empower

External API reference

CUSTOMER INTEGRATION GUIDE

This document describes the Empowering HR (Empower) API endpoints that are available to external systems. It covers how to connect and authenticate, the conventions shared by every endpoint, and each endpoint in turn with example request and response payloads.

Document	Customer technical reference
Environments	UAT and Production
Authentication	OAuth 2.0 client credentials
Version	1.0
Issued	November 2026

Contents

1 Environments ... 3

2 Getting access ... 3

2.1 Get the ClientAdmin role ... 3

2.2 Generate API credentials ... 3

3 Authentication ... 6

3.1 Request an access token ... 6

3.2 Use the token ... 7

3.3 Token lifecycle ... 8

3.4 Common authentication errors ... 8

4 Conventions ... 10

5 Errors ... 10

6 Endpoint summary ... 12

7 Absence integration walkthrough ... 12

8 Absences ... 14

8.1 List absence types ... 14

8.2 List absence reasons ... 15

8.3 Create an absence ... 16

8.4 List absences ... 18

8.5 Get an absence ... 20

8.6 Update an absence ... 21

8.7 Delete an absence ... 23

9 Cases ... 25

9.1 List case topics ... 25

9.2 Create a case ... 25

9.3 List an employee's cases ... 27

10 Case notes and attachments ... 30

10.1 List case notes ... 30

10.2 Get a case note ... 31

10.3 Create a case note ... 33

10.4 Download a case attachment ... 35

10.5 Download a case note document ... 36

11 Case escalations ... 38

11.1 List case escalations ... 38

11.2 Get a case escalation ... 39

12 Using the Postman collection ... 41

1 Environments

Environment	API base URL	Authorisation server
UAT	`https://api.uat.empoweringhr.com/api`	`https://auth.uat.empoweringhr.com`
Production	`https://api.empoweringhr.com/api`	`https://auth.empoweringhr.com`

All paths in this document are relative to the API base URL. All requests must use HTTPS. Use UAT for development and testing and move to Production when you are ready to go live. Credentials are issued per environment.

2 Getting access

This section reproduces the Empowering HR API getting started guide, which is also published at https://blue-field-038713a03.7.azurestaticapps.net/. Getting connected takes three steps: get the ClientAdmin role, generate API credentials, then authenticate.

2.1 Get the ClientAdmin role

Before you can generate API credentials, you need the **ClientAdmin** role assigned to your account. This role grants access to the API Credentials section in the admin portal.

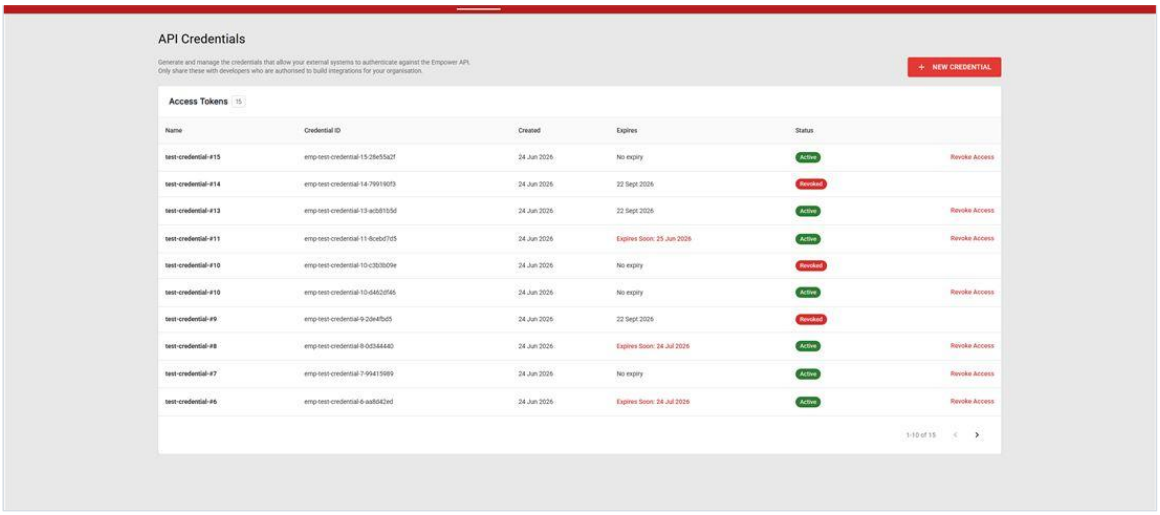
1. **Submit a ticket.** Go to the AdviserPlus support portal and raise a request for the ClientAdmin role to be assigned to your account.
2. **Wait for confirmation.** Once the role has been assigned, you will be able to see the API Credentials tab in the admin portal.
3. **Generate credentials.** Follow the steps in the next section to create your first API credential.

2.2 Generate API credentials

API credentials are managed through the admin portal. Each credential is a client ID and secret pair that your systems use to authenticate against the Empowering HR API. Credentials are issued per environment.

View existing credentials

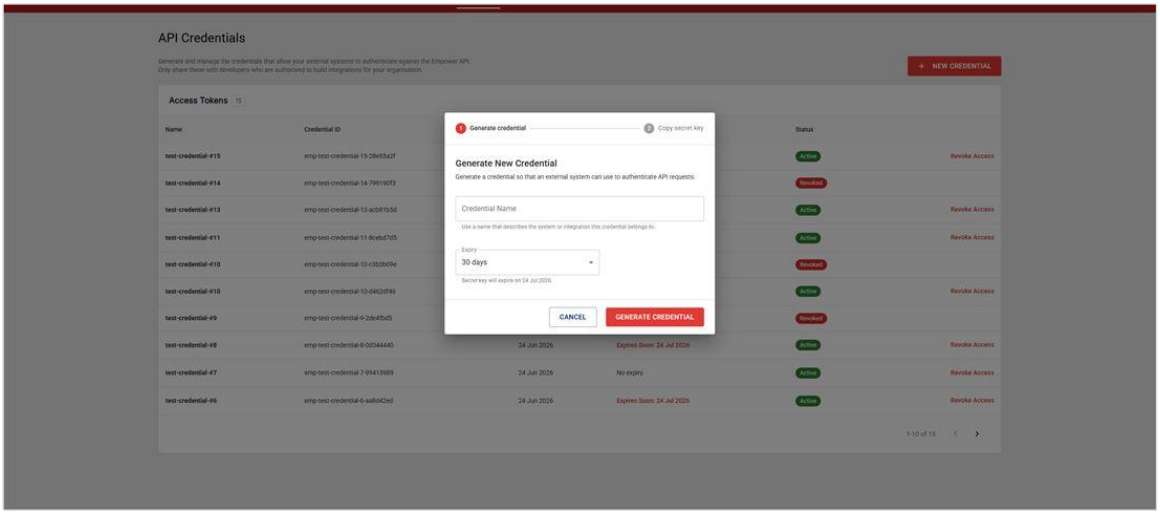
Navigate to the admin portal and select the **API Credentials** tab. You will see all credentials for your organisation with their name, client ID, creation date, expiry and status. Active credentials show a green badge; revoked ones show red.



The API Credentials tab listing the credentials for an organisation

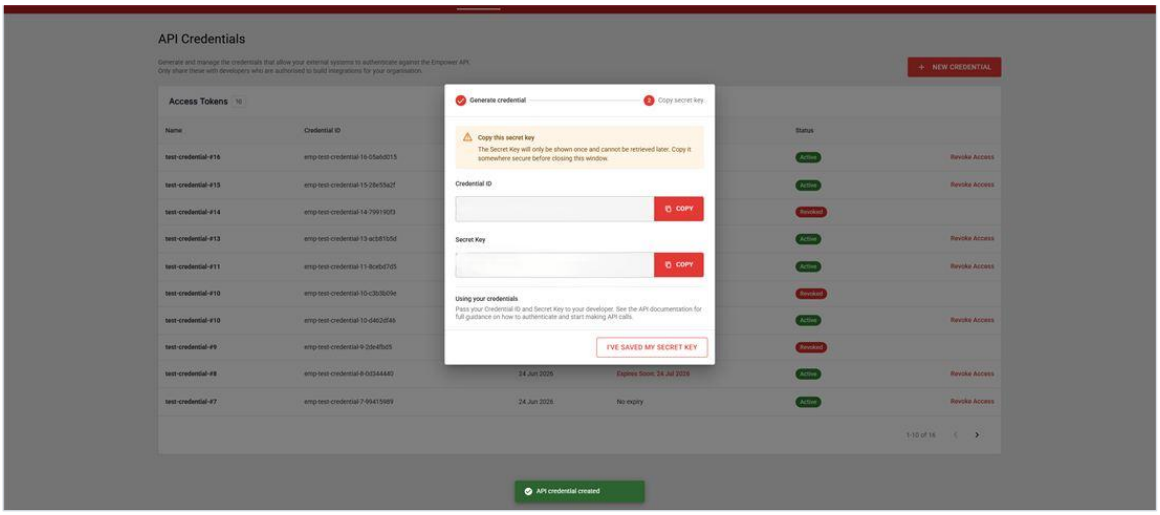
Create a new credential

1. **Open the creation dialog.** Click the **+ New Credential** button in the top right corner.
2. **Enter a name and expiry.** Give the credential a descriptive name that identifies the system or integration it belongs to, for example `Payroll Integration` or `HR Dashboard`. Choose an expiry period from the dropdown, or leave it without an expiry if the integration should run indefinitely. The expiry controls when the secret key will stop working.



The new credential dialog with name and expiry fields

3. **Save the secret key immediately.** After clicking **Generate Credential**, you will see your new **Credential ID** and **Secret Key**. Copy both values using the copy buttons and store them securely.



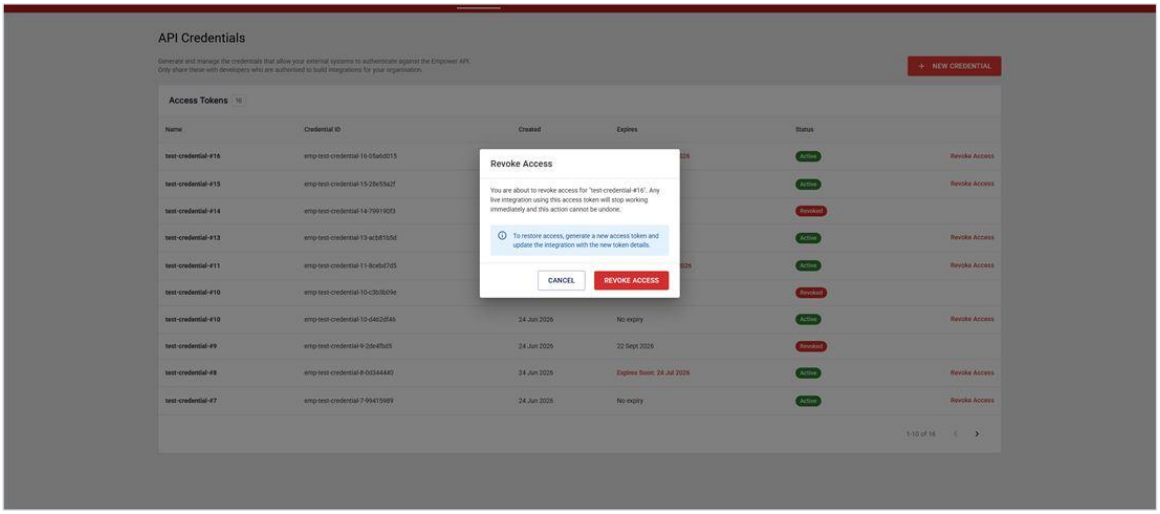
The generated Credential ID and Secret Key

> **The secret key is shown only once.** It cannot be retrieved after you close this dialog. If you lose it, you will need to revoke the credential and create a new one.

4. Click **I've saved my secret key** when you have stored the values securely. These are the `client_id` and `client_secret` used in the authentication step below. Pass them to the developer configuring the integration through a secure channel.

Revoke a credential

If a credential is compromised or no longer needed, revoke it from the credentials list by clicking **Revoke Access** next to the credential.



Revoking a credential from the credentials list

Revoking a credential takes effect immediately. Any integration using the credential will stop working until you generate a new one and update the integration's configuration.

> **Revoking is permanent.** To restore access, generate a new credential and update your integration with the new client ID and secret.

3 Authentication

The API uses **OAuth 2.0 Client Credentials** to authenticate machine-to-machine requests. You exchange your client ID and secret for a short-lived access token, then pass that token in every API request.

> **Keep your secret safe.** Your client secret is shown once when the credential is created and cannot be retrieved again. Store it in a secrets manager or vault. Never commit it to source control or log it.

1. **Request a token.** Send your credentials to the token endpoint. You will receive a JWT access token.
2. **Call the API.** Include the token in the `Authorization` header of every request.
3. **Refresh when expired.** Tokens expire after 1 hour. Request a new one before it expires or when you receive a `401` response.

3.1 Request an access token

Make a `POST` request to the authorisation server's token endpoint with your credentials. The examples use Production; for UAT use `https://auth.uat.empoweringhr.com`.

cURL:

```
curl -X POST https://auth.empoweringhr.com/connect/token \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d
"grant_type=client_credentials&client_id=YOUR_CLIENT_ID&client_secret=YOUR_CLIENT_SECRET&scope=proc
essengineapiaccess"
```

C#:

```
using var httpClient = new HttpClient();

var response = await httpClient.PostAsync(
    "https://auth.empoweringhr.com/connect/token",
    new FormUrlEncodedContent(new Dictionary<string, string>
    {
        ["grant_type"]    = "client_credentials",
        ["client_id"]     = "YOUR_CLIENT_ID",
        ["client_secret"] = "YOUR_CLIENT_SECRET",
        ["scope"]         = "processengineapiaccess"
    }));

var json = await response.Content.ReadFromJsonAsync<JsonElement>();
var accessToken = json.GetProperty("access_token").GetString();
```

Python:

```
import requests

response = requests.post(
    "https://auth.empoweringhr.com/connect/token",
    data={
        "grant_type":    "client_credentials",
        "client_id":     "YOUR_CLIENT_ID",
        "client_secret": "YOUR_CLIENT_SECRET",
        "scope":         "processengineapiaccess",
    }
)
access_token = response.json()["access_token"]
```

Parameters (sent as `application/x-www-form-urlencoded`)

Name	Type	Required	Description
`grant_type`	string	Yes	Always client_credentials.
`client_id`	string	Yes	The client ID returned when the credential was created.
`client_secret`	string	Yes	The client secret shown once at credential creation.
`scope`	string	Yes	Always processengineapiaccess.

Response - `200 OK`

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "processengineapiaccess"
}
```

Field	Description
`access_token`	A signed JWT. Include this in API requests.
`token_type`	Always Bearer.
`expires_in`	Token lifetime in seconds. Typically 3600 (1 hour).
`scope`	The granted scope.

3.2 Use the token

Pass the access token in the `Authorization` header on every API call.

```
GET /api/case-topics HTTP/1.1
Host: api.empoweringhr.com
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...
```

> **Cache your token.** Request a new token only when the current one is expired or about to expire. A single token can be reused for all requests within its lifetime.

3.3 Token lifecycle

Access tokens are short-lived by design. Handle them correctly to avoid unnecessary token requests and failed API calls.

Property	Value
`Token type`	JWT (JSON Web Token), signed with RS256
`Lifetime`	3600 seconds (1 hour)
`Clock skew tolerance`	5 minutes
`Refresh tokens`	Not issued. Request a new access token when the current one expires.

Cache the token in memory and track the expiry time. Request a new token when the cached one is expired or within a short buffer of expiring:

```
public class TokenProvider
{
    private string? _token;
    private DateTime _expiresAt = DateTime.MinValue;
    private static readonly TimeSpan Buffer = TimeSpan.FromMinutes(2);

    public async Task<string> GetTokenAsync()
    {
        if (_token is not null && DateTime.UtcNow < _expiresAt - Buffer)
            return _token;

        // Request a new token (see Request an access token)
        var response = await RequestTokenAsync();
        _token = response.AccessToken;
        _expiresAt = DateTime.UtcNow.AddSeconds(response.ExpiresIn);
        return _token;
    }
}
```

> **Do not decode the token to check claims.** The token's internal structure is an implementation detail. Use the `expires_in` value from the token response to manage expiry. Token claims and format may change without notice.

3.4 Common authentication errors

These are the most frequent errors during integration setup.

Missing token - `401 Unauthorized`. The `Authorization` header is absent. Add `Authorization: Bearer YOUR_TOKEN` to the request.

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.3.1",
  "title": "Unauthorized",
  "status": 401,
  "detail": "Missing authorization token"
}
```

Expired token - `401 Unauthorized`. The token has passed its expiry time. Request a new token from the token endpoint.

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.3.1",
  "title": "Unauthorized",
  "status": 401,
  "detail": "Token has expired"
}
```

Wrong role - `403 Forbidden`. The token is valid but the client does not have the required role. This usually means the credential was not provisioned correctly. Contact your Empowering HR administrator.

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.3",
  "title": "Forbidden",
  "status": 403,
  "detail": "Insufficient permissions. Required one of the following roles: ExternalMachineClient"
}
```

Invalid credentials at the token endpoint - `400 Bad Request`. If the token request itself fails, the authorisation server returns a standard OAuth 2.0 error. The client ID or secret is wrong; verify you are using the correct credentials for this environment.

```
{
  "error": "invalid_client"
}
```

4 Conventions

Format. Request and response bodies are JSON (`application/json`) with camelCase property names, except where an endpoint states otherwise (creating a note uses `multipart/form-data`, and downloads return the file).

Dates and times. All date-time values are UTC in ISO 8601 format, for example `2026-11-16T08:00:00Z`. Send UTC values in requests; whether an absence is active is evaluated against the current UTC time.

Identifying people. People are identified by their EmployeeId, the employee number held on their Empowering HR record, for example `100234`. It is a string.

Identifiers. Absences, case topics and stored files are identified by GUIDs. Cases, notes, escalations and documents are identified by integers.

Enumerations. Enumerated values are returned as strings, for example `"status": "Open"`.

Acting on behalf of a person. Endpoints that read case content require the `X-On-Behalf-Of-User` header, holding the EmployeeId of the person the request is made for. The response is limited to what that person is permitted to see. The endpoints that need it say so.

Correlation. Most endpoints accept an optional `X-CorrelationId` header containing a GUID of your choosing. Quote it when raising a support query so that the request can be traced.

Pagination. List endpoints accept `pageNumber` (default 1) and `pageSize` (default 10, maximum 100) and return this envelope:

```
{
  "items": [],
  "pageNumber": 1,
  "totalPages": 3,
  "totalCount": 27,
  "hasPreviousPage": false,
  "hasNextPage": true
}
```

Field	Type	Description
`items`	array	The results on this page.
`pageNumber`	integer	The page returned.
`totalPages`	integer	The number of pages available.
`totalCount`	integer	The total number of results across all pages.
`hasPreviousPage`	boolean	Whether a page exists before this one.
`hasNextPage`	boolean	Whether a page exists after this one.

Availability. Endpoint groups are enabled per environment. If every call to a group returns `404` with a valid token, that group has not yet been enabled for your environment; contact your AdviserPlus representative.

5 Errors

Errors use standard HTTP status codes and are returned as RFC 7807 problem details:

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.4",
  "title": "Not Found",
  "status": 404,
  "detail": "The requested resource was not found."
}
```

When individual request fields fail validation the response is `422` and an `errors` object lists the messages for each field:

```
{
  "title": "One or more validation errors occurred.",
  "status": 422,
  "errors": {
    "AbsenceTypeCode": [
      "An absence type must be supplied."
    ],
    "StartDateTime": [
      "'Start Date Time' must not be empty."
    ]
  }
}
```

The following can be returned by any endpoint:

Status	Meaning	What to do
400	Bad Request	The request is malformed. Check the detail message.
401	Unauthorized	The access token is missing, expired or invalid. Request a new token and retry.
403	Forbidden	The credential does not have access to this endpoint. Contact your AdviserPlus representative.
404	Not Found	The resource does not exist, or the endpoint group is not enabled for your environment.
409	Conflict	The request conflicts with existing data, for example a duplicate reference.
422	Unprocessable Entity	The request is well formed but cannot be processed. Check detail or errors.
500	Internal Server Error	An unexpected error occurred. Retry with backoff; if it persists, contact support quoting your X-CorrelationId.

6 Endpoint summary

Area	Operation	Method	Path
Absences	List absence types	GET	/external/absence-types
Absences	List absence reasons	GET	/external/absence-reasons
Absences	Create an absence	POST	/external/absences
Absences	List absences	GET	/external/absences
Absences	Get an absence	GET	/external/absences/{absenceId}
Absences	Update an absence	PATCH	/external/absences/{absenceId}
Absences	Delete an absence	DELETE	/external/absences/{absenceId}
Cases	List case topics	GET	/case-topics
Cases	Create a case	POST	/external/cases
Cases	List an employee's cases	GET	/external/employees/{employeeId}/cases
Case notes and attachments	List case notes	GET	/external/cases/{caseId}/notes
Case notes and attachments	Get a case note	GET	/external/cases/{caseId}/notes/{noteId}
Case notes and attachments	Create a case note	POST	/external/cases/{caseId}/notes
Case notes and attachments	Download a case attachment	GET	/external/cases/{caseId}/attachments/{storageId}/download
Case notes and attachments	Download a case note document	GET	/external/cases/{caseId}/notes/{noteId}/documents/{documentId}/download
Case escalations	List case escalations	GET	/external/cases/{caseId}/escalations
Case escalations	Get a case escalation	GET	/external/cases/{caseId}/escalations/{escalationId}

7 Absence integration walkthrough

A time and attendance system that sends absences to Empowering HR would typically use the endpoints below.

1. **Obtain a token** from the authorisation server and cache it.
2. **Load reference data.** Call List absence types and List absence reasons and map your own absence codes to the Empower codes. These lists change rarely and can be cached and refreshed periodically.
3. **When an absence starts**, call Create an absence with the EmployeeId, the absence type code, the start date and time and, where known, the absence reason code. Send your own absence identifier as `externalId` and store the `id` that is returned.
4. **When the absence changes or ends**, call Update an absence with only the fields that changed, for example `endDateTime` on return to work.
5. **When an absence was recorded in error**, call Delete an absence.
6. **To reconcile**, call List absences filtered by `externalId` or by `employeeId`.

`externalId` must be unique, so a repeated create for the same absence is rejected with `409` rather than creating a duplicate. If a create request times out, look the absence up by `externalId` before retrying.

Where the integration should also open a case, call List case topics to find the topic and Create a case with the employee as `caseSubject` and their manager or case handler as `owner`.

8 Absences

Record, maintain and query employee absences, and look up the absence type and reason codes that absences are recorded against.

8.1 List absence types

```
GET  /external/absence-types
```

Returns a page of the absence types that are currently enabled, ordered by code. Use the `code` values as `absenceTypeCode` when creating or updating an absence.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Query parameters

Name	Type	Required	Description
`pageNumber`	integer	No	Page to return. Minimum 1. Default 1.
`pageSize`	integer	No	Results per page. Minimum 1, maximum 100. Default 10.

Example request

```
GET /api/external/absence-types HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
```

Example response - `200 OK`

```
{
  "items": [
    {
      "code": "SICK",
      "description": "Sickness"
    },
    {
      "code": "UNAUTH",
      "description": "Unauthorised absence"
    }
  ],
  "pageNumber": 1,
  "totalPages": 1,
  "totalCount": 2,
  "hasPreviousPage": false,
  "hasNextPage": false
}
```

Response fields

Field	Type	Description
`items[].code`	string	The code that identifies the absence type.

Field	Type	Description
`items[].description`	string, nullable	The description of the absence type.

The paging fields are described under Pagination.

Errors (in addition to the common errors)

Status	When
400	pageNumber or pageSize is out of range.

8.2 List absence reasons

`GET /external/absence-reasons`

Returns a page of the absence reasons that are currently enabled, ordered by code. Use the `code` values as `absenceReasonCode` when creating or updating an absence.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Query parameters

Name	Type	Required	Description
`pageNumber`	integer	No	Page to return. Minimum 1. Default 1.
`pageSize`	integer	No	Results per page. Minimum 1, maximum 100. Default 10.

Example request

```
GET /api/external/absence-reasons HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
```

Example response - `200 OK`

```
{
  "items": [
    {
      "code": "BACK",
      "description": "Back problems"
    },
    {
      "code": "COLD_FLU",
      "description": "Cold / flu"
    }
  ],
  "pageNumber": 1,
  "totalPages": 1,
  "totalCount": 2,
  "hasPreviousPage": false,
  "hasNextPage": false
}
```

Response fields

Field	Type	Description
`items[].code`	string	The code that identifies the absence reason.
`items[].description`	string, nullable	The description of the absence reason.

The paging fields are described under Pagination.

Errors (in addition to the common errors)

Status	When
400	pageNumber or pageSize is out of range.

8.3 Create an absence

`POST /external/absences`

Creates an absence for an employee and returns it, including its Empower identifier (`id`) and the `externalId` you supplied. Store the returned `id`: it is the key for the get, update and delete operations.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Request body (`application/json`)

Field	Type	Required	Description
`employeeId`	string	Yes	The EmployeeId of the person the absence is for.
`absenceTypeCode`	string	Yes	Code of an enabled absence type. At most 200 characters.
`startDateTime`	date-time	Yes	Date and time the absence starts (UTC).
`endDateTime`	date-time	No	Date and time the absence ends (UTC). Omit or send null for an open-ended absence.
`absenceReasonCode`	string	No	Code of an enabled absence reason. At most 200 characters.
`externalId`	string	No	Your own reference for the absence. At most 50 characters. Must be unique.

Example request

```
POST /api/external/absences HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
Content-Type: application/json

{
  "employeeId": "100234",
  "absenceTypeCode": "SICK",
  "startDateTime": "2026-11-16T08:00:00Z",
  "endDateTime": null,
  "absenceReasonCode": "COLD_FLU",
  "externalId": "MYTIME-000123456"
}
```

Example response - 201 Created

```
{
  "id": "8f2b6c1e-4d7a-4c1b-9a3e-5b7d2f9c0a11",
  "externalId": "MYTIME-000123456",
  "employeeId": "100234",
  "type": {
    "code": "SICK",
    "description": "Sickness"
  },
  "reason": {
    "code": "COLD_FLU",
    "description": "Cold / flu"
  },
  "startDateTime": "2026-11-16T08:00:00Z",
  "endDateTime": null,
  "isActive": true
}
```

Response fields

Field	Type	Description
`id`	GUID	The Empower identifier of the absence.
`externalId`	string, nullable	Your own reference for the absence.
`employeeId`	string, nullable	The EmployeeId of the employee the absence belongs to.
`type.code`	string	The code of the absence type.
`type.description`	string, nullable	The description of the absence type.
`reason`	object, nullable	The reason for the absence. Null when none is recorded.
`reason.code`	string	The code of the absence reason.
`reason.description`	string, nullable	The description of the absence reason.
`startDateTime`	date-time, nullable	When the absence starts (UTC).
`endDateTime`	date-time, nullable	When the absence ends (UTC). Null for an open-ended absence.

Field	Type	Description
`isActive`	boolean	True once the absence has started and until it ends.

Errors (in addition to the common errors)

Status	When
400	The body is missing or is not valid JSON.
404	No employee exists with the supplied employeeId.
409	An absence with the supplied externalId already exists.
422	A field failed validation, or the absence type or reason code was not found or is not enabled.

Example `409` response:

```
{
  "title": "Conflict",
  "status": 409,
  "detail": "An absence with the supplied external ID already exists."
}
```

8.4 List absences

`GET /external/absences`

Returns a page of absences, most recently recorded first. All filters are optional and can be combined. Deleted absences are never returned. Filtering by `externalId` is the way to find the Empower `id` of an absence from your own reference.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Query parameters

Name	Type	Required	Description
`employeeId`	string	No	Return only absences belonging to this employee.
`externalId`	string	No	Return only the absence with this external reference. Returns 404 when nothing matches.
`isActive`	boolean	No	true returns absences that have started and have no end date or an end date in the future. false returns all others. Omit for both.
`pageNumber`	integer	No	Page to return. Minimum 1. Default 1.
`pageSize`	integer	No	Results per page. Minimum 1, maximum 100. Default 10.

Example request

```
GET /api/external/absences?employeeId=100234&isActive=true HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
```

Example response - 200 OK


```
{
  "items": [
    {
      "id": "8f2b6c1e-4d7a-4c1b-9a3e-5b7d2f9c0a11",
      "externalId": "MYTIME-000123456",
      "employeeId": "100234",
      "type": {
        "code": "SICK",
        "description": "Sickness"
      },
      "reason": {
        "code": "COLD_FLU",
        "description": "Cold / flu"
      },
      "startDateTime": "2026-11-16T08:00:00Z",
      "endDateTime": null,
      "isActive": true
    }
  ],
  "pageNumber": 1,
  "totalPages": 1,
  "totalCount": 1,
  "hasPreviousPage": false,
  "hasNextPage": false
}
```


Response fields

Field	Type	Description
items[].id	GUID	The Empower identifier of the absence.
items[].externalId	string, nullable	Your own reference for the absence.
items[].employeeId	string, nullable	The EmployeeId of the employee the absence belongs to.
items[].type.code	string	The code of the absence type.
items[].type.description	string, nullable	The description of the absence type.
items[].reason	object, nullable	The reason for the absence. Null when none is recorded.
items[].reason.code	string	The code of the absence reason.
items[].reason.description	string, nullable	The description of the absence reason.
items[].startDateTime	date-time, nullable	When the absence starts (UTC).
items[].endDateTime	date-time, nullable	When the absence ends (UTC). Null for an open-ended absence.
items[].isActive	boolean	True once the absence has started and until it ends.

The paging fields are described under Pagination.

Errors (in addition to the common errors)

Status	When
400	isActive, pageNumber or pageSize is not valid.
404	An externalId filter was supplied and no absence matches it.

8.5 Get an absence

```
GET  /external/absences/{absenceId}
```

Returns the absence with the specified Empower identifier. Deleted absences are not returned.

Path parameters

Name	Type	Description
`absenceId`	GUID	The Empower identifier of the absence.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Example request

```
GET /api/external/absences/8f2b6c1e-4d7a-4c1b-9a3e-5b7d2f9c0a11 HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
```

Example response - `200 OK`

```
{
  "id": "8f2b6c1e-4d7a-4c1b-9a3e-5b7d2f9c0a11",
  "externalId": "MYTIME-000123456",
  "employeeId": "100234",
  "type": {
    "code": "SICK",
    "description": "Sickness"
  },
  "reason": {
    "code": "COLD_FLU",
    "description": "Cold / flu"
  },
  "startDateTime": "2026-11-16T08:00:00Z",
  "endDateTime": null,
  "isActive": true
}
```

Response fields

Field	Type	Description
`id`	GUID	The Empower identifier of the absence.
`externalId`	string, nullable	Your own reference for the absence.

Field	Type	Description
`employeeId`	string, nullable	The EmployeeId of the employee the absence belongs to.
`type.code`	string	The code of the absence type.
`type.description`	string, nullable	The description of the absence type.
`reason`	object, nullable	The reason for the absence. Null when none is recorded.
`reason.code`	string	The code of the absence reason.
`reason.description`	string, nullable	The description of the absence reason.
`startDateTime`	date-time, nullable	When the absence starts (UTC).
`endDateTime`	date-time, nullable	When the absence ends (UTC). Null for an open-ended absence.
`isActive`	boolean	True once the absence has started and until it ends.

Errors (in addition to the common errors)

Status	When
404	No absence exists with this identifier, or it has been deleted.

8.6 Update an absence

```
PATCH  /external/absences/{absenceId}
```

Updates only the fields you supply; omitted fields are left unchanged. At least one field must be supplied and unknown fields are rejected. The typical use is to send `endDateTime` when the employee returns to work. The employee and `externalId` of an absence cannot be changed.

Path parameters

Name	Type	Description
`absenceId`	GUID	The Empower identifier of the absence.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Request body (`application/json`)

Field	Type	Required	Description
`absenceTypeCode`	string	No	New absence type code. Null is not accepted.
`startDateTime`	date-time	No	New start date and time (UTC). Null is not accepted.

Field	Type	Required	Description
endDateTime	date-time	No	New end date and time (UTC). Send null to clear the end date.
absenceReasonCode	string	No	New absence reason code. Send null to clear the reason.

Example request

```
PATCH /api/external/absences/8f2b6c1e-4d7a-4c1b-9a3e-5b7d2f9c0a11 HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
Content-Type: application/json

{
  "endDateTime": "2026-11-19T17:00:00Z"
}
```

Example response - 200 OK

```
{
  "id": "8f2b6c1e-4d7a-4c1b-9a3e-5b7d2f9c0a11",
  "externalId": "MYTIME-000123456",
  "employeeId": "100234",
  "type": {
    "code": "SICK",
    "description": "Sickness"
  },
  "reason": {
    "code": "COLD_FLU",
    "description": "Cold / flu"
  },
  "startDateTime": "2026-11-16T08:00:00Z",
  "endDateTime": "2026-11-19T17:00:00Z",
  "isActive": false
}
```

Response fields

Field	Type	Description
id	GUID	The Empower identifier of the absence.
externalId	string, nullable	Your own reference for the absence.
employeeId	string, nullable	The EmployeeId of the employee the absence belongs to.
type.code	string	The code of the absence type.
type.description	string, nullable	The description of the absence type.
reason	object, nullable	The reason for the absence. Null when none is recorded.
reason.code	string	The code of the absence reason.
reason.description	string, nullable	The description of the absence reason.
startDateTime	date-time, nullable	When the absence starts (UTC).

Field	Type	Description
`endDateTime`	date-time, nullable	When the absence ends (UTC). Null for an open-ended absence.
`isActive`	boolean	True once the absence has started and until it ends.

Errors (in addition to the common errors)

Status	When
400	The body is missing, is not valid JSON, or contains an unknown field.
404	No absence exists with this identifier, or it has been deleted.
422	No fields were supplied, a field failed validation, or the absence type or reason code was not found or is not enabled.

Example `422` response:

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "detail": "Absence reason was not found or is not enabled."
}
```

8.7 Delete an absence

```
DELETE  /external/absences/{absenceId}
```

Deletes an absence. An absence that is linked to a case cannot be deleted unless that case has been marked invalid.

Path parameters

Name	Type	Description
`absenceId`	GUID	The Empower identifier of the absence.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Example request

```
DELETE /api/external/absences/8f2b6c1e-4d7a-4c1b-9a3e-5b7d2f9c0a11 HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
```

Example response - `204 No Content`

No response body.

Errors (in addition to the common errors)

Status	When
404	No absence exists with this identifier, or it has already been deleted.
422	The absence is linked to a case and cannot be deleted.

Example 422 response:

```
{
  "title": "Unprocessable Entity",
  "status": 422,
  "detail": "The absence is linked to a case and cannot be deleted."
}
```

9 Cases

Look up case topics, create cases and list the cases an employee is the subject of.

9.1 List case topics

`GET /case-topics`

Returns the case topics enabled for your organisation. Use a topic `id` as `caseTopicId` when creating a case. The result is a plain array and is not paginated. Note that this route has no `/external` segment.

Query parameters

Name	Type	Required	Description
`employeeId`	string	No	Return only the topics available to this employee's primary access level.

Example request

```
GET /api/case-topics?employeeId=100234 HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
```

Example response - `200 OK`

```
[
  {
    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "name": "Absence"
  },
  {
    "id": "7cb2e4a1-8d3f-4b1e-9c5a-6d2f8e4b3a1c",
    "name": "Disciplinary"
  }
]
```

Response fields

Field	Type	Description
`[].id`	GUID	The identifier of the case topic.
`[].name`	string	The display name of the topic.

9.2 Create a case

`POST /external/cases`

Creates a new case about an employee (`caseSubject`) and assigns it to an owner. The workflow the case follows is determined by the topic and the subject's primary access level.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Request body (`application/json`)

Field	Type	Required	Description
`caseTopicId`	GUID	Yes	The case topic, from List case topics.
`caseSubject`	string	Yes	The EmployeeId of the person the case is about.
`owner`	string	Yes	The EmployeeId of the person who will own the case. Must differ from caseSubject.

Example request

```
POST /api/external/cases HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
Content-Type: application/json

{
  "caseTopicId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "caseSubject": "100234",
  "owner": "100871"
}
```

Example response - `201 Created`

```
{
  "id": 48213,
  "status": "Open",
  "createdAt": "2026-11-16T08:05:12.431Z"
}
```

Response fields

Field	Type	Description
`id`	integer	The case identifier.
`status`	string	Open, Closed or RemovedInvalid.
`createdAt`	date-time	When the case was created (UTC).

Errors (in addition to the common errors)

Status	When
400	The body is missing or is not valid JSON.
409	An open case already exists for the employee on a workflow that does not allow multiple cases.
422	A field failed validation, or the topic, subject or owner could not be resolved, or no enabled workflow exists for the topic at the subject's access level.

Example `409` response:

```
{
  "title": "Conflict",
  "status": 409,
  "detail": "An open case already exists for employee on this workflow"
}
```

9.3 List an employee's cases

```
GET  /external/employees/{employeeId}/cases
```

Returns a page of the cases the employee is the subject of, newest first. Invalid cases are never returned, and cases that the on-behalf-of person is blocked from are left out.

Path parameters

Name	Type	Description
`employeeId`	string	The EmployeeId of the employee whose cases are returned.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-On-Behalf-Of-User`	string	Yes	The EmployeeId of the person the request is made on behalf of. One value, at most 100 characters.
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Query parameters

Name	Type	Required	Description
`open`	boolean	No	true returns only open cases, false only closed cases. Omit for both.
`pageNumber`	integer	No	Page to return. Minimum 1. Default 1.
`pageSize`	integer	No	Results per page. Minimum 1, maximum 100. Default 10.

Example request

```
GET /api/external/employees/100234/cases?open=true HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
X-On-Behalf-Of-User: 100871
```

Example response - 200 OK


```
{
  "items": [
    {
      "id": 48213,
      "topic": {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "name": "Absence"
      },
      "status": "Open",
      "owner": {
        "employeeId": "100871",
        "name": "Priya Shah"
      },
      "createdAt": "2026-11-16T08:05:12.431Z",
      "lastActionAt": "2026-11-17T10:22:40.118Z",
      "closedAt": null
    }
  ],
  "pageNumber": 1,
  "totalPages": 1,
  "totalCount": 1,
  "hasPreviousPage": false,
  "hasNextPage": false
}
```


Response fields

Field	Type	Description
`items[].id`	integer	The case identifier.
`items[].topic.id`	GUID	The identifier of the case topic.
`items[].topic.name`	string, nullable	The display name of the topic.
`items[].status`	string	Open or Closed.
`items[].owner`	object, nullable	The current owner of the case.
`items[].owner.employeeId`	string, nullable	The EmployeeId of the owner, when one is recorded.
`items[].owner.name`	string	The full name of the owner.
`items[].createdAt`	date-time	When the case was created (UTC).
`items[].lastActionAt`	date-time, nullable	When the most recent activity on the case occurred (UTC).
`items[].closedAt`	date-time, nullable	When the case was closed (UTC). Null unless closed.

The paging fields are described under Pagination.

Errors (in addition to the common errors)

Status	When
400	The X-On-Behalf-Of-User header is missing, or a query parameter is not valid.
404	No employee exists with the supplied employeeId.

Status	When
422	The on-behalf-of employee was not found.

10 Case notes and attachments

Read and create notes on a case and download the files attached to them. Read operations are performed on behalf of a named person: notes marked `Private` are only returned when that person is an HR user.

10.1 List case notes

```
GET  /external/cases/{caseId}/notes
```

Returns the visible notes and replies on a case as a flat, paginated list, newest first. Draft notes are excluded. Replies carry the identifier of their parent in `parentNoteId`. Reading notes does not mark them as read.

Path parameters

Name	Type	Description
`caseId`	integer	The identifier of the case.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-On-Behalf-Of-User`	string	Yes	The EmployeeId of the person the request is made on behalf of. One value, at most 100 characters.
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Query parameters

Name	Type	Required	Description
`pageNumber`	integer	No	Page to return. Minimum 1. Default 1.
`pageSize`	integer	No	Results per page. Minimum 1, maximum 100. Default 10.

Example request

```
GET /api/external/cases/48213/notes HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
X-On-Behalf-Of-User: 100871
```

Example response - 200 OK


```
{
  "items": [
    {
      "id": 912004,
      "parentNoteId": null,
      "visibility": "Public",
      "author": {
        "id": "5d0c3b7e-2a49-4f6e-8c11-9e4a7b2d6f30",
        "name": "Priya Shah"
      },
      "createdAt": "2026-11-17T10:22:40.118Z",
      "attachmentCount": 1
    }
  ],
  "pageNumber": 1,
  "totalPages": 1,
  "totalCount": 1,
  "hasPreviousPage": false,
  "hasNextPage": false
}
```


Response fields

Field	Type	Description
`items[].id`	integer	The identifier of the note.
`items[].parentNoteId`	integer, nullable	The parent note for a reply. Null for a top-level note.
`items[].visibility`	string	Public or Private.
`items[].author.id`	GUID	The identifier of the author.
`items[].author.name`	string, nullable	The display name of the author.
`items[].createdAt`	date-time	When the note was created (UTC).
`items[].attachmentCount`	integer	The number of files and documents attached to the note.

The paging fields are described under Pagination.

Errors (in addition to the common errors)

Status	When
400	The X-On-Behalf-Of-User header is missing or not valid, or a paging parameter is out of range.
404	The case does not exist.
422	The on-behalf-of employee was not found, or cannot view this case.

10.2 Get a case note

`GET /external/cases/{caseId}/notes/{noteId}`

Returns the subject, sanitised body, recorded email recipients and attachments of a visible note or reply. Use each attachment's `downloadUrl` (relative to the API base URL) to download the file.

Path parameters

Name	Type	Description
caseId	integer	The identifier of the case.
noteId	integer	The identifier of the note or reply.

Headers (in addition to Authorization)

Name	Type	Required	Description
X-On-Behalf-Of-User	string	Yes	The EmployeeId of the person the request is made on behalf of. One value, at most 100 characters.
X-CorrelationId	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Example request

```
GET /api/external/cases/48213/notes/912004 HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
X-On-Behalf-Of-User: 100871
```

Example response - 200 OK

```
{
  "id": 912004,
  "parentNoteId": null,
  "visibility": "Public",
  "author": {
    "id": "5d0c3b7e-2a49-4f6e-8c11-9e4a7b2d6f30",
    "name": "Priya Shah"
  },
  "createdAt": "2026-11-17T10:22:40.118Z",
  "attachmentCount": 1,
  "subject": "Return to work meeting",
  "body": "<p>Return to work meeting held on 17 November. Fit note attached.</p>",
  "attachments": [
    {
      "storageId": "c0a8f1d2-6b3e-4e59-a7c4-1f2e3d4c5b6a",
      "fileName": "fit-note.pdf",
      "downloadUrl": "external/cases/48213/attachments/c0a8f1d2-6b3e-4e59-a7c4-
1f2e3d4c5b6a/download"
    }
  ],
  "traditionalDocumentAttachments": [
    {
      "documentId": 3305,
      "fileName": "Return to work form"
    }
  ],
  "recipients": [
    {
      "type": "To",
      "email": "priya.shah@example.com",
      "sentFromUserId": "5d0c3b7e-2a49-4f6e-8c11-9e4a7b2d6f30",
      "sentToUserId": "b71e9a04-3c5d-4d2f-9f68-0a1b2c3d4e5f"
    }
  ]
}
```

Response fields

Field	Type	Description
`id, parentNoteId, visibility, author, createdAt, attachmentCount`		As for List case notes.
`subject`	string	The note subject. Empty when none was recorded.
`body`	string	The sanitised note body. Empty when none was recorded.
`attachments[].storageId`	GUID	The storage identifier of the attached file.
`attachments[].fileName`	string	The name of the attached file.
`attachments[].downloadUrl`	string	API-relative link for downloading the file.
`traditionalDocumentAttachments[].documentId`	integer	The identifier of an attached library document.
`traditionalDocumentAttachments[].fileName`	string	The name of the library document.
`recipients[].type`	string	From, To, Cc or None.
`recipients[].email`	string, nullable	The recorded email address.
`recipients[].sentFromUserId`	GUID	The recorded sender.
`recipients[].sentToUserId`	GUID	The recorded recipient.

Errors (in addition to the common errors)

Status	When
400	The X-On-Behalf-Of-User header is missing or not valid.
404	The case or note does not exist, or the note is not visible to the on-behalf-of person.
422	The on-behalf-of employee was not found, or cannot view this case.

10.3 Create a case note

`POST /external/cases/{caseId}/notes`

Creates a public note on a case, optionally with file attachments, in a single `multipart/form-data` request. The note and all of its attachments are created together or not at all.

Path parameters

Name	Type	Description
`caseId`	integer	The identifier of the case.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Request body (`multipart/form-data`)

Field	Type	Required	Description
`metadata`	JSON part	Yes	A form part named metadata containing the JSON object below.
`metadata.subject`	string	Yes	The subject of the note. 1 to 500 characters.
`metadata.body`	string	Yes	The body of the note. 1 to 100,000 characters.
`metadata.employeeId`	string	Yes	The EmployeeId of the person the note is created on behalf of. They must be able to add notes to the case.
`(file parts)`	file	No	Zero or more file parts, each with a filename. Each file may be up to 10 MB by default and must be of an allowed file type. A request may contain at most 50 parts.

Example request

```
POST /api/external/cases/48213/notes HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
Content-Type: multipart/form-data; boundary=----boundary

------boundary
Content-Disposition: form-data; name="metadata"
Content-Type: application/json

{
  "subject": "Absence reported",
  "body": "Employee reported absent via MyTime on 16 November.",
  "employeeId": "100871"
}
------boundary
Content-Disposition: form-data; name="files"; filename="fit-note.pdf"
Content-Type: application/pdf

<file bytes>
------boundary--
```

Example response - `201 Created`

```
{
  "noteId": 912011,
  "attachments": [
    {
      "id": 77120,
      "fileName": "fit-note.pdf"
    }
  ]
}
```

Response fields

Field	Type	Description
`noteId`	integer	The identifier of the created note.
`attachments[].id`	integer	The identifier of the attachment.
`attachments[].fileName`	string	The file name of the attachment.

Errors (in addition to the common errors)

Status	When
400	The request is not valid multipart/form-data, the metadata part is missing or fails validation, or a file is too large or of a disallowed type.
404	The case does not exist.
422	The employeeId could not be resolved, or that person cannot add a note to this case.

Example `400` response:

```
{
  "title": "Bad Request",
  "status": 400,
  "detail": "The following file(s) have a disallowed file type: macro.exe"
}
```

10.4 Download a case attachment

`GET /external/cases/{caseId}/attachments/{storageId}/download`

Returns a file attached to a case note or escalation. The response body is the file itself; the `Content-Type` header carries its media type and `Content-Disposition` its file name. Attachments on private notes and on escalations are only returned when the on-behalf-of person is an adviser.

Path parameters

Name	Type	Description
`caseId`	integer	The identifier of the case.
`storageId`	GUID	The storage identifier of the attachment, from a note or escalation.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-On-Behalf-Of-User`	string	Yes	The EmployeeId of the person the request is made on behalf of. One value, at most 100 characters.
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Example request

```
GET /api/external/cases/48213/attachments/c0a8f1d2-6b3e-4e59-a7c4-1f2e3d4c5b6a/download HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
X-On-Behalf-Of-User: 100871
```

Example response - `200 OK`

```
HTTP/1.1 200 OK
Content-Type: application/pdf
Content-Disposition: attachment; filename="fit-note.pdf"

<file bytes>
```

Errors (in addition to the common errors)

Status	When
400	The X-On-Behalf-Of-User header is missing.
404	The case or attachment does not exist, or is not visible to the on-behalf-of person.
422	The on-behalf-of employee was not found, or cannot view this case.

10.5 Download a case note document

```
GET  /external/cases/{caseId}/notes/{noteId}/documents/{documentId}/download
```

Returns a library document attached to a case note or escalation (listed under `traditionalDocumentAttachments`). The response body is the file itself, named after the document's title. Documents on private notes and on escalations are only returned when the on-behalf-of person is an adviser.

Path parameters

Name	Type	Description
`caseId`	integer	The identifier of the case.
`noteId`	integer	The identifier of the note or escalation the document is attached to.
`documentId`	integer	The identifier of the document.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-On-Behalf-Of-User`	string	Yes	The EmployeeId of the person the request is made on behalf of. One value, at most 100 characters.
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Example request

```
GET /api/external/cases/48213/notes/912004/documents/3305/download HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
X-On-Behalf-Of-User: 100871
```

Example response - `200 OK`

```
HTTP/1.1 200 OK
Content-Type: application/pdf
Content-Disposition: attachment; filename="fit-note.pdf"

<file bytes>
```

Errors (in addition to the common errors)

Status	When
400	The X-On-Behalf-Of-User header is missing.
404	The case, note or document does not exist, or is not visible to the on-behalf-of person.

Status	When
422	The on-behalf-of employee was not found, or cannot view this case.

11 Case escalations

Read the escalations raised on a case. Escalations are adviser-only communications.

11.1 List case escalations

`GET /external/cases/{caseId}/escalations`

Returns a page of the published escalations on a case, newest first.

Path parameters

Name	Type	Description
`caseId`	integer	The identifier of the case.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-On-Behalf-Of-User`	string	Yes	The EmployeeId of the person the request is made on behalf of. One value, at most 100 characters.
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Query parameters

Name	Type	Required	Description
`pageNumber`	integer	No	Page to return. Minimum 1. Default 1.
`pageSize`	integer	No	Results per page. Minimum 1, maximum 100. Default 10.

Example request

```
GET /api/external/cases/48213/escalations HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
X-On-Behalf-Of-User: 100871
```

Example response - `200 OK`

```
{
  "items": [
    {
      "id": 5521,
      "author": "Daniel Okafor",
      "createdAt": "2026-11-18T14:03:55.902Z",
      "attachmentCount": 1
    }
  ],
  "pageNumber": 1,
  "totalPages": 1,
  "totalCount": 1,
  "hasPreviousPage": false,
  "hasNextPage": false
}
```

Response fields

Field	Type	Description
`items[].id`	integer	The identifier of the escalation.
`items[].author`	string	The full name of the escalation author.
`items[].createdAt`	date-time	When the escalation was created (UTC).
`items[].attachmentCount`	integer	The number of files attached to the escalation.

The paging fields are described under Pagination.

Errors (in addition to the common errors)

Status	When
400	The X-On-Behalf-Of-User header is missing or not valid, or a paging parameter is out of range.
404	The case does not exist.
422	The on-behalf-of employee was not found, or cannot view this case.

11.2 Get a case escalation

`GET /external/cases/{caseId}/escalations/{escalationId}`

Returns a published escalation, the people it was sent to and its attachments.

Path parameters

Name	Type	Description
`caseId`	integer	The identifier of the case.
`escalationId`	integer	The identifier of the escalation.

Headers (in addition to `Authorization`)

Name	Type	Required	Description
`X-On-Behalf-Of-User`	string	Yes	The EmployeeId of the person the request is made on behalf of. One value, at most 100 characters.
`X-CorrelationId`	GUID	No	Caller-supplied GUID used to trace the request end to end. Send the same value on related calls to correlate them.

Example request

```
GET /api/external/cases/48213/escalations/5521 HTTP/1.1
Host: api.uat.empoweringhr.com
Authorization: Bearer <access_token>
X-On-Behalf-Of-User: 100871
```

Example response - 200 OK


```
{
  "id": 5521,
  "escalationReason": "Complex case",
  "escalatedTo": [
    {
      "employeeId": "100950",
      "name": "Helen Marsh"
    }
  ],
  "body": "<p>Third absence in six months. Please review before the stage 2 meeting.</p>",
  "attachments": [
    {
      "storageId": "9e1d4c7a-0b2f-4a6d-8c3e-7f5a6b4c3d2e",
      "fileName": "absence-history.xlsx",
      "downloadUrl": "external/cases/48213/attachments/9e1d4c7a-0b2f-4a6d-8c3e-
7f5a6b4c3d2e/download"
    }
  ]
}
```


Response fields

Field	Type	Description
`id`	integer	The identifier of the escalation.
`escalationReason`	string	The configured reason for the escalation.
`escalatedTo[].employeeId`	string, nullable	The EmployeeId of a recipient, when one is recorded.
`escalatedTo[].name`	string	The full name of the recipient.
`body`	string	The sanitised escalation message body.
`attachments[].storageId`	GUID	The storage identifier of the attached file.
`attachments[].fileName`	string	The name of the attached file.
`attachments[].downloadUrl`	string	API-relative link for downloading the file.

Errors (in addition to the common errors)

Status	When
400	The X-On-Behalf-Of-User header is missing or not valid.
404	The case or escalation does not exist.
422	The on-behalf-of employee was not found, or cannot view this case.

12 Using the Postman collection

1. In Postman choose Import and select the collection file and the environment file for the environment you are using.
2. Select the environment and set `clientId` and `clientSecret` to the credentials you were issued. Enter them as current values so that they are not synced or exported.
3. Set `employeeId`, `ownerEmployeeId` and `onBehalfOfEmployeeId` to EmployeeIds that exist in that environment, and `absenceTypeCode` and `absenceReasonCode` to codes returned by the List absence types and List absence reasons requests.
4. Send any request. The collection requests an access token automatically and renews it when it expires; the Get access token request can also be run on its own to check your credentials.

Requests that create or list resources store the identifiers they return (`absenceId`, `caseTopicId`, `caseId`, `noteId`, `storageId`, `documentId`, `escalationId`) in the environment so that the requests that follow can be run without copying values by hand. Each request also has a saved example showing a typical response.