

# EDT Core API for Maple – WIP Notes

## EDT 11

### Contents

Contents.....	1
Revisions .....	3
EDT 11.6.2.....	9
Changes .....	9
Additions.....	10
Terminology.....	10
Introduction .....	11
Environment Names.....	11
Proof of Concept server .....	11
Testing the API .....	12
Detailed Documentation.....	12
Guide to using the API.....	13
Identifying Resources.....	13
EDT IDs .....	13
Using Keys to identify resources .....	13
Summary of API endpoints.....	15
Endpoint Details.....	16
Cases.....	16
Getting a list of Cases .....	16
Case Custom Fields.....	16
CaseIdLookup - Looking up a Case using a key or a custom field .....	16
CaseIdsLookup.....	17
5.2.1 Creating a Case.....	17
5.2.2 Updating the metadata of a Case.....	18
5.2.3 Merging an Agency File or a Court File – DRAFT.....	19
Case Users .....	19
5.3.1 Adding & Removing Users to a Case .....	19

Extra scenario – Add a user to multiple cases - SyncUserCases .....	20
Extra scenario – GetUserCases – Get a list of the cases a specified user has access to .....	21
Case Groups.....	21
Supporting endpoint: Get the Groups in a Case .....	21
Supporting endpoint: Get a Group’s details .....	21
Updating a group’s users using the SyncGroupUsers endpoint (ISL) .....	21
Updating a group’s users using AddUserToCaseGroup and RemoveUserFromCaseGroup (DIAM) .....	22
Getting the groups of a user .....	22
Participants/Persons.....	22
5.4.1 Adding a Participant .....	23
Supporting endpoint: Get all the active Person resources .....	24
Supporting endpoint: Get the details of a Person resource using an ID and using a Key .....	24
5.4.2 Updating a Participant .....	24
5.4.3 Add a Participant to a Case.....	25
5.4.4 Remove a Participant from a Case.....	27
Supporting endpoint: GetCaseParticipants.....	27
Supporting endpoint: GetParticipantTypes.....	27
5.4.5 Merging Participant Details .....	27
Supporting endpoints – Identifiers.....	28
Records.....	29
Searching for Records.....	29
Using Keys with the Record APIs.....	30
5.5.1 Adding a Record .....	31
5.5.2 Updating a Record.....	32
5.5.4 Deleting a Record .....	33
Extra - Deleting all records in a Case .....	33
User Accounts .....	33
Extra Scenario - Creating a User.....	33
Supporting User endpoints.....	34
Extra scenario: UpdateUser .....	34
Extra scenario: Add and remove user to/from OU Group .....	35
Extra scenario: GetUserOuGroups.....	35
Other .....	36
Get the Version .....	36
Appendix – TimeZones in EDT .....	36

## 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"> <li>POST /api/v1/cases <b>[Update:</b> this route is no longer valid. See <i>5.2.1 Creating a Case</i>]</li> </ul>
13 May 11.3.365	<p>Exposed API routes:</p> <ul style="list-style-type: none"> <li>POST /api/v1/cases/{caseId}/case-users/sync</li> <li>POST /api/v1/cases/{caseId}/groups/{groupId}/sync</li> <li>GET /api/v1/version</li> <li>GET /api/v1/cases/{caseId}/groups</li> <li>GET /api/v1/cases/{caseId}/groups/{groupId}</li> </ul> <p>The Swagger endpoints have changed to:</p> <ul style="list-style-type: none"> <li>auth: <a href="https://346n1v5oeh.execute-api.ca-central-1.amazonaws.com/swagger/index.html">https://346n1v5oeh.execute-api.ca-central-1.amazonaws.com/swagger/index.html</a></li> <li>open: <a href="https://346n1v5oeh.execute-api.ca-central-1.amazonaws.com/swagger-ui/index.html">https://346n1v5oeh.execute-api.ca-central-1.amazonaws.com/swagger-ui/index.html</a></li> </ul> <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"> <li>Properties removed: Id, IsS3, IsThumbnailsEnabled, ExportTempPath, StagingAreaPath</li> <li>Properties changed: <ul style="list-style-type: none"> <li>int TemplateId is now string TemplateCase. This now accepts either a Case Name or a Case Id</li> <li>TimeZone is now TimeZonId</li> <li>Suffix is now DatabaseFileSuffix</li> </ul> </li> </ul> <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 which 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 <i>5.2.2 Updating the metadata of a Case</i> scenario:</p> <p>Exposed API routes:</p> <ul style="list-style-type: none"> <li>GET /api/v1/cases/{caseId}</li> <li>PUT /api/v1/cases/{caseId}</li> </ul> <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 [<b>Update</b>: these Person routes have been changed to include OrgUnit]</p> <ul style="list-style-type: none"> <li>• POST /api/v1/persons</li> <li>• GET /api/v1/persons</li> <li>• GET /api/v1/persons/{personId}</li> </ul> <p>Scenario: 5.5.1 Adding a Record</p> <ul style="list-style-type: none"> <li>• POST /api/v1/cases/{caseId}/records</li> <li>• PUT /api/v1/cases/{caseId}/records/{docId}/{fileType}</li> </ul>
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 [<b>Update</b>: this route has changed in include OrgUnit]</p> <ul style="list-style-type: none"> <li>• POST /api/v1/persons. Updated to support Title and Custom Fields</li> </ul> <p>Scenario: 5.4.2 Updating a Participant [<b>Update</b>: this route has changed in include OrgUnit]</p> <ul style="list-style-type: none"> <li>• PUT /api/v1/persons</li> </ul>
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"> <li>• GetByIdByKey /api/v1/cases/{key}/id</li> </ul> <p>Scenario: 5.5.2 Updating a Record</p> <ul style="list-style-type: none"> <li>• PUT /api/v1/cases/{caseId}/records</li> </ul> <p>Scenario: 5.5.4 Deleting a Record</p> <ul style="list-style-type: none"> <li>• DELETE /api/v1/cases/{caseId}/records/{idOrKey}</li> </ul> <p>Added Users endpoints (additional to the scenarios in the DEMS specification):</p> <ul style="list-style-type: none"> <li>• CreateUser: POST /api/v1/users</li> <li>• GetUsers: GET /api/v1/users</li> <li>• GetUser: GET /api/v1/users/{userIdOrKey}</li> </ul>
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: [<b>Update</b>: these routes have changed in include OrgUnit]</p> <ul style="list-style-type: none"> <li>• GET /api/v1/persons/{idOrKey}</li> <li>• PUT /api/v1/persons/{idOrKey}</li> </ul>

Date	Exposed endpoints and other changes
4 Aug 11.5.558	<p><b>OrgUnits</b> 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 &amp; Removing Users to a Case</i> updated:</p> <ul style="list-style-type: none"> <li>The CaseUsersSync endpoint now uses arrays of Ids and/or Keys.</li> </ul> <p>Records.</p> <ul style="list-style-type: none"> <li>The way records are identified has changed. See <i>Using Keys with the Record APIs</i> for details.</li> </ul> <p>The CreateCase and UpdateCase endpoints have been updated in the following ways:</p> <ul style="list-style-type: none"> <li>The Key field has been added so the Key value can be set/changed.</li> <li>The way that custom fields are specified has changed.</li> </ul>
19 Aug 11.5.633	<p><b>Case ID Lookup.</b> 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><b>DeleteRecords</b> 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><b>SyncUserCases</b> endpoint added. (Sometimes referred to as 5.3.1c:)</p> <pre> POST /api/v1/org-units/1/users/{userIdOrKey}/cases/sync </pre> <p><b>Persons</b> 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><b>Corrections:</b></p> <ul style="list-style-type: none"> <li>Removed the notes about "Document+ID" from the RecordUpdate and RecordDelete APIs.</li> <li>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 "timeZoneld".</li> </ul>

Date	Exposed endpoints and other changes
23 Sep 11.5.887	<ul style="list-style-type: none"> <li>CaseldLookup <ul style="list-style-type: none"> <li>Added support for "Key:" prefix.</li> <li>If specifying a Key, the endpoint now ensures the returned Case is in the specified OrgUnit.</li> </ul> </li> <li>Implemented the GetCaseParticipants API. Example: <pre>GET /api/v1/cases/{caseId}/participants</pre> <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> </li> <li>Implemented the GetParticipantTypes API: <pre>GET /api/v1/org-units/{orgUnitId}/participant-types</pre> </li> <li>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.]</li> <li>Implemented the AddCaseParticipant endpoint as an alternative to SyncCaseParticipants. See the notes for <i>"5.4.3 Add a Participant to a Case"</i></li> <li>Draft notes for <i>5.2.3 Merging an Agency File or a Court File</i></li> </ul>
29 Sep 11.5.912	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul>
7 Oct 11.5.987	<ul style="list-style-type: none"> <li>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.</li> <li>Ability to add and remove a user to/from an Org Unit (site) group.</li> <li>Person resources. Fixed: CreatePerson does not set custom fields. (Only Text and DateTime custom fields are currently supported for Person resources)</li> </ul>
24 Oct 11.5.1051	<ul style="list-style-type: none"> <li>Duplicate case names allowed in CreateCase and UpdateCase</li> <li>UpdateUser endpoint added.</li> <li>GetOrgUnitFields added: <pre>GET /api/v1/org-units/{orgUnitId}/fields</pre> <p>This replaces the previous route which was:</p> <pre>GET /api/v1/site/fields?organisationalUnitId=1 // DO NOT USE!</pre> </li> <li>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.</li> </ul>

Date	Exposed endpoints and other changes
16 Nov 11.5.1255 API: 11.5.234	<ul style="list-style-type: none"> <li>Added the ability to use custom field names in the CaseldLookup endpoint</li> <li><b>Breaking change:</b> Changed the result of the CaseldLookup endpoint to return a list of case ID/Key values.</li> <li>Added a new GetUserCases endpoint to get a list of the cases a specified user is assigned to.</li> <li>Added the ability to find records using a multiple field values. See the <i>Records &gt; Searching for Records</i> section for more information.</li> <li>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.</li> </ul>
1 Dec 11.5.1375 API: 11.5.309	<ul style="list-style-type: none"> <li>Added GetUserOuGroups endpoint to return a list of the OU (site) Groups that a user belongs to. See <i>Extra scenario: GetUserOuGroups</i>.</li> </ul>
<b>2023</b>	
9 Feb 11.5.1762/ 11.5.470	<ul style="list-style-type: none"> <li>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</li> </ul>
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"> <li>Terminology section added</li> <li>Introduction updated</li> <li>Summary of API endpoints section added</li> </ul> <p>New endpoints:</p> <ul style="list-style-type: none"> <li>AddUserToCaseGroup (DIAM, NED-232)</li> </ul> <pre>POST /api/v1/cases/{caseId}/groups/{groupId}/users/{userId}</pre> <ul style="list-style-type: none"> <li>RemoveUserFromCaseGroup (DIAM, NED-232)</li> </ul> <pre>DELETE /api/v1/cases/{caseId}/groups/{groupId}/users/{userId}</pre> <p><b>Note:</b> 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}
```



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

```
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.

## Terminology

Term	Definition
Published	Published endpoints are those that can be used by 3 <sup>rd</sup> party clients. Published endpoints are supported by EDT. 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.

All endpoints in this document are published. All other endpoints in the EDT Core API are not published and not to be used.

## 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 which API routes 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 which 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 exposed<sup>1</sup> 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

<sup>1</sup> 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.

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:

```
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  
Whether to wait for the new case to be created or return immediately. Default: True.
- name string  
nullable: true  
The name of the Case. Must be unique.
- description string  
nullable: true

# 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 which 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 which 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 which 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.

GetVersion	GET /api/v1/version
GetCases	GET /api/v1/org-units/{orgUnitId}/cases
GetCustomFields	GET /api/v1/org-units/{orgUnitId}/fields
CaseIdLookup	GET /api/v1/org-units/{orgUnitId}/cases/{keyOrField}/id
CreateCase	POST /api/v1/org-units/{orgUnitId}/cases
GetCaseDetail	GET /api/v1/cases/{caseId}
SaveCaseSettings	PUT /api/v1/cases/{caseId}
ExportToCase	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
GetPerson	GET /api/v1/org-units/{orgUnitId}/persons/{personIdOrKey}
UpdatePerson	PUT /api/v1/org-units/{orgUnitId}/persons/{personIdOrKey}
AddCaseParticipant	POST /api/v1/cases/{caseId}/participants
<del>RemoveCaseParticipant</del>	<del>POST Use SyncCaseParticipants [See Added in 11.6.2 section below]</del>
<i>Added May 2023</i>	
SyncCaseParticipants	POST /api/v1/cases/{caseId}/participants/sync [NOW DEPRECATED]
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
GetOuGroups	GET /api/v1/org-units/{orgUnitId}/groups
AddUserToOuGroup	POST /api/v1/org-units/{orgUnitId}/groups/{ouGroupId}/users
RemoveUserFromOuGroup	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

Added in 11.6.2

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}

```

## 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.

#### CaseIdLookup - Looking up a Case using a 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 which 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 values. So, for example, if a single Case is found, the response body is



something like:

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

8 Nov 2022: Previously this returned a single case id. This is a breaking change. Any existing code that uses this endpoint must be updated to handle the changed behaviour. Typically, we would not allow breaking changes and would handle a change like this by issuing a new version of the endpoint. However, Maple requested this change and is currently the only client of this endpoint, and since the API has not officially been published yet, we believe it is reasonable to have a breaking change at this time.

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" which 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
```

## CaseIdsLookup

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"]
}
```

### 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) <sup>2</sup>.

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 a GET:

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

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

---

<sup>2</sup> At present, the way custom property values are specified is undergoing change. This is the reason why specifying custom properties for Cases differs from the way it is done for Persons.

```
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

#### *ExportToCase*

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"
}
```

## 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 which should be assigned to the Case. Any existing users assigned to the case which are not included in the request body will be removed from the Case. Any users identified in the request body which 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	<p>Optional. The groups within each case that the user should be added to.</p> <p>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.</p> <p>If a specified group is not found within a case, it is ignored.</p>

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.

### 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.

### 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 which are used to specify who can access the system. Persons are business objects which 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 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"
}
```

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	Optional. If provided, it must be unique within the site.
Email	If the Address.Email value is not null, the value must be unique within the Org Unit.

### 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" which 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" which 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: Get all the active Person resources

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

### 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 Participant

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 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.

### 5.4.3 Add a 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 [Deprecated – May be removed]

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*. This endpoint has been deprecated.

#### 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 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 every 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	<p>Filter on Identifier fields. The available fields are: "ItemType", "ItemId", "IdentifierType", "IdentifierValue".</p> <p>Separate the fields to search by with a comma. e.g.</p> <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	A filter to apply to the records in the case. The syntax matches that of the Search Bar in the UI. E.g.

```
DocKind:Email AND EDTID:>=200000 AND DocumentID:NOT []
```

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.

Field names: fields in the filter are specified using their display names.

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> <pre>sort=title,edtid sort=title asc,edtid asc</pre> <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>
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 which 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 2<sup>nd</sup> 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	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.

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).

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.

The renditionAction can have one of 3 values: "none", "delete", or "regenerate":

- 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 exception is the Title and FileExtension fields, which are required.
- 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
```

## 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.

```
### AddUserToOuGroup - 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:

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

# RemoveUserFromOuGroup - 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
```

## 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 /l 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	