EDT Core API for Maple – WIP Notes

EDT 11

Contents

Revisions

May 2022 – Initial draft of document

Updates below:

API changes

2022	Exposed endpoints and other changes
28 Apr	Exposed API routes: • POST /api/v1/cases
13 May 11.3.365	Exposed API routes: • 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} The Swagger endpoints have changed to: • 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 The start of other routes has changed from "/v1" to "/api/v1" to be consistent with the documentation.

20 May
11.3.403
	Model changed for the CreateCase API:
	• 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
	
	Documentation below updated to reflect the changes above and add additional information.
	
	The implementation of Case creation using a template has been completed.
	
	The *Add Users to Case* 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)
	
	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.
	
	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.

30 May 11.3.469	This build exposes the primary API routes used for the *5.2.2 Updating the metadata of a Case* scenario: Exposed API routes: • GET /api/v1/cases/{caseId} • PUT /api/v1/cases/{caseId} 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 *Detailed Documentation* contains some further information.

2022	Exposed endpoints and other changes
8 Jun 11.5.54	Scenario: *5.4.1 Adding a Participant* • POST /api/v1/persons • GET /api/v1/persons • GET /api/v1/persons/{personId} Scenario: *5.5.1 Adding a Record* • POST /api/v1/cases/{caseId}/records • PUT /api/v1/cases/{caseId}/records/{docId}/{fileType}
23 Jun 11.5.238	Added some notes about some functionality under consideration. See "[Proposed] Using Keys to as an alternative to EDT IDs" Scenario: *5.4.1 Adding a Participant* • POST /api/v1/persons. Updated to support Title and Custom Fields Scenario: *5.4.2 Updating a Participant* • PUT /api/v1/persons
1 Jul 11.5.307	The section describing the proposed introduction of Keys has been revised with a new approach. The following endpoint can be used to lookup a Case ID using a Key: • GetIdByKey /api/v1/cases/{key}/id Scenario: *5.5.2 Updating a Record* • PUT /api/v1/cases/{caseId}/records Scenario: *5.5.4 Deleting a Record* • DELETE /api/v1/cases/{caseId}/records/{idOrKey} Added Users endpoints (additional to the scenarios in the DEMS specification): • CreateUser: POST /api/v1/users • GetUsers: GET /api/v1/users • GetUser: GET /api/v1/users/{userIdOrKey}
8 Jul 11.5.342	The proposed introduction of Keys has moved out of the proposal stage. Keys will be introduced to routes and resources in the coming weeks. Participant (Person) APIs updated to support specifying an ID or Key: • GET /api/v1/persons/{idOrKey} • PUT /api/v1/persons/{idOrKey}

2022	Exposed endpoints and other changes

4 Aug
11.5558
	OrgUnits We have started to make the APIs *OrgUnit* aware. EDT is currently introducing OrgUnits, whose primarily 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.
	
	The following routes have been updated to require/support specifying the OrgUnit.
	
	`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}`
	
	Case Endpoints: change to the routes for GetCases and CreateCase to include specifying of the OrgUnit:
	
	`CreateCase POST /api/v1/org-units/{orgUnitId}/cases`
	`GetCases GET /api/v1/org-units/{orgUnitId}/cases`
	
	Scenario *5.3.1 Adding & Removing Users to a Case* updated:
	• The CaseUsersSync endpoint now uses arrays of Ids and/or Keys.
	
	Records.
	• The way records are identified has changed. See *Using Keys with the Record APIs* for details.
	
	The CreateCase and UpdateCase endpoints have been updated in the following ways:
	• The Key field has been added so the Key value can be set/changed.
	• The way that custom fields are specified has changed.

Introduction

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.

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[1] 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 provided in the Authorization header of requests like this:

```
Authorization: Bearer 7N6EYCR6TD88O7MV1VXQNBP0K
```

Related to this, most of the routes listed by our Swagger page are not currently callable and will return a 404 result.

[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.

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:

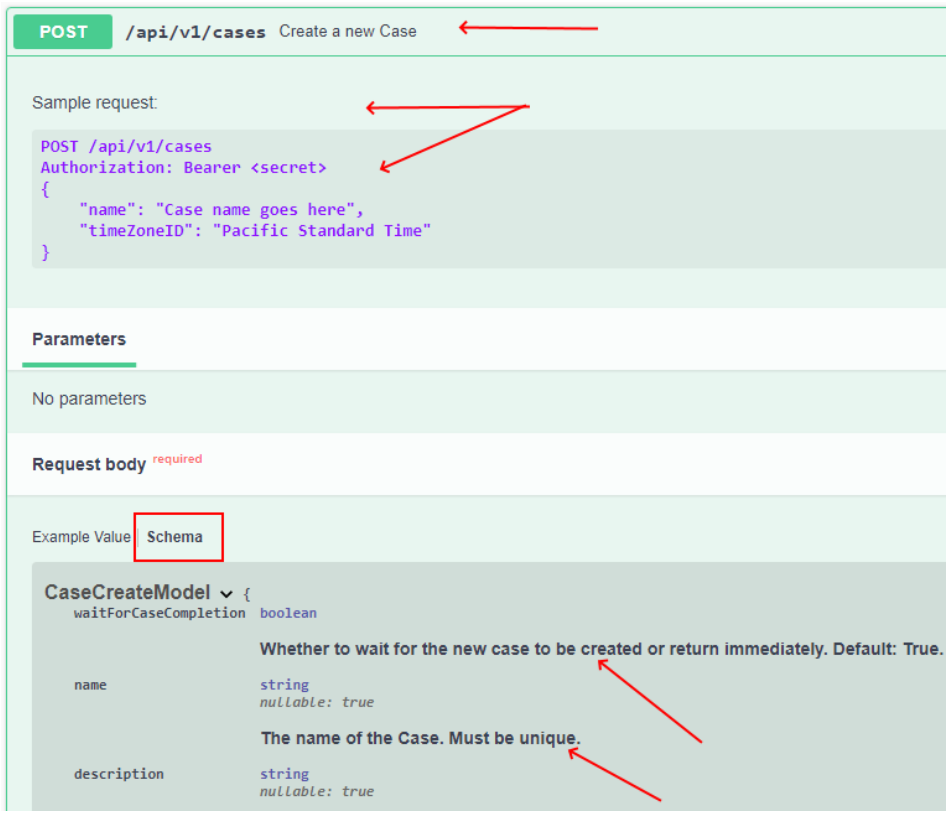
```
GET /api/v1/version
```

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.



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

Implementation for the following is underway.

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.

Available API endpoints

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
```

Looking up Case ID using a Key

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:

```
GET /api/v1/cases/{key}/id
```

If a Case with the specified Key is found, the response body is like:

```
{
  "id": 3
}
```

If the case is not found, the request returns 404 (Not Found).

This endpoint will be changed to in the next build to the following:

Looking up a Case using the key or a custom field

Coming soon.

```
GET /api/v1/cases/{keyOrField}/id
```

To search by Key, use "abc" or "key:abc" or "123" as the keyOrField value.

To search by custom field, specify the field id and value. E.g. to find the Case where field 123 has the value blue, use: "123:blue" as the keyOrField value. If more than one case is found with the specified field value, an error (400) is raised.

Replace any spaces in the key or field value with a plus (+).

5.2.1 Creating a Case

Example:

```
POST /api/v1/org-units/1/cases
Authorization: Bearer 7N6EYCR6TD88O7MV1VXQNBP0K

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

For cases, the custom properties are set using the field ID and value[2].

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

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

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:

[2] 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",
  "useDocIdForDownloads": true,
  "timeZone": "E. Australia Standard Time",
  "removeGeneratedFiles": false,
  "searchHistoryRetention": 10,
  "fields": [
    {
      "id": "23", "value": "On hold"
    }
  ],
  "dateFormat": "d MMM yyyy",
  "showMilliseconds": true
}
```

Case Users

5.3.1 Adding & Removing Users to a Case

Update the users assigned to a Case using the CaseUsersSync 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
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	An array of string values. The Ids of the Users to be synced. Note: Almost all resources in EDT have numeric IDs. User accounts are one exception that uses string IDs.
UserKeys	An array of string values. The keys of the Users to be synced.

Example:

To remove all user from the Case:

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

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

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 GroupsMembersSync endpoint

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

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

{
  "name": "D Duck",
  "firstName": "Donald",
  "lastName": "Duck",
  "address": {
    "email": "dduck@test.com"
  },
  "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" }
  ],
  "status" : "Active"
}
```

Note: “settings” is no longer supported. This has been replaced by “fields”.

Fields	An array of custom fields. Any fields specified must already exist. Each field has a 'name' and 'value'. 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.
Title	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.
Status	Optional. Valid values are "Active" and "Inactive". The default is "Active".

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": "3", "value": "Red" }
  ],
  "displayName": "K Simpson",
  "organisationTypeId": 1,
  "permissions": null,
  "status": "Active"
}
```

Notes:

- You cannot change details of a specific organisation using this endpoint, however you can change which organisations the Person belongs to.

5.4.3 Add a Participant to a Case

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": [ ]
}
```

5.4.4 Remove a Participant from a Case

Use the SyncCaseParticipants endpoint as described above in *5.4.3 Add a Participant to a Case*.

Records

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 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:

```
PUT /api/v1/cases/{caseId}/records/{recordId}/Native
content-type: multipart/form-data

[bytes]
```

Notes:

- The route will eventually support 3 types of content: Native, Pdf, and Text. Only “Native”, as shown in the example above, is currently supported.

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.
- If using a Key, currently the key must be specified using "Document+ID". Eg `../Document+ID:API-001`

5.5.4 Deleting a Record

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

Note:

- If using a Key, currently the key must be specified using "Document+ID". Eg `../Document+ID:API-001`

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
```

Other

Get the EDT 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	