fields. For example, if there is a MV field called "OtherRccs" that has an ID of 12, then to find which case has the RCC "myRcc55" in the MV field you can use either of the following:

```
GET /api/v1/org-units/1/cases/OtherRccs:myRcc55/id
GET /api/v1/org-units/1/cases/12:myRcc55/id
```

CaseldsLookup

Added in 11.6.2.

```
POST /api/v1/org-units/{orgUnitId}/cases/lookup-ids
content-type: application/json
{{authorization}}

{
  "keys" : ["enron", "api-1", "api-2", "api-4"]
}
```

Response

- Returns the Id and Key of each case.

Updated in 11.8: Now also returns the Status of each case.

5.2.1 Creating a Case

Example:


```
POST /api/v1/org-units/1/cases
Authorization: Bearer <secret>

{
  "name": "Case name goes here",
  "key": "ABC-123",
  "description": "First case creation API test",
  "timeZoneId": "Pacific Standard Time",
  "templateCase": "12"
}
```

For cases, the custom properties are set using the field ID and value (but see update below) ².

The following partial example shows how to set custom property values for cases:

```
{
  "fields" : [
    { "id": "23", "value": "Returned" },
    { "id": "25", "value": 1234 }
  ]
}
```

Update 7 Oct 2022: *fields can be specified using either the ID or Display Name.* The first field in the following example shows how to use a field's name instead of the field ID. The second field specifies both ID and Display Name, and in this scenario the Display Name is ignored and the field is located by ID.

```
"fields" : [
  { "name": "My Case Status", "value": "Returned" },
  { "id": "25", "name": "DOB", "value": "2002-05-15" }
]
```

Update 7 Oct 2022: *list field and MV list field values can be specified using either the value ID or textual value.*

This example specifies a MV field and its list values using IDs:

```
{
  "id": 28,
  "value": [ { "id": 12 }, { "id": 16 } ]
}
```

This example uses a MV field's display name and the textual values of its list items:

```
{
  "name": "Flags",
  "value": [ "Indigenous", "Level 5" ]
},
```

If a list item or multi-value list item is specified using text and the value doesn't exist already, a new list item with that value will automatically be created.

5.2.2 Updating the metadata of a Case

To update the metadata on a case, you would typically retrieve the existing metadata using the GetCaseDetail endpoint³:

² At present, the way custom property values are specified is changing. This is the reason why specifying custom properties for Cases differs from the way it is done for Persons.

³ Note that calling the GetCaseDetail endpoint before the case has finished creating will result in a 500 error code.

```
GET /api/v1/cases/4007
```

And then update the JSON as required, and send the changed model using a PUT:

```
PUT /api/v1/cases/4007
content-type: application/json

{
  "name": "API Test Case",
  "key": "ABC-123",
  "caseNumber": "123",
  "description": "This is a test case",
  "category": "API Template",
  "status": "Active",
  "cfsLocationType": 1,
  "exportTempPath": "C:\\Temp",
  "stagingAreaPath": "\\edt-dev-staging\\QA-V10\\Site01_Case67115",
  "timeZoneId": "E. Australia Standard Time",
  "removeGeneratedFiles": false,
  "searchHistoryRetention": 10,
  "fields": [
    {
      "id": "23", "value": "On hold"
    }
  ],
  "dateFormat": "d MMM yyyy",
  "showMilliseconds": true
}
```

5.2.3 Merging an Agency File or a Court File – DRAFT

MergeCase

All documents require a Document ID because EDT uses Document IDs in the destination case.

EDT will create all Folders and Locations as needed in the destination case.

A prefix can be specified to use a Case field which can be custom by encapsulating the field with %FieldName% e.g. %CaseId%

```
POST /api/v1/cases/{caseId}/export-to-case/merge-case/{destinationCaseId}
{
  "prefix": "string"
}
```

The prefix parameter must not contain characters that are illegal file system characters. It is highly recommended that it consists of only: hyphens, underscores, and alphanumeric characters. It is recommended that spaces are not included.

Case Users

5.3.1 Adding & Removing Users to a Case

Update the users assigned to a Case using the SyncCaseUsers endpoint.

The request body specifies all the users that should be assigned to the Case. Any existing users assigned to the case that are not included in the request body will be removed from the Case. Any users identified in the request body that are not currently assigned to the case will be added.

Example of syncing 5 users to a Case; 2 users using their IDs and 3 users using their Keys:

```

POST /api/v1/cases/{caseId}/case-users/sync [DEPRECATED]
POST /api/v1/cases/{caseId}/users/sync [NEW]

content-type: application/json

{
  "userIds": ["05f8995d-522c-426a-b752-14312d49f7bc", "15a19095-a360-4d7b-ba14-edcab7d3a94e"],
  "userKeys": [ "PI1234", "PI1235", "PI1236" ]
}

```

For each user, specify either their id or key (if they have one).

UserIds	<p>An array of string values.</p> <p>The Ids of the Users to be synced.</p> <p>Note: Almost all resources in EDT have numeric IDs. User accounts are one exception that uses string IDs.</p>
UserKeys	<p>An array of string values.</p> <p>The keys of the Users to be synced.</p>

Example:

To remove all user from the Case:

```

POST /api/v1/cases/{caseId}/case-users/sync [DEPRECATED]
POST /api/v1/cases/{caseId}/users/sync [NEW]

content-type: application/json

{
  "userIds": [],
  "userKeys": []
}

```

Extra scenario – Add a user to multiple cases - SyncUserCases

Example:

```

POST /api/v1/org-units/1/users/key:PID-1234/cases/sync
content-type: application/json

{
  "caseField": "ID",
  "values": ["2004", "4007"],
  "caseGroups" : ["Reviewers","Missing", "Case Manager"]
}

```

Parameter	Description
caseField	Required. Valid values are: "ID", "Key", or "<custom-column-id>".
values	<p>The values that indicate which cases to sync. The value depends on the caseField parameter. For example, if caseField is "ID" then the values must be an array of case IDs.</p> <p>To remove the user from all cases, make the values parameter an empty array.</p>

caseGroups	Optional. The groups within each case that the user should be added to. The caseGroups parameter is only used when the user is added to a case. That is, if they are already a case member then their existing group membership will not change. If a specified group is not found within a case, it is ignored.
------------	--

Removing a user from a case removes them from all the groups in the case.

Extra scenario – GetUserCases – Get a list of the cases a specified user has access to

Added 11 Oct 2022.

Returns a list of the Active and Inactive cases the specified user can access within the specified OrgUnit.

```
GET /api/v1/org-units/{orgUnitId}/users/{userId}/cases
Or
GET /api/v1/org-units/{orgUnitId}/users/key:{userKey}/cases
```

Examples:

```
### UserCases - using user id
GET /api/v1/org-units/1/users/1c34184d-b763-4ebe-b86d-efd2a4b01c29/cases

### UserCases - using user key
GET /api/v1/org-units/1/users/key:PID-1234/cases
```

Returns an array of cases. For each case, the case ID and case Key are included. For example, if the specified user belongs to 2 cases the result will be something like:

```
[
  {
    "id": 2004,
    "key": "PCH"
  },
  {
    "id": 8032,
    "key": null
  }
]
```

As mentioned, the GetUserCases endpoint returns Active and Inactive cases by default. An optional caseStatus query string parameter can be included to return cases with a specific status:

```
GET /api/v1/org-units/1/users/key:PID-1234/cases?caseStatus=removed
```

Case Groups

Planned change: Include the Key property

Supporting endpoint: Get the Groups in a Case

```
GET /api/v1/cases/{caseId}/groups
```

Returns summary information about each Group.

Changed in EDT 11.6.8: For performance reasons, the response model no longer includes counts by default for each

group. To include the counts, add a `includeCounts=true` query string parameter.

Supporting endpoint: Get a Group's details

```
GET /api/v1/cases/{caseId}/groups/{groupId}
```

Returns more detailed information about the specified Group, including its members.

Changed in EDT 11.6.8: The response model no longer includes counts by default. To include them, add a `includeCounts=true` query string parameter.

Updating a group's users using the SyncGroupUsers endpoint (ISL)

Planned change: Revise to support using the Key property

Replaces all the Users assigned to a Group.

Example using the standard `FullName` field of User resources:

```
POST /api/v1/cases/4007/groups/1/sync
content-type: application/json

{
  "keyField": "FullName",
  "values": [ "Brian Griffin", "Meg Griffin", "Chris Griffin", "Lois Griffin",
             "Peter Griffin", "Stewie Griffin", "Missing Griffin" ]
}
```

KeyField	<p>String. The default is "Id".</p> <p>The name of the User field to use. The field can either be a standard or custom field of the User resource.</p> <p>The value is case-insensitive.</p>
Values	<p>An array of string values.</p> <p>Case Users who have one of the specified values in their <i>KeyField</i> will be added to the Group, all other users will be removed from the Group.</p> <p>The values are case-insensitive.</p> <p>If a specified value does not match to a Case User, the value is ignored. At this stage no error is raised.</p>

Updating a group's users using AddUserToCaseGroup and RemoveUserFromCaseGroup (DIAM)

This section was added 16 Mar 2023.

AddUserToCaseGroup

```
POST /api/v1/cases/{caseId}/groups/{groupId}/users/{userId}
```

RemoveUserFromCaseGroup

```
DELETE /api/v1/cases/{caseId}/groups/{groupId}/users/{userId}
```

Neither request includes a body. Both return responses of 204 (No content) if successful. See Swagger UI for more information about other possible return values.

These endpoints do not currently support the use of Key fields to identify case or user resources.

Getting the groups of a user

GetCaseGroupsOfUser

```
GET /api/v1/cases/{caseId}/users/{userId}/groups
```

Participants/Persons

All examples assume the Person resources are in the default OrgUnit. That is, the OrgUnit with an ID of 1.

Users and Persons are different. Users are system objects that are used to specify who can access the system. Persons are business objects that can be used to record people of interest who do not have access to the system. For example, prosecutors, defence, and accused persons.

5.4.1 Adding a Person (previously called a Participant)

To add a (DEMS) Participant, you create a *Person* resource in the Persons collection.

Below is an example of the minimum required to create a Person resource.

```
POST /api/v1/org-units/1/persons
content-type: application/json

{
  "address": {
    "email": null
  },
  "fields": []
}
```

Here is a fuller example:

POST /api/v1/org-units/1/persons
content-type: application/json

```
{
  "key": "pid-1234",
  "name": "D Duck",
  "firstName": "Donald",
  "lastName": "Duck",
  "address": {
    "email": "dduck@test.com",
    "mobile": "0413 999 999",
    "phone": "555-123-345",
    "postalAddress": null,
    "postalAddressAsJson": null,
    "isPrivate": false
  },
  "title": {
    "id": 0,
    "title": "Mr."
  },
  "fields": [
    { "name" : "2", "value" : "He/him" },
    { "name" : "PartID", "value" : "abcdef" },
    { "name" : "DOB", "value" : "1 Jan 2000" },
    { "name" : "Priority List", "value" : "High" }
  ],
  "status" : "Active",
  "category" : "Other",
  "dob" : "2000-12-16"
}
```

Fields	<p>An array of custom fields.</p> <p>Any fields specified must already exist.</p> <p>Each field has a 'name' and 'value'. When setting field values, the "name" can be the display name of the field or the (internal) ID of the field. The value specified must be compatible with the field's type.</p>
Title	<p>If title.id is 0 (or not provided), EDT will look up the title.title (eg "Mr.") and set the ID automatically. If the specified title.title does not exist, EDT will create it.</p>
Status	<p>Optional. Valid values are "Active" and "Inactive". The default is "Active".</p>
Key	<p>Optional. If provided, it must be unique within the site.</p>
Email	<p>If the Address.Email value is not null, the value must be unique within the Org Unit.</p>
DOB	<p>Optional. The value must be a valid date. The value must be a date or datetime using the format "2025-03-12" or "2025-03-12T00:00:00". If a time is included in the value, it will be ignored.</p> <p>Added in API 11.8.0.203.</p>
Category	<p>Optional. The value must be the relevant CategoryType.ID, not the CategoryType.Description. (See Site Settings > Fields > Person > CategoryType).</p> <p>The value is case-insensitive. For example, specifying a value of "other" will match the "Other" CategoryType.ID.</p> <p>Added in API 11.8.0.203.</p>

Using the Fields property

The Fields property is an array of the custom fields of the Person resource.

The Fields property does not work in quite the same way as the Fields properties on some other resources such as Cases. The main difference is that cannot specify an "id" property on a Field, However, you can enter the field's ID in the "name" property as an alternative to the field name. For example, if you have a custom field called "Colour" that has a field ID of 3, you could set its value using either:

```
{ "name": "3", "value": "Red" }
```

or

```
{ "name": "Colour", "value": "Red" }
```

When updating a Person, all custom fields of the resource must be specified, otherwise those not included will be deleted.

Proposed change

The Fields property is an array of the custom fields of the Person resource.

The Fields property does not work in quite the same way as the Fields properties on some other resources such as Cases. The main difference is that cannot specify an "id" property on a Field, However, you can enter the field's ID in the "name" property as an alternative to the field name. For example, if you have a custom field called "Colour" that has a field ID of 3, you could set its value using either:

```
{ "name": "3", "value": "Red" }
```

or

```
{ "name": "Colour", "value": "Red" }
```

When updating a Person, all custom fields of the resource must be specified, otherwise those not included will be deleted.

Supporting endpoint: GetPersons (v1) - Deprecated

Get all the active Person resources.

```
GET /api/v1/org-units/1/persons
```

Deprecated in EDT 11.6.8. This is version 1. Use version 2.

Supporting endpoint: GetPersons (v2)

New in EDT 11.6.8.

```
GET /api/v2/org-units/1/persons
GET /api/v2/org-units/1/persons?page=2&pagesize=25&filter=dob:1966-12-31
```

Query Params

Parameter	Description
Filter	Filter on Person fields. The available fields are: <ul style="list-style-type: none">Standard fields: "Id", "FirstName", "LastName", "Key", "Name", "Status", "Email", "Category" (added in 11.8), "DOB" (added in 11.8). See the <i>Important</i> note below.Custom fields.

Separate the fields to search using a comma. e.g.

```
.../persons?filter=lastname:bob,dob:1966-12-31
```

Page The page of results to display. Defaults to 1.

PageSize The number of records to return on each page. Defaults to 50.

IncludeInactive The endpoint returns Active persons by default. Use this parameter to also include Inactive persons in the results. Defaults to False.

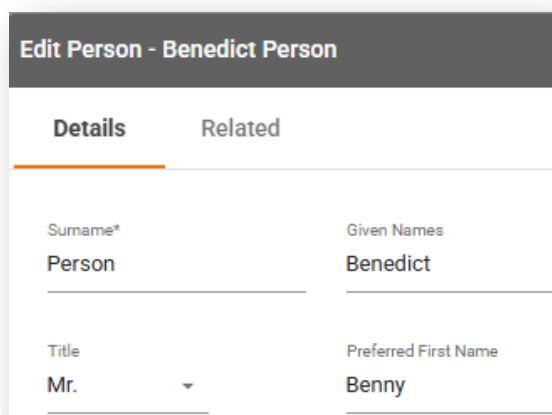
```
.../persons?IncludeInactive=true
```

Important Note: Field name inconsistency when filtering in 11.8+:

There is an issue/inconsistency when filtering in the GetPersons (v2) endpoint:

- Specifying FirstName in the filter parameter filters on the Preferred First Name UI field. To be consistent with this and other endpoints, it should be filtering on the Given Names UI field.
- Specifying GivenNames in the filter parameter filters on the Given Names UI field. To be consistent, this should not be an option as the Given Names field was not a known field prior to 11.8.

For example, given the following Person in the UI:



Calling the GetPersons endpoint with either of the 2 filters⁴ below will return the person above:

```
.../persons?filter=givennames:bene
```

or

```
.../persons?filter=firstname:benn
```

The response from the endpoint (these are expected results):

- Includes the Given Names from the UI in the FirstName response property.
- Does not include the Preferred First Name value in the response.

⁴ Filtering on a string field does a partial match, not an exact match. So, `filter=givennames:bene` will match on "Benedict" and "xxxbenexxx".

```
{
  "items": [
    {
      "id": 4,
      "key": "5",
      "name": null,
      "title": {
        "id": 0,
        "title": "Mr.",
        "organisationalUnitId": null
      },
      "firstName": "Benedict",
      "lastName": "Person",
      "address": {
```

Supporting endpoint: Get the details of a Person resource using an ID and using a Key

```
GET /api/v1/org-units/1/persons/8
GET /api/v1/org-units/1/persons/pid-1234
```

5.4.2 Updating a Person (previously called a Participant)

EDT 11.6.8 and earlier

When updating a Person, the highlighted properties below are checked and updated if changed. If you exclude any of the highlighted properties from the request body, they will be changed to null rather than being ignored. For this reason, it is good practice to retrieve the Person details using a GET, change just the relevant properties, then update the resource using the PUT as shown here.

```
PUT {{host}}/api/v1/org-units/1/persons/{personIdOrKey}
content-type: application/json
```

```
{
  "id": 23,
  "key": "KS",
  "name": "K Simpson",
  "title": {
    "id": 8,
    "title": "Mrs."
  },
  "firstName": "K",
  "lastName": "Simpson",
  "address": {
    "id": 1025,
    "email": "k_simpson@test.com",
    "mobile": "0123123123",
    "phone": "98766789",
    "postalAddress": null,
    "postalAddressAsJson": null,
    "isPrivate": false
  },
  "orgs": [
    {
      "organisationId": 23,
      "personId": 23,
      "organisationTypeId": 1,
      "organisationMemberId": 23,
      "startDateTime": "2022-07-07T03:11:26.29",
      "endDateTime": null
    }
  ],
}
```

```

"cases": [],
"fields": [
  { "name": "2", "value": "partid-3333" },
  { "name": "Colour", "value": "Red" },
  { "name": "Priority", "value": "High" }
],
"displayName": "K Simpson",
"organisationTypeId": 1,
"permissions": null,
"status": "Active"
}

```

Other Notes:

- You cannot change the details of a specific organisation using this endpoint. However, you can change which organisations the Person belongs to.
- In the example above, the Priority custom field is a List. This illustrates how to set the value of a list.

EDT 11.8

EDT 11.8 EDT moved to a new Person/Orgs schema. The existing API endpoints were updated to use this model but the behaviour changed in some ways.

In 11.6.8, whenever a Person was created, a “shadow org” was created. (see the example above). This is no longer true in 11.8. The orgs property in the UpdatePerson request model, if provided, is now ignored.

The new Person schema does not include a “Name” property. The existing API endpoints still include this property in the request/response models. Response models will always return null for Name. Attempting to update the Name property with any non-null value will result in a 400 response. Use Firstname and Lastname instead.

The behaviour of the Fields array, which is used to update custom fields, has changed:

- If you do not want to update any custom field values, either do not specify a fields property at all, or use `"fields": []`
- To update custom field values, only specify the values of the fields to be updated in the `fields` array.

In 11.6.8, not providing values for the request model properties highlighted in yellow above, would set their values to null. This is no longer true in 11.8. Not providing these properties will not update the value.

Example: To update only the custom field called “OTC” to “1234”

```

PUT {{host}}/api/v1/org-units/1/persons/{personIdOrKey}
content-type: application/json

{
  "fields": [
    { "name": "OTC", "value": "1234" }
  ]
}

```

Example: To update a Person’s first name (which appears as “Given Names” in the UI), email address, and Key:

```

PUT {{host}}/api/v1/org-units/1/persons/{personIdOrKey}
content-type: application/json

{
  "firstname": "Mary",
  "address": {
    "email": "m_simpson@test.com"
  },
  "key": "ABC-1234"
}

```

Update in build 11.8.0.203 of the API - 12 Mar 2025

Two new standard fields were added to the UpdatePerson request model: dob and category.

If the dob or category fields are not specified, the respective values will not be updated.

There is currently no way to clear/remove the values from these fields.

When setting the dob field:

- The value must be a valid date.
- The value must be a date or datetime using the format "2025-03-12" or "2025-03-12T00:00:00". If a time is included, it will be ignored.
- If the value provided is not valid, a 400 response is returned.

When setting the category field:

- The value must be the relevant CategoryType.ID, not the CategoryType.Description. (See Site Settings > Fields > Person > CategoryType)
- The value is case-insensitive. For example, specifying a value of "other" will match the "Other" CategoryType.ID.
- If the value provided is not valid, a 400 response is returned.

5.4.3 Add a Person (Participant) to a Case

AddCaseParticipant

Add a Person to a Case. That is, make them a case participant.

Example:

```
POST {{host}}/api/v1/cases/{caseId}/participants
content-type: application/json

{
  "personIdOrKey" : "sgriffen",
  "participantType": "Accused"
}
```

SyncCaseParticipants

To add/remove persons to/from a Case, use the SyncCaseParticipants endpoint:

```
POST /api/v1/cases/{caseId}/participants/sync
content-type: application/json

{
  "Participants": [
    { "personIdOrKey": "MS", "participantType": "Adjudicator" },
    { "personIdOrKey": "26", "participantType": "Accused" }
  ]
}
```

Notes:

- The specified Case and Persons must belong to the same OrgUnit.

To remove all participants from a case:

```
POST /api/v1/cases/1/participants/sync
content-type: application/json

{
  "Participants": [ ]
}
```

Added functionality 19 Sep 2022:

The SyncCaseParticipants endpoint can also take an optional "ParticipantTypeFilter" setting. This can be used to sync just the participants of the specified type, with all the other participants on the case remaining untouched.

So, the API can now sync the participants in one of 2 modes:

- All (individual) participants on the case. Specify each person and their participant type.
- Participants of a specified type on the case. Use the *participantTypeFilter* to specify the type of participants to sync. If provided, all participants listed in the participants collection of the request must also specify the same type.

Example 1: Set two *Accused* participants on the case. All other types of participants on the case remain untouched:

```
POST {{host}}/api/v1/cases/1/participants/sync
content-type: application/json

{
  "ParticipantTypeFilter" : "Accused",
  "Participants": [
    { "personIdOrKey": "KS", "participantType": "Accused" },
    { "personIdOrKey": "26", "participantType": "Accused" }
  ]
}
```

Example 2. Remove all *Accused* participants on the case. All other types of participants on the case remain untouched:

```
POST {{host}}/api/v1/cases/1/participants/sync
content-type: application/json

{
  "ParticipantTypeFilter" : "Accused",
  "Participants": [ ]
}
```

5.4.4 Remove a Participant from a Case

RemoveCaseParticipant endpoint. Added in 11.6.2.

```
DELETE /api/v1/cases/{caseId}/participants/{personIdOrKey}
```

Note: Prior to 11.6.2 the SyncCaseParticipants endpoint was the defined endpoint to remove case participants, as described above in 5.4.3 *Add a Participant to a Case*.

Supporting endpoint: GetCaseParticipants

```
GET /api/v1/cases/{caseId}/participants
```

Optional filters can be added to limit the results to a specific Person (ID or Key) and/or ParticipantType.

Examples of filtering the Case Participants:

```
GET /api/v1/cases/1/participants?person=1234&participantType=Accused
GET /api/v1/cases/1/participants?person=key:NS&participantType=Accused
GET /api/v1/cases/1/participants?participantType=Accused
```

Supporting endpoint: GetParticipantTypes

```
GET /api/v1/org-units/{orgUnitId}/participant-types
```

5.4.5 Merging Person (previously called Participant) Details

API endpoint added in EDT 11.6.2.

Responsibilities:

- EDT provides an API endpoint to reassign the cases, and associated data within each case, to another Person and deactivate the redundant Person.
- It is the responsibility of the client code to handle other tasks such as:
 - Copying across any field values from the redundant Person that are to be kept.
- Recommended: Remove, or rename, the Identifiers used by export-to-participant (EdtExternalId & DisclosurePortalCase) from the redundant Person. While the redundant person will be deactivated by the ReassignParticipantCases endpoint, if the Person is ever reactivated, assigned to a case, and an export-to-participant is run, the results may be unintended.

Endpoint: **ReassignParticipantCases**

```
POST /api/v1/org-units/{orgUnitId}/persons/{fromPersonIdOrKey}/reassign-cases/{toPersonIdOrKey}
```

When called, this endpoint:

1. Looks up all the cases the "from" Person is assigned to, and for each of those cases:
 - Looks for exports where the fromPersonId is assigned to the export and reassigns them to the toPersonId.
 - Looks for exports-to-participant the fromPersonId and reassigns them to the toPersonId.
 - Reassigns the Person assigned to the case from the fromPersonId person to the toPersonId person.
2. Deactivates the "from" Person.

Supporting endpoints – Identifiers

GetIdentifiers

```
GET /api/v1/org-units/{orgUnitId}/identifiers
```

Query Params

Note: These query params use the term "Item" whereas other endpoints for Identifiers use the term "Entity" to refer to the same thing.

Parameter	Description
Filter	Filter on Identifier fields. The available fields are: "ItemType", "ItemId", "IdentifierType", "IdentifierValue". Separate the fields to search by with a comma. e.g.
<pre>...?filter=itemType:Person,itemId:1,identifierType:EdtExternalId</pre>	

Page	The page of results to display. Defaults to 1.
PageSize	The number of records to return on each page. Defaults to 50.
Sort	<p>The field or fields used to sort the results. The available fields are: "ItemType", "ItemId", "IdentifierType", "IdentifierValue".</p> <p>By default, the results are in ascending order but the direction can be specified explicitly using <code>asc</code> and <code>desc</code>.</p> <p>For example, to sort by ItemId (Entity ID) and then by descending ItemType (Entity type), use:</p> <pre>sort=itemId,itemType desc</pre>

GetIdentifierById

```
GET /api/v1/identifiers/{identifierId}
```

GetIdentifiersForEntity

```
GET /api/v1/org-units/{orgUnitId}/identifiers/{entityType}/{entityId}/{identifierType}
```

CreateIdentifier

```
POST /api/v1/org-units/{orgUnitId}/identifiers
```

Request Body:

```
{
  "entityType": "Person",           // Must be "Person"
  "entityId": 0,
  "identifierType": "string",       // eg "EdtExternalId"
  "identifierValue": "string",      // Provide this OR
  "autoIncrementOptions": {        // ... this
    "format": "string",            // Specifies the number of numeric chars eg "000000"
    "prefix": "string"            // Optional
  }
}
```

Provide either identifierValue or autoIncrementOptions

UpdateIdentifier

```
PUT /api/v1/identifiers
```

Request Body:

```
{
  "id": 0,          // Required. This is the Identifier ID, not the Person ID
  "identifierValue": "string", // Provide this, OR
  "autoIncrementOptions": {   // ... this
    "format": "string",       // Specifies the number of numeric chars eg "000000"
    "prefix": "string"        // Optional
  }
}
```

Provide either identifierValue or autoIncrementOptions.

DeleteIdentifier

```
DELETE /api/v1/identifiers/{identifierId}
```

Records

Searching for Records

New – 11 Nov 2022.

The GetRecords endpoint returns the records in the case:

```
GET /api/v1/cases/2004/records
```

The results of the GetRecords endpoint can be controlled using 5 optional query string parameters:

Parameter	Description
filter	<p>A filter to apply to the records in the case. The syntax matches that of the Search Bar in the UI. E.g.</p> <pre>DocKind:Email AND EDTID:>=200000 AND DocumentID:NOT []</pre> <p>The example returns Emails where the EDT ID is greater or equal to 200,000 and where the DocumentID has a value i.e. is not null.</p> <p>Field names: fields in the filter are specified using their display names.</p>
fields	<p>Additional fields to return for each record. If not specified, a set of standard fields plus Title are returned.</p> <p>Field names: Some fields must be specified using their back-end name rather than their display name. See <i>Using field names in filter, sort, and fields parameters</i> below.</p>
sort	<p>The sort order and direction of the results. Multiple sort fields can be specified and the sort direction ("asc" or "desc") can be specified for each one.</p> <p>The following example sorts by results by Title ascending, then by EDT ID in descending order.</p> <pre>sort=title,edtid desc</pre> <p>The following are identical because ascending order is the default sort order.</p>


```
sort=title,edtid
sort=title asc,edtid asc
```

Field names: Some fields must be specified using their back-end name rather than their display name. See *Using field names in filter, sort, and fields parameters* below.

page	The page of results to display. The default is 1.
pagesize	The number of documents per page of results. The default is 50 records per page.

Using field names in filter, sort, and fields parameters

Some fields must be referred to differently, depending on whether they are specified in the Filter or Sort/Fields parameters. The display name of fields is required in the Filter parameter whereas the back-end name of fields is often required in the Sort/Fields parameters. Custom fields are a prime example. For example, the "Relevant" custom field is specified as 'Relevant' in the Filter parameter, but as 'cc_Relevant' in the Sort and Fields parameters, and in the results.

Examples

Get "all" the records in case 2004. This actually returns the first 50 records, because the default page size is 50.

```
GET /api/v1/cases/2004/records
```

Example 1 of explicitly specifying some search parameters:

```
GET /api/v1/cases/2004/records?filter=DocKind:Email AND EDTID:>=200000&sort=Title
desc&fields=Title,cc_Relevant
```

Example 2 of explicitly specifying some search parameters:

```
GET /api/v1/cases/2004/records?filter=Title:(edt trial)&sort=title
desc&pagesize=1000&page=4
```

Using Keys with the Record APIs

The Key for records is the "DocumentID" property. That is, unlike other types of resources that have a dedicated "Key" property, the record's DocumentID property is used when referencing the record resource by key.

The Record endpoints can specify a record by ID or Key. The following are examples of how to specify a record using its key:

```
DELETE /api/v1/cases/{caseId}/records/key:API-001
DELETE /api/v1/cases/{caseId}/records/API-001
```

(The 2nd example can be used because the shown DocumentID's value is not all digits and therefore will not be mistaken for an EDT ID.)

5.5.1 Adding a Record

Adding a record requires two API calls, one to create the Record resource and a second to upload the Record's *native* file. The native file is original file in its native format. For example, a DOCX or PDF file.

Part 1 - Create the record:

```

POST /api/v1/cases/{caseId}/records
content-type: application/json

{
  "title": "Share trades",
  "custodian": "Donald Duck",
  "location": "DDuck laptop - Deleted Items",
  "folder": "D Duck",
  "documentId": "API-001",
  "fileExtension": ".pdf",
  "primaryDateUtc": "2022-06-01T07:03:07.089Z",
  "fields": [
    { "name": "Custom Text", "value": "Custom text data" },
    { "name": "Custom Boolean", "value": "True" },
    { "name": "Custom Date", "value": "2022-04-22 14:00:00.000" },
    { "name": "Custom List", "value": "One" },
    { "name": "Custom MVL", "value": "One;Three" }
  ]
}

```

Notes about the above example:

- The specified Custodian, Location, Folder, and custom fields must exist on the case before creating the record.
- Custodian and Location are properties that can be used to specify where the record originally came from. Eg the record was collected from Donald Duck and the location was the Deleted items directory on his laptop.
- The Folder property indicates the folder within EDT where the record should be put.

Part 2 – Upload the native file:

UpdateRecordFile:

```

PUT /api/v1/cases/{caseId}/records/{recordId}/native?renditionAction=delete
content-type: multipart/form-data

[content]

```

Parameter	Description
renditionAction	<p>Optional. Query string parameter. Used to specify what happens to the other renditions of a record. This parameter was added in EDT 11.6.2.</p> <p>There are 2 main renditions of a document. Native (format) and PDF. One or the other or both may exist for a record. In addition, a record may have a MarkupPDF if the record has markup (redactions and/or annotations).</p> <p>The Native file, if it exists, is considered the primary rendition. If the Native file is updated, the PDF and MarkupPDF may be affected. However, if the PDF is updated the Native file will not be touched.</p> <p>The renditionAction can have one of 3 values: "none", "delete", or "regenerate":</p> <ul style="list-style-type: none"> • None. No action is taken on the other renditions of a record. This is the default if renditionAction is not specified. • Delete. Delete relevant renditions. • If updating the Native file, delete the PDF and MarkupPDF renditions if they exist. • If updating the PDF file, delete the MarkupPDF rendition if it exists. • Regenerate. Regenerate relevant renditions. • If updating the Native file, regenerate the PDF and MarkupPDF renditions if they exist. • If updating the PDF file, regenerate the MarkupPDF rendition if it exists.

Note: There is a known issue in 11.6.2 that if no renditionAction is specified, the Web Server's cached version of the record file is not refreshed, so users may continue to see the old version of the file for a period of time.

Notes:

- The route also supports updating 2 types (rendition types) of content: Native, Pdf.

Example in C#

```
public async Task<string> UploadNativeFile(string filePath)
{
    // Example only. HttpClient should not be instantiated every time like this.
    using var client = new HttpClient();

    // Set the base address of the REST API
    client.BaseAddress = new Uri(https://<host>/);
    client.DefaultRequestHeaders.Authorization =
        new AuthenticationHeaderValue("Bearer", "<secret>");

    using var content = new MultipartFormDataContent();
    content.Add(new StreamContent(File.OpenRead(filePath)), "file",
        Path.GetFileName(filePath));

    // Send the PUT request to the REST API endpoint with the content
    var response = await
client.PutAsync("/api/v1/cases/256/records/10/native?renditionAction=delete", content);
    response.EnsureSuccessStatusCode();

    // Return the response content as a string
    return await response.Content.ReadAsStringAsync();
}
```

5.5.2 Updating a Record

Note: To update the file content, use the API described in Part 2 of the previous topic.

To update the metadata of a record, use the PUT method:

```
PUT /api/v1/cases/{caseId}/records/{idOrKey}
content-type: application/json

{
  "title": "NewTitle",
  "body": "New Body keep as null to not update",
  "location": "Record Upload",
  "folder": "Other Documents",
  "custodian": "Andrea Ring",
  "documentId": "ACDC.321",
  "fileExtension": ".pdf",
  "primaryDateUtc": "2022-06-01T07:03:07.089Z",
  "fields": [
    { "name": "Custom Text", "value": "Hello WOrld" }
  ]
}
```

Note:

- If you don't provide a field in request body, it will not be updated. This is somewhat different behaviour than other Update APIs, and may be revised to be more consistent. The exceptions are:
 - Title and FileExtension fields are required,
 - DocumentID and PrimaryDateUtc will be cleared if they are not included.

- If using a Key, use the "Key:" prefix. The prefix can be omitted if the value is not a number. Eg `../Key:API-001` or `../API-001`

5.5.4 Deleting a Record

```
DELETE /api/v1/cases/{caseId}/records/{idOrKey}
```

Extra - Deleting all records in a Case

This will be used by the ISL as part of:

- 5.2.3 Merging an Agency File or a Court File
- 5.2.5 Returning a Case to the Police

```
DELETE /api/v1/cases/{caseId}/records
```

Extra - Deleting all of a record's renditions

Endpoint: DeleteRenditions

```
DELETE /api/v1/cases/{caseId}/records/{idOrKey}/renditions
```

Removes the renditions of a record. For example, the native file, PDF, Markup PDF, etc

Requested by Maple to complete 5.2.5. Added in EDT 11.6.5.

User Accounts

Extra Scenario - Creating a User

The CreateUser endpoint:

```
POST /api/v1/users
content-type: application/json

{
  "key" : "PID-12345",
  "username" : "buzer",
  "email" : "buzer@test.com",
  "fullname" : "Bob Uzer",
  "accountType": "Saml",
  "role" : "User"
}
```

Key	Optional. Must be a unique value for all users. The Key provides an alternative way to specify a User account. It is often used when the user to create a mapping between a User account in EDT and a user identifier from an external system.
AccountType	Valid values are "Saml" or "EDT"
Password	(not shown in the example above), Required if the AccountType is "EDT".

Optional and ignored if the AccountType is "Saml" since the user will log in via the SAML Identity provider and not into EDT directly.

Notes:

- Setting custom properties is not currently implemented.

Supporting User endpoints

GetUsers:

```
GET /api/v1/users
```

GetUser using the user's ID:

```
GET /api/v1/users/1c34184d-b763-4ebe-b86d-efd2a4b01c29
```

GetUser by Key (the "key:" prefix must be specified when using a Key for Users):

```
GET /api/v1/users/key:PID-1234
```

Extra scenario: UpdateUser

Example:

```
PUT /api/v1/users
content-type: application/json

{
  "id": "05f8995d-522c-426a-b752-14312d49f7bc",
  "key": "ABC",
  "email": "quagmire@test.com",
  "password": null,
  "phone": null,
  "accountType": "EDT",
  "role": "User",
  "fullName": "Glen Quagmire",
  "isActive": true
}
```

Id	Must be passed to indicate which user resource to update
Password	Optional and ignored if the AccountType is "Saml" since the user will log in via the SAML Identity provider and not into EDT directly. If the AccountType is "EDT", pass null to leave the password unchanged. Provide a password to change the password.

Extra scenario: Add and remove user to/from OU Group

OU Groups are also referred to as Site Groups. These are groups at the Site/Org Unit level, not at the Case level.

Get a list of Org Unit Groups within an Org Unit:

```
GET /api/v1/org-units/{orgUnitId}/groups
```

Adding a user to an Org Unit Group. The following example shows adding the user using their User Id. Alternatively

the user can be specified using their Key. If using the Key, ensure the "key:" prefix is included.

```
### AddUserToOrgUnitGroup2 - Using User Id.
POST /api/v1/org-units/{orgUnitId}/groups/{ouGroupId}/users
Content-Type: application/json

{
  "userIdOrKey": "1c34184d-b763-4ebe-b86d-efd2a4b01c29"
}
```

Removing a user from an Org Unit Group:

```
DELETE /api/v1/org-units/{orgUnitId}/groups/{ouGroupId}/users/{userId}
```

Or

```
DELETE /api/v1/org-units/{orgUnitId}/groups/{ouGroupId}/users/key:{user key}
```

Examples of removing a user from group 2 in org unit 1:

```
# RemoveUserFromOrgUnitGroup2 - By User Id
DELETE {{host}}/api/v1/org-units/1/groups/2/users/1c34184d-b763-4ebe-b86d-efd2a4b01c29

# RemoveUserFromOrgUnitGroup2 - by User Key
DELETE {{host}}/api/v1/org-units/1/groups/2/users/Key:PID-1234
```

Extra scenario: GetUserOuGroups

(OU Groups are also referred to as Site Groups. These are groups at the Site/Org Unit level, not at the Case level.)

Get the OU Groups that a user is a member of:

```
GET /api/v1/org-units/{orgUnitId}/users/{userIdOrKey}/groups
```

Examples (Authorization not shown. The IDs/Keys shown are not from the POC/PreProd environments):

```
### GetUserOuGroups - using User Key
GET /api/v1/org-units/1/users/key:PID-1234/groups

### GetUserOuGroups - using User ID
GET /api/v1/org-units/1/users/1c34184d-b763-4ebe-b86d-efd2a4b01c29/groups

### GetUserOuGroups - User who is in no OU groups
GET /api/v1/org-units/1/users/32cd4b27-f2ef-411a-812c-55b57436479c/groups

### GetUserOuGroups - User does not exist. Should return 404 (Not found).
GET /api/v1/org-units/1/users/key:xxxxx/groups

### GetUserOuGroups - Specified user is an admin, which should return 400 (Bad Request)
error
GET /api/v1/org-units/1/users/4BC1C746-6D23-4267-91C7-FACFDA03622B/groups
```

Platform Events

New in 11.8

Get Platform Events

Platform Events enable external systems to monitor events occurring within the platform. At present two event types

are available:

- ExportToParticipant
- ExportToParticipantDelete

Event Types added in 11.9

- LoadFileImportUI
- LoadFileImportAPI. To be used by the PDEMS project.

Request

Endpoint Name: GetPlatformEvents

```
GET api/v1/platform-events
```

The PlatformEventsRequest model supports the following query string parameters:

- **Page**. Optional. Default: 1.
- **PageSize**. Optional. Default: 50. Max value: 1000.
- **Filter**. Supported filter fields:
 - **Id**. The ID of the PlatformEvent.
 - **CreatedUtc**. ISO-8601 format. e.g. 2024-03-14T13:23:45.678Z. 24-hour time. Must include the "Z" at the end to indicate it is UTC. Must contain a day and time down to seconds. Milliseconds are optional.
 - **EventType**. Multiple event types can be specified using a semi-colon separator.

Examples:

```
// Get events where CreatedUtc >= 5 May 2024 12:34:56 AND Id > 222.
GET api/v1/platform-events?filter=CreatedUtc:>=2024-05-05T12:34:56,Id:>222

// Return events where the ID is greater than 123 which have an EventType of
// ExportToParticipant OR ExportToParticipantDeleted
GET api/v1/platform-
events?filter=Id:>123,EventType:ExportToParticipant;ExportToParticipantDelete

// Get the event with an ID of 222
GET api/v1/platform-events?filter=Id:222
```

Response

Returns a paged list of **PlatformEvent** objects.

The sort order is always **Id** ascending.

Examples of the 2 event types (these examples are indicative only; the real response models may differ):

```
{
  "id": 1447,
  "createdUtc": "2024-11-15T06:41:28.963",
  "createdBy": "3946f5d8-b15f-48a9-812f-ec617df26b2d",
  "eventType": "ExportToParticipant",
  "caseId": 85412,
  "fields": [
    { "name": "ExportToParticipantId", "value": "60" },
    { "name": "ExportToParticipantCreatedUtc", "value": "2024-11-27T04:52:09.020000Z" },
    { "name": "ToEdtInstanceId", "value": "1" },
    { "name": "ToEdtInstance", "value": "Local" },
    { "name": "ToCaseId", "value": "85665" },
    { "name": "ToCase", "value": "XUAN Dang (000354)" },
    { "name": "PersonId", "value": "1408" },
    { "name": "PersonKey", "value": "DXV11" },
    { "name": "ParticipantType", "value": ["Accused Person"] },
    { "name": "Status", "value": "Completed" },
    { "name": "ExportName", "value": "Exported dev.admin 2024_11_15_13_33_49" },
    { "name": "ExportId", "value": "1092" },
    { "name": "CaseId", "value": "85412" }
  ],
  "organisationalUnitId": 1
},
{
  "id": 1448,
  "createdUtc": "2024-11-15T06:41:28.963",
  "createdBy": "3946f5d8-b15f-48a9-812f-ec617df26b2d",
  "eventType": "ExportToParticipantDelete",
  "caseId": 85412,
  "fields": [
    { "name": "ExportToParticipantId", "value": "60" },
    { "name": "ExportToParticipantDeletedUtc", "value": "2024-11-27T14:27:55.433000Z" },
    { "name": "ToEdtInstanceId", "value": "1" },
    { "name": "ToEdtInstance", "value": "Local" },
    { "name": "ToCaseId", "value": "85665" },
    { "name": "ToCase", "value": "XUAN Dang (000354)" },
    { "name": "PersonId", "value": "1408" },
    { "name": "PersonKey", "value": "DXV11" },
    { "name": "ParticipantType", "value": ["Accused Person"] },
    { "name": "Status", "value": "Completed" },
    { "name": "ExportName", "value": "Exported dev.admin 2024_11_15_13_33_49" },
    { "name": "ExportId", "value": "1092" },
    { "name": "CaseId", "value": "85412" }
  ],
  "organisationalUnitId": 1
}
```

Notifications

New in 11.8, Notifications appear in the Notifications tab of the Activity Log view.

Creating a new Notification

Name: CreateNotification


```
POST /api/v1/org-units/{orgUnitId}/notifications
content-type: application/json

{
  "notificationType": "Export To Participant",
  "level": "Information",
  "message*": "Some text to display to the user",
  "url": "{{baseUrl}}/review/71174",
  "caseId": 71174
}
```

Properties	Description
NotificationType	<p>The type of notification. The only supported notification types are currently:</p> <ul style="list-style-type: none"> *Export To Participant "Delete Export To Participant" <p>The registration of new NotificationType values can be requested through support.</p>
Level	One of: "Information", "Warning", "Error".
Message	The message to display
Url	Optional. If provided, the message will be displayed as a hyperlink in Activity Log > Notifications.
Caseld	Optional. If provided, the notification will be associated with the specified Case.

Other

Get the Version

As of 24 Oct 2022, this returns the version of the EDT API, not the EDT (Web) version

```
GET /api/v1/version
```

Appendix – TimeZones in EDT

When the API references TimeZones or TimeZoneIDs, it is referring to a defined set of IDS provided by Microsoft in Windows.

When creating a case, the TimeZone parameter must be a valid Microsoft TimeZone ID. The list of TimeZone IDs is provided below.

The TimeZone ID is used by EDT to look up information about the TimeZone which includes all sorts of details about the specified timezone, including the rules for when DST starts and ends.

The timeZone parameter must be entered exactly as shown in the list, otherwise a 500 series error will occur.

Other ways to get the list of valid TimeZone IDs include:

- The TimeZone column at <https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/default-time-zones?view=windows-10>
- FYI apart from the list below and the Microsoft webpage sent through the other day, another way of seeing the available TimeZones on your machine is to run the system utility TZUTIL from a command line with the /I switch:

```
C:> tzutil /l
```

The 2nd line of each listed timezone is the TimeZone ID.

- The EDT API endpoint (currently not accessible)

TimeZone IDs:

Dateline Standard Time	Cape Verde Standard Time	Central Asia Standard Time
UTC-11	Sao Tome Standard Time	Bangladesh Standard Time
Aleutian Standard Time	Morocco Standard Time	Omsk Standard Time
Hawaiian Standard Time	W. Europe Standard Time	Myanmar Standard Time
Marquesas Standard Time	Central Europe Standard Time	SE Asia Standard Time
Alaskan Standard Time	Romance Standard Time	Altai Standard Time
UTC-09	Central European Standard Time	W. Mongolia Standard Time
Pacific Standard Time (Mexico)	W. Central Africa Standard Time	North Asia Standard Time
UTC-08	Jordan Standard Time	N. Central Asia Standard Time
Pacific Standard Time	GTB Standard Time	Tomsk Standard Time
US Mountain Standard Time	Middle East Standard Time	China Standard Time
Mountain Standard Time (Mexico)	Egypt Standard Time	North Asia East Standard Time
Mountain Standard Time	E. Europe Standard Time	Singapore Standard Time
Yukon Standard Time	Syria Standard Time	W. Australia Standard Time
Central America Standard Time	West Bank Standard Time	Taipei Standard Time
Central Standard Time	South Africa Standard Time	Ulaanbaatar Standard Time
Easter Island Standard Time	FLE Standard Time	Aus Central W. Standard Time
Central Standard Time (Mexico)	Israel Standard Time	Transbaikal Standard Time
Canada Central Standard Time	South Sudan Standard Time	Tokyo Standard Time
SA Pacific Standard Time	Kaliningrad Standard Time	North Korea Standard Time
Eastern Standard Time (Mexico)	Sudan Standard Time	Korea Standard Time
Eastern Standard Time	Libya Standard Time	Yakutsk Standard Time
Haiti Standard Time	Namibia Standard Time	Cen. Australia Standard Time
Cuba Standard Time	Arabic Standard Time	AUS Central Standard Time
US Eastern Standard Time	Turkey Standard Time	E. Australia Standard Time
Turks And Caicos Standard Time	Arab Standard Time	AUS Eastern Standard Time
Paraguay Standard Time	Belarus Standard Time	West Pacific Standard Time
Atlantic Standard Time	Russian Standard Time	Tasmania Standard Time
Venezuela Standard Time	E. Africa Standard Time	Vladivostok Standard Time
Central Brazilian Standard Time	Volgograd Standard Time	Lord Howe Standard Time
SA Western Standard Time	Iran Standard Time	Bougainville Standard Time
Pacific SA Standard Time	Arabian Standard Time	Russia Time Zone 10
Newfoundland Standard Time	Astrakhan Standard Time	Magadan Standard Time
Tocantins Standard Time	Azerbaijan Standard Time	Norfolk Standard Time
E. South America Standard Time	Russia Time Zone 3	Sakhalin Standard Time
SA Eastern Standard Time	Mauritius Standard Time	Central Pacific Standard Time
Argentina Standard Time	Saratov Standard Time	Russia Time Zone 11
Greenland Standard Time	Georgian Standard Time	New Zealand Standard Time
Montevideo Standard Time	Caucasus Standard Time	UTC+12
Magallanes Standard Time	Afghanistan Standard Time	Fiji Standard Time
Saint Pierre Standard Time	West Asia Standard Time	Chatham Islands Standard Time
Bahia Standard Time	Ekaterinburg Standard Time	UTC+13
UTC-02	Pakistan Standard Time	Tonga Standard Time
UTC	Qyzylorda Standard Time	Samoa Standard Time
GMT Standard Time	India Standard Time	Line Islands Standard Time
Greenwich Standard Time	Sri Lanka Standard Time	
Azores Standard Time	Nepal Standard Time	