# Endeavour Enterprise REST Specification

Contents

[Endeavour Enterprise REST Specification 1](#_Toc444177942)

[/api/security 3](#_Toc444177943)

[/api/security/login 3](#_Toc444177944)

[/api/security/selectOrganisation 3](#_Toc444177945)

[/api/security/changePassword 4](#_Toc444177946)

[/api/security/logoff 4](#_Toc444177947)

[/api/security/setPasswordFromInviteEmail 4](#_Toc444177948)

[/api/admin 5](#_Toc444177949)

[/api/admin/getUsers 5](#_Toc444177950)

[/api/admin/saveUser 5](#_Toc444177951)

[/api/admin/deleteUser 6](#_Toc444177952)

[/api/admin/resendInviteEmail 6](#_Toc444177953)

[/api/admin/saveOrganisation 6](#_Toc444177954)

[/api/folder 7](#_Toc444177955)

[/api/folder/saveFolder 7](#_Toc444177956)

[/api/folder/deleteFolder 7](#_Toc444177957)

[/api/folder/getFolders 7](#_Toc444177958)

[/api/folder/getFolderContents 8](#_Toc444177959)

[/api/report/ 9](#_Toc444177960)

[/api/report/getReport 9](#_Toc444177961)

[/api/report/saveReport 9](#_Toc444177962)

[/api/report/deleteReport 9](#_Toc444177963)

[/api/query 11](#_Toc444177964)

[/api/query/getQuery 11](#_Toc444177965)

[/api/query/saveQuery 11](#_Toc444177966)

[/api/query/deleteQuery 12](#_Toc444177967)

[/api/listOutput 13](#_Toc444177968)

[/api/listOutput/getListOutput 13](#_Toc444177969)

[/api/listOutput/saveListOutput 13](#_Toc444177970)

[/api/listOutput/deleteListOutput 13](#_Toc444177971)

# /api/security

This domain contains endpoints for authentication and signon

## /api/security/login

Prerequisites: none

Method: POST

Parameters:

{“username”: <email address>,

“password”, <password>}

Response:

Session cookie, plus the below JSON

{“organisations”: [

{“uuid”: <org UUID>,

“name”: <org name>,

“nationalId”: <org ID>,

“permissions”: <1=user, 2=admin>}

…repeated for each organisation the user can log on to

]}

First call, used by a user to authenticate themselves. If the email and password match a record in the EndUser table then that EndUser record and will be returned along with all the Organisations that the Person has access to and the permission level at each. Expectation is that the webapp looks at the organisations returned. If multiple organisations are returned, then the user is presented with a selector to choose the organisation they want to log on to.

## /api/security/selectOrganisation

Prerequisites: authenticated password

Method: POST

Parameters:

{“uuid”: <org UUID to select>}

Response:

New session cookie, recording the newly selected organisation, plus JSON giving the details of the newly selected organisation:

{“uuid”: <org UUID>,

“name”: <org name>,

“nationalId”: <org ID>,

“permissions”: <1=user, 2=admin>}

Optional call, used when a user selects the organisation they want to log on to. Not required if they only have access to a single organisation.

## /api/security/changePassword

Prerequisites: authenticated password

Method: POST

Parameters:

{“password”: <new password>}

Response:

none

Call to change the password for the logged on user. A non-error response indicates success.

## /api/security/logoff

Method: POST

Prerequisites: none

Parameters:

none

Response:

Updated cookie, closing the session

Call to end the session, which will remove the token from the server, preventing any further secure functions using that token. If the token has already been removed, then no error is raised.

## /api/security/setPasswordFromInviteEmail

Method: POST

Prerequisites: none

Parameters:

{“token”: <token from email>,

“password”: <new password>}

Response: cookie with session details

Function to allow the initial password to be set for a user, after they’ve received an invite email. The invite email will show a URL containing a unique token. This token is then sent up, with the new password, to complete the registration process. Once this function is performed, the user is automatically signed on, so they can proceed to perform subsequent functions without having to enter their password again.

Note: this is in the security domain because it’s to do with passwords and a successful request results in an authenticated user.

# /api/admin

This domain contains endpoints for managing user accounts

## /api/admin/getUsers

Method: GET

Prerequisites: authenticated password and an admin user

Parameters:

none

Response:

{“users”, [

{“uuid”: <UUID of user>

“username”: <email address>,

“title”: <title>,

“forename”: <forename>,

“surname”: <surname>,

“isSuperUser”: null,

“permissions”: <1=user, 2=admin>}

…repeated for each user at the organisation

]}

Function to retrieve the list of registered users at the current organisation.

## /api/admin/saveUser

Method: POST

Prerequisites: authenticated password and admin permission

Parameters:

{“uuid”: <optional UUID of user>,

“username”: <email address>,

“title”: <title>,

“forename”: <forename>,

“surname”: <surname>,

“permissions”: <1=user, 2=admin>,

“isSuperUser”: null}

Response:

{“uuid”: <user UUID>}

Creates or updates a user. If a UUID is specified, it updates an existing user. If the UUID is omitted, it creates a new person. In both cases, it validates that the the email address is unique. Also validates to ensure that non-superusers can’t create/promote other users to be superusers. In both cases, the UUID of the user is returned.

## /api/admin/deleteUser

Prerequisites: authenticated password and admin permissions

Parameters:

{“uuid”: <user UUID>}

Response: none

Call used by an admin to delete a person from their organisation, preventing them logging on. If a user attempts to delete themselves, an error is raised in response.

## /api/admin/resendInviteEmail

Method: POST

Prerequisites: authenticated password and admin permission

Parameters:

{“uuid”: <user UUID>}

Response: none

Call to resend the invite email for the given Person.

## /api/admin/saveOrganisation

Method: POST

Prerequisites: authenticated password and a superUser account

Parameters:

{“uuid”: <org UUID to update an existing one>,

“name”: <organisation name>,

“nationalId”: <national ID>}

Response:

{“uuid”: <org UUID>}

Superuser-only Function to create or update an organisation. Sending an UUID will update one, and omitting this will create a new one. In both cases JSON containing the UUID of the organisation is returned.

Validates that an organisation with the same name and ID doesn’t already exist.

# /api/folder

Contains end points for working with folders.

## /api/folder/getFolders

Prerequisites: authenticated password

Method: GET

Parameters:

?folderType=<1=library, 2=reports >&parentUuid=<optional parent>

Response:

{“folders”: [

{“uuid”: <folder UUID>,

“name”: <name>,

“type”: <1=library, 2=reports>,

“parentFolderUuid”: <UUID of parent or not present for top-level ones>,

“contentCount”: <number of items in this folder>}

…repeated for each folder

]}

Returns all folders at the currently selected organisation for the given type. Child folder records will have a ParentUuid attribute, allowing the hierarchy to be easily constructed.

## /api/folder/getFolderContents

Prerequisites: authenticated password

Method: GET

Parameters:

?folderUuid=<folder UUID>

Response:

{“contents”: [

{“uuid”: <uuid of item>,

“type”: <1=report, 2=query, 6=list output>.

“typeDesc”: <textual representation of the type>,

“lastModified”: <date time>,

“lastRun”: <only present for reports>,

“isScheduled”: <only present for reports>}

…repeated for each query in the folder

]}

Returns details of the content of a folder, including all queries, outputs and reports in that folder. Depending on the folder type (library or reports), either only reports or a mix of queries and outputs will be returned.

## /api/folder/saveFolder

Prerequisites: authenticated password

Method: POST

Parameters:

{“uuid”: <folder UUID when updating>,

"folderName": <folder name>,

“folderType”: <1=library, 2=reports>,

“parentFolderUuid”: <parent folder UUID>}

Response:

{“uuid”: <folder UUID>}

Creates or updates a folder. Specifying a folderUuid element will update an existing folder with that UUID, changing the folder name or parent as specified (note, changing the folder type is not permitted). Omitting the folderUuid element will create a new folder and issue a new UUID. In both cases the UUID of the folder is returned.

## /api/folder/deleteFolder

Prerequisites: authenticated password

Method: POST

Parameters:

{“uuid”: <folder UUID >}

Response:

none

Deletes a folder and all contents in it, including reports, queries and sub-folders etc.

# /api/report/

Domain for endpoints relating to report entities

## /api/report/getReport

Prerequisites: authenticated password

Method: GET

Parameters:

?uuid=<report UUID>

Response:

{“uuid”: <report UUID>,

“name”: <report name>,

“description”: <report description>,

“xmlContent”: <XML structure of report>,

“isDeleted”: <whether deleted or not>}

Gets the full detail of a report to allow amending or viewing it.

## /api/report/saveReport

Prerequisites: authenticated password

Method: POST

Parameters:

{“uuid”: <report UUID>,

“name”: <report name>,

“description”: <report description>,

“xmlContent”: <XML structure of report>,

“isDeleted”: <whether deleted or not>,

“folderUuid”: <folder to save/move to>}

Response:

{“uuid”: <report UUID>}

Creates or updates a report. Specifying a UUID will update the report, and omitting it will create a new one. In both cases, the report UUID is returned to the client.

When creating a report, the Name and XmlContent are the minimum requirements.

When updating a report, the UUID is the only mandatory element. Only elements that have been changed need to be sent (e.g. if renaming, just send the UUID and the new name).

## /api/report/deleteReport

Prerequisites: authenticated password

Method: POST

Parameters:

{“uuid”: <report UUID>}

Response:

none

Deletes a report for the given UUID

# /api/query

Domain for endpoints relating to queries

## /api/query/getQuery

Prerequisites: authenticated password

Method: GET

Parameters:

?uuid=<folder UUID>

Response:

{“uuid”: <query UUID>,

“name”: <query name>,

“description”: <query description>,

“xmlContent”: <XML structure of query>,

“isDeleted”: <whether deleted or not>}

Gets the full detail of a query to allow amending or viewing it.

## /api/query/saveQuery

Prerequisites: authenticated password

Method: POST

Parameters:

{“uuid”: <query UUID>,

“name”: <query name>,

“description”: <query description>,

“xmlContent”: <XML structure of query>,

“isDeleted”: <whether deleted or not>,

“folderUuid”: <folder to save/move to>}

Response:

{“uuid”: <query UUID>}

Creates or updates a report. Specifying a UUID will update the query, and omitting it will create a new one. In both cases, the query UUID is returned to the client.

When creating a query, the Name and XmlContent are the minimum requirements.

When updating a query, the UUID is the only mandatory element. Only elements that have been changed need to be sent (e.g. if renaming, just send the UUID and the new name).

## /api/query/deleteQuery

Prerequisites: authenticated password

Method: POST

Parameters:

{“uuid”: <query UUID>}

Response:

none

Deletes a query for the given UUID

# /api/listOutput

Domain for endpoints relating to listOutputs.

## /api/listOutput/getListOutput

Prerequisites: authenticated password

Method: GET

Parameters:

?uuid=<list output UUID>

Response:

{“uuid”: <output UUID>,

“name”: <output name>,

“description”: <output description>,

“xmlContent”: <XML structure of output>,

“isDeleted”: <whether deleted or not>}

Gets the full detail of a list output to allow amending or viewing it.

## /api/listOutput/saveListOutput

Prerequisites: authenticated password

Method: POST

Parameters:

{“uuid”: <output UUID>,

“name”: <output name>,

“description”: <output description>,

“xmlContent”: <XML structure of output>,

“isDeleted”: <whether deleted or not>,

“folderUuid”: <folder to save/move to>}

Response:

{“uuid”: <list output UUID>}

Creates or updates a list output. Specifying a UUID will update a list output, and omitting it will create a new one. In both cases, the list output UUID is returned to the client.

When creating an output, the Name and XmlContent are the minimum requirements.

When updating an output, the UUID is the only mandatory element. Only elements that have been changed need to be sent (e.g. if renaming, just send the UUID and the new name).

## /api/listOutput/deleteListOutput

Prerequisites: authenticated password

Method: POST

Parameters:

{“ uuid”: <listOutput UUID>}

Response:

none

Deletes a listOutput for the given UUID