SOP Organizer 1.0.0 OAS3

auth

POST /auth/signin Signs in a user

Parameters Try it out

Name	Description
Email * required string *(formData)*	Email of user to sign in Email
Password * required string *(formData)*	Password of user to sign in Password

Responses

Code	Description	Links
200	Returns a user object.	*No links*

POST /auth/signout Signs a user out

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Returns 'You've been signed out!'.	*No links*

POST /auth/preregister Pre-registers a user

Parameters Try it out

Name	Description
email * required string *(formData)*	Email of user to pre-register email

Responses

Code	Description	Links
200	Returns 'record inserted'.	*No links*
409	Returns 'User already registered'.	*No links*

POST /auth/register Registers a user

Parameters Try it out

Name	Description
Name * required string *(formData)*	Name of user to register Name
Email * required string *(formData)*	Email of user to register Email
Password * required string *(formData)*	Password of user to register Password

Responses

Code	Description	Links
200	Returns 'record inserted'.	*No links*
403	Returns 'User already registered.'.	*No links*
409	Returns 'User not pre-registered by admin.'.	*No links*

POST `/auth/perms` Updates the permission of a user

Parameters

Try it out

Name	Description
(Email)	

Responses

Code	Description	Links
200	Returns 'user privilege updated.'.	*No links*
409	Returns 'User does not exist with that email.'	*No links*

DELETE `/auth/removeFile` Deletes a file

Parameters

Try it out

Name	Description
id * required integer *(formData)*	ID of file to delete id

Responses

Code	Description	Links
200	Returns 'Deleted Row(s)'.	*No links*
500	Returns an error.	*No links*

directory

GET `/directory` Gets all directories

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Returns an array of all directory objects.	*No links*

POST `/directory` Creates a single directory

Parameters Try it out

Name	Description
directory object with name as a key and the name of the directory as a string * required `JSON` *(formData)*	JSON for a directory directory object with name as a key and the r

Responses

Code	Description	Links
200	Returns an array of directory objects.	*No links*

GET `/directory/:id` Gets a single directory

Parameters Try it out

Name	Description
id * required `integer` *(path)*	ID of the directory to get id

Responses

Code	Description	Links
200	Returns the directory with the specified id.	*No links*

DELETE `/directory/:id` Deletes a directory

Parameters Try it out

Name	Description
id * required `integer` *(path)*	ID of the directory to delete id

Responses

Code	Description	Links
200	Returns the directory deleted.	*No links*

GET **/directory/getSops/:id** Gets the SOPs in a directory

Parameters

Try it out

Name	Description
id * required `integer` *(path)*	ID of the directory to get id

Responses

Code	Description	Links
200	Returns an array of SOP objects.	*No links*

documents

GET **/documents** Gets all documents

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Returns an array of all document objects.	*No links*

GET **/documents/:id** Gets a single document

Parameters

Try it out

Name	Description
id * required `integer` *(path)*	ID of the document to get id

Responses

Code	Description	Links
200	Returns the document with the specified id.	*No links*

DELETE **/documents/:id** Deletes a document

Parameters

Try it out

Name	Description
id * required `integer` *(path)*	ID of the document to delete id

Responses

Code	Description	Links
200	Returns the document deleted.	*No links*

PATCH **`/documents/mark-delete/:id`** Marks a single document for deletion

Parameters

Try it out

Name	Description

id * required
`integer`
`(path)`
	ID of the document to mark for deletion
	id

Responses

Code	Description	Links
200	Returns the document to be marked for deletion.	*No links*

POST **`/documents/uploadNew`** Upload a new document; API not supported

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Returns the document to uploaded.	*No links*

POST **`/documents/updateExisting`** Updates an existing document; API not supported

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Returns the updated document.	*No links*

POST **`/documents/save`** Save changes to a document; API not supported

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Returns the saved document.	*No links*

POST **`/documents/download`** Downloads a document; API not supported

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Returns the downloaded document.	*No links*

POST `/documents/afterDownload` Deletes the converted .docx file; API not supported

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Returns "	*No links*

sops

GET `/sops` Gets all SOPs

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Returns an array of all SOP objects.	*No links*

POST `/sops` Creates an SOP; API not supported

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Returns the SOP created.	*No links*

DELETE `/sops` Deletes a SOP

Parameters Try it out

Name	Description
SOP id * required `integer` *(formData)*	ID of the sop to delete SOP id
Directory name * required `string` *(formData)*	Directory of the sop to delete Directory name

Responses

Code	Description	Links
200	Returns the sop deleted.	*No links*

GET **/sops/:id** Gets a single SOP

Parameters Try it out

Name	Description
id * required integer *(path)*	ID of the sop to get id

Responses

Code	Description	Links
200	Returns the sop with the specified id.	*No links*

PATCH **/sops/:id** Patches an SOP

Parameters Try it out

Name	Description
id * required string *(path)*	ID of the SOP to patch id

Responses

Code	Description	Links
200	Returns the patched SOP.	*No links*

GET **/sops/search/:searchTerm** Searches for SOPs by content

Parameters Try it out

Name	Description
searchTerm * required string *(path)*	Term that SOPs should contain searchTerm

Responses

Code	Description	Links
200	Returns the sop containing the specified content.	*No links*

GET **/sops/changeDirectory** Gets a single SOP

Parameters Try it out

Name	Description
sop_id * required integer *(formData)*	ID of the sop to change directories

Name	Description
	sop_id
old_directory_id * required integer *(formData)*	Old ID of the directory old_directory_id
new_directory_id * required integer *(formData)*	New ID of the directory new_directory_id

Responses

Code	Description	Links
200	Returns the sop with the specified id.	*No links*

users

GET `/users` Gets all users

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Returns an array of all user objects.	*No links*

POST `/users` Creates an user; API not supported

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Returns the user created.	*No links*

DELETE `/users` Deletes a user

Parameters Try it out

Name	Description
SOP id * required integer *(formData)*	ID of the user to delete SOP id

Responses

Code	Description	Links
200	Returns the user deleted.	*No links*

GET `/users/:id` Gets a users sop

Parameters

Try it out

Name	Description
id * required integer *(path)*	ID of the users to get id

Responses

Code	Description	Links
200	Returns the users with the specified id.	*No links*

PATCH `/users/:id` Patches a user

Parameters

Try it out

Name	Description
id * required string *(path)*	ID of the user to patch id

Responses

Code	Description	Links
200	Returns the patched user.	*No links*