# Endeavour Enterprise REST Specification

## /api/security/login

Prerequisites: none

Method: POST

Parameters: {“username”: <email address>, “password”, <password>}

Response:

Cookie, plus the below JSON

{“organisations”: [

{“organisationUuid”: <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: {“organisationUuid”: <organisation to select>}

Response: JSON of the OrganisationPersonLink object for that person and organisation, giving the permissions they have there

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: none

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/getUsers

Method: GET

Prerequisites: authenticated password and an admin user

Parameters: none

Response:

{“users”, [

{“username”: <email address>, “title”: <title>, “forename”: <forename>, “surname”: <surname>, “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/createOrganisation

Method: POST

Prerequisites: authenticated password and a superUser account

Parameters: {“name”: <organisation name>, “nationalId”: <national ID>}

Response:

Superuser-only Function to create a new organisation. Validates that an organisation with the same name and ID doesn’t already exist.

## /api/admin/createUser

Method: POST

Prerequisites: authenticated password and admin permission

Parameters: {“username”: <email address>, “title”: <title>, “forename”: <forename>, “surname”: <surname>, “permissions”: <1=user, 2=admin>, “isSuperUser”: null}

Response: none

Call to create a new person, passing up name and email and desired permission. If optional superUser element is present, it will also validate that the signed on person is a superUser.

Response is a defined constant:

1. New person created and invitation email sent
2. Person already exists (i.e. email already in use) at another organisation, so email sent but without invitation, since the person already has a password set up and just needs to be told about their new access privileges.
3. Person already exists at this organisation, so cannot add new person

## /api/admin/deleteUser

Prerequisites: authenticated password and admin permissions

Parameters: {“username”: <email address>}

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: {“username”: <email>}

Response: none

Call to resend the invite email for the given Person.

## api/folder/createFolder

Prerequisites: authenticated password

Method: POST

Parameters: {"folderName": <new folder name>, “parentFolderUuid”: <optional parent folder uuid>}

Response: none

Create a new folder for queries to go in. If a folder for the same name and at the same level already exists, an error response will be raised. A parentFolderUuid can be specified to make the new folder a child of an existing one. Without this, the new folder will be created as a top-level folder.

## api/folder/renameFolder

Prerequisites: authenticated password

Method: POST

Parameters: {“folderUuid”: <UUID of folder>, “folderName”: <new folder name>}

Response: none

Renames a folder. If a duplicate folder at the same level is found, an error is raised.

## api/folder/deleteFolder

Prerequisites: authenticated password

Method: POST

Parameters: {“folderUuid”: <UUID of folder>}

Response: none

Deletes a folder and all items and in it and folders below it.

## api/folder/moveFolder

Prerequisites: authenticated password

Method: POST

Parameters: {“folderUuid”: <UUID of folder>, “parentFolderUuid”: <UUID of new parent>}

Response: none

Moves a folder. If the move would result in a duplicate, an error is raised. To move a folder to be a top-level one, omit the parentFolderUuid element.

## api/folder/getFolders

Prerequisites: authenticated password

Method: GET

Parameters: none

Response: ….JSON of all folders with count of items in each one

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