AGORA API – Short description

My typical suggestion to a customer trying to integrate our API is to perform the same operations on our web front end (which also uses the API) and examine the requests and responses.

First of all: most of the time, when you create a resource (room, file, ...) in AGORA you simply want to inherit the settings defined in the parent "container". For this reason, a proper setup of the Trust Room is the key to simplifying all the successive tasks.
Regarding the properties: The most important are:

rid	the resource ID (generated automatically by the system)
pid	the parent resource ID
cid	the trust room ID
type	the type of resource (1 = trust room, 2 = user, 3 = group, 4 = room, 5 = file, 6 = Discussion, 7 = Event, ...)
subtype	the subtype of the resource
permissions	who can do what with the resource
notifications	if you want to send a notification for this resource
approval	if you want that a file or the files inside a room should be approved before publication

Walkthrough

Login

At the beginning you always need to login in order to retrieve the access token to use for all the successive requests:

POST https://collaboration.agora-secureware.ch/agora-api/api/v1/auth

Request headers

Content-Type: application/x-www-form-urlencoded
Accept: application/json, text/plain, */*

Form-data

grant_type=password&client_id=chc&client_secret=secret&username=**[YOUR_USERNAME]&**password=**[YOUR_PASSWORD]**&community=**[YOUR_TRUST_ROOM_NAME]**

If successful (200), this request returns a JSON object containing the user information (settings), the data of the last session (sessionInformation), and the access token (access_token) to be used. You will have to save this last value.

Create user account

Next, we will create a user account for a new customer.

Here we should discuss the properties you would like to set for each customer, particularly the type of authentication (only password, 2-factor authentication with SMS code or OTP), whether you want AGORA to send them the password or prefer your API to do it.

To create a new user account, you have to send a POST request to /users:

POST https://collaboration.agora-secureware.ch/agora-api/api/v1/users

Request Headers

Accept: application/json, text/plain, */*
Authorization: Bearer [YOUR_ACCESS_TOKEN]
Content-Type: application/json;charset=UTF-8

Body

JSON Object	Description

{
"username": "mickey.mouse@disney.com",
"lastName": "Mouse",
"firstName": "Mickey",
"email": "mickey.mouse@disney.com",
"inheritSettings": true,
 	
"type": 2,
"subtype": 3,
"createPersonalRoom": true,
 	
"authenticationMethod": 0,
 	
"loginInfo": false,
"password": "VeryComplexPas$w0rdGeneratedByYou"
}
	The username to use for the log-in
	Last name
	First name
	The email address
	With this parameter you avoid specifying the permissions
	Type 2 is User
	Subtype 3 is Guest (2 is member)
	With this parameter you automatically create a room for this new user
	“Only password” authentication method (other values are 2 for SMS challenge and 7 for OTP)
	Do not send the login information (if you want to send them from your app)
	The password you have generated (if not provided AGORA will generate one automatically, but in this case, you will have to leave to AGORA the responsibility to send it to the user)

In the response of the creation call you will receive the properties of the created account, among them you will find the resource ID (rid) or the user.
You should store on your end this value for further use. If you decided to also create a room for this new user, the ID of that room is also available inside the response (roomId). You probably should also store this value for later use.

By default, the room is created with following permissions: the owners of the User will be owner also of the room. In addition, the user will have a contributor permission on it (this mean he will be able to save new information (files, …) into it.
Also, by default, notifications are activated on this room: this means that if someone for example uploads a file inside it, the other allowed users will be notified by email/SMS.

Retrieve the content of a room

To retrieve the content of a room, send the following request

GET **https://collaboration.agora-secureware.ch/agora-api/api/v1/rooms/[ID_OF_THE_ROOM]/content**

Upload a file

Let’s us now upload a file inside this room:

POST https://collaboration.agora-secureware.ch/agora-api/api/v1/rooms/[ID_OF_THE_ROOM]/files

Request headers

Accept: application/json, text/plain, */*
Authorization: Bearer [YOUR_ACCESS_TOKEN]
Content-Type: multipart/form-data; boundary=----[YOUR_BOUNDARY]

Body

```
------[YOUR_BOUNDARY]
Content-Disposition: form-data; name="Resource" {"type":5,"subtype":2,"inheritSettings":true,"name":"example.txt"}
------[YOUR_BOUNDARY]
Content-Disposition: form-data; name="file"; filename="example.txt" Content-Type: text/plain

[BYTES OF THE FILE]

------[YOUR_BOUNDARY]--
```

As you can see, you will have to send a multipart request with two parts: the JSON meta-data and the file itself. The JSON object of the first part can be quite simple: near the type (2=file) and subtype (2=file) you only have to specify the name of the file (which could be different form the actual filename) and set the property “inheritSettings”:true in order to simply inherit all the settings (permissions, notifications, approvals, …) from the room.

In the response of this request you will receive a JSON object with the properties of the created file. Among them you have the generated resource ID (rid).

Delete a file

By default, Trust Rooms have activated the “Trash bin”, so that usually you do not delete directly a file, but you simply move it to the trash bin.

For doing it simply send this request

POST https://collaboration.agora-secureware.ch/agora-api/api/v1/trashbin/[RID_OF_THE_FILE]/move

You can however also directly delete definitely the file by sending

DELETE https://collaboration.agora-secureware.ch/agora-api/api/v1/files/[RID_OF_THE_FILE]

Delete the room

Similarly, you can use similar methods for deleting the room of the user:

POST https://collaboration.agora-secureware.ch/agora-api/api/v1/trashbin/[RID_OF_THE_ROOM]/move

You can however also directly delete definitely the file by sending

DELETE https://collaboration.agora-secureware.ch/agora-api/api/v1/rooms/[RID_OF_THE_ROOM]

(This way also all the content of the room will be trashed / deleted)

Delete a user

Usually this operation is not necessary since you simply can delete the user by sending

DELETE https://collaboration.agora-secureware.ch/agora-api/api/v1/users/[RID_OF_THE_USER]

Here you can pass as optional query string parameter "?**room=true**" in order to also delete the personal room of the user with its content. If you don't want to delete it simply do not pass this parameter.

POST https://collaboration.agora-secureware.ch/agora-api/api/v1/users/1700000000119459?room=true

Disable / Enable a user

In some case instead of deleting a user, you may prefer to simply disable it.
For doing it – like for any update operation in our API – you will need to send the updateDate of the object (this in order to avoid concurrency problems). For that simply make a GET request to the user

GET https://collaboration.agora-secureware.ch/agora-api/api/v1/users/[RID_OF_THE_USER]

And store the "updateDate".

Then you can disable the user by sending

PUT https://collaboration.agora-secureware.ch/agora-api/api/v1/users/[RID_OF_THE_USER]/enable?enable=false&updateDate=[UPDATE_DATE]

If you then want to reactivate the user, you can send

PUT https://collaboration.agora-secureware.ch/agora-api/api/v1/users/[RID_OF_THE_USER]/enable?enable=true&updateDate=[UPDATE_DATE]

AGORA API Endpoints

Authentication

URL	Method	Description	Parameters	Response

/api/v1/auth	POST	Login	username, password, community	{
				"access_token":"acbf5706-38f4-4360-ba3c-d25d4e01ef3e",
				"token_type":"bearer",
				"refresh_token":"98d5d68f-93a1-422b-95c9-d1ef874c17b6",
				"expires_in":1799,
				"scope":"read",
				"authMethod":0,
				"settings": {
				"userId":10,
				"shortName":"UP",
				"rootUsersPermission":5,
				"rootUsersId":9,
				"rootRoomsPermission":5,
				"rootRoomsId":49,
				"rootGroupsPermission":5,
				"rootGroupsId":29,
				"communityPermission":5,
				"communityId":1,
				"communityName":"Test Community",
				"language":"en_US",
				"displayName":"User Patrick",
				"lastLogin":1398938400000
				}
				}

/api/v1/logout	GET	Logout	access_token	-

System

URL	Method	Description	Parameters	Example	Response Example

/api/v1/system/settings	GET	Retrieves system settings (languages, default language)			{
					"LANGUAGES":[
					"de_CH",
					"en_US",
					"fr_CH",
					"it_CH"
					],
					"DEFAULT_LANGUAGE":"en_US"
					}

/api/v1/system/texts	GET	Retrieves the texts for the given language	lang	lang=[en]	{
					"USER_SEND_LOGIN_INFO_TITLE_ONLY_PASSORD":"AGORA – Password",
					....
					"RESOURCE_VALIDATION_INVALID_APPROVAL":"Not valid approval",
					"APP_LOCALE_EN":"English",
					"approvalRequest.file":"Request approval for this document",
					"ACTIONS_CLOSE":"Close",
					"APPROVALS_ABSTAIN":"You cannot vote abstain.",
					"notification.sms.fileInsert.body.nohtml":"New file in the community "6 "",
					"NOTIFICATIONS_SETTINGS_PRIORITY_NORMAL":"Normal priority"
					}

/api/v1/system/formats	GET	Retrieves the formats for the given language	lang	lang=[en]	{
					"GROUPING_SEPARATOR":",",
					"DATE":"M/d/yyyy",
					"DECIMAL_SEPARATOR":"."
					}

/api/v1/system/countries	GET	Retrieves the countries for the given language	lang	lang=[en_US]	[{ "code":"af", "name":"Afghanistan", "default":false }, { "code":"al", "name":"Albania", "default":false }, { "code":"dz", "name":"Algeria", "default":false }, { "code":"ye", "name":"Yemen", "default":false }, { "code":"zm", "name":"Zambia", "default":false, }, { "code":"zw", "name":"Zimbabwe", "default":false

					}]

/api/v1/system/passwordcomplexity	POST	Checks the complexity of the given password	lang, password, minLength	lang=[en_US],
				password=
				[Patrick123],
				minLength=[8]
					{
					"score":73,
					"complexity":2
					"problems":
					[
					"The password entered contains consecutive lower-case characters",
					"The password entered contains consecutive digits",
					"The password entered contains sequential digits"
					]
					}

/api/v1/system/smslogin	POST	Performs the SMS login	challenge	TODO	TODO

Current user

URL	Method	Description	Parameters	Example	Response

/api/v1/user/settings	GET	Retrieve the settings of the current user			{
					"userId":10,
					"shortName":"UP",
					"rootUsersPermission":5,
					"rootUsersId":9,
					"rootRoomsPermission":5,
					"rootRoomsId":49,
					"rootGroupsPermission":5,
					"rootGroupsId":29,
					"communityPermission":5,
					"communityId":1,
					"communityName":"Test Community",
					"language":"en_US",
					"displayName":"User Patrick",
					"lastLogin":1398938400000
					}

/api/v1/user/changepassword	POST	Change the password of the current user	{oldPassword, newPassword, updateDate}	TODO	1400234577557

Communities

URL	Method	Description	Parameters	Example	Response
/api/v1/communities/new	POST	Creates a new community	communityName, lastName, firstName, email, password	communityName=[Community Name], lastName=[community], firstName=[manager], email=[comm.manager@community.com], password=[password]	-

/api/v1/community	GET	Retrieves the current community properties			{
					"rid":1,
					"pid":0,
					"cid":1,
					"subtype":1,
					"status":0,
					"authorId":10,
					"authorName":"User Patrick",
					"authorShortName":"UP",
					"updateDate":1400234239279,
					"createDate":1400234239279,
					"visibleFrom":1400234239279,
					"expireDate":2524604400000,
					"name":"Test Community",
					"notification":0,
					"notificationPriorityHigh":false,
					"approval":0,
					"approvalNotification":0,
					"approvalMustEverybodyVote":false,
					
					"permissions":
					{
					"items":
					[
					{
					"rid":30,
					
					"displayName":"Community Managers",
					
					"shortName":"CM",
					"type":3,
					"level":5,
					"inheritance":0
					}
					]
					},
					"permission":5,
					"loginName":"test",
					"defaultLanguage":"en_US",
					"communityGuestTypes":
					[
					{
					"type":3,
					"enabled":false
					},
					{
					"type":4,

					"enabled":false
					}
					],
					"communityGuestCreation":0,
					"passwordMinLength":8,
					"passwordMinQuality":0,
					"passwordChangeType":0,
					"passwordPreventReuse":0,
					"passwordMaxLoginTries":0,
					"communityNotificationChannels":
					[
					{ <<< "type":1,
					"enabled":true <<<
					},
					{
					"type":2,
					"enabled":true
					},
					],
					"defaultAuthenticationMethod":1,
					"generateOwnGridCard":false,
					"spaceTreshold":0,
					"communityAdminProperties":
					{
					"authenticationMethods":
					[
					{
					"type":0,
					"enabled":true
					},
					{
					"type":2,
					"enabled":false
					},
					{
					"type":1,
					"enabled":false
					}
					],
					"diskSpace":1024,
					"functions":0,
					"sendLoginInfoChannels":
					[
					{
					"type":1,
					"enabled":true
					},
					{
					"type":2,
					"enabled":false
					},
					{
					"type":4,
					"enabled":false
					}
					],
					"notificationChannels":
					[
					{
					"type":1,
					"enabled":true
					},
					{
					"type":2,
					"enabled":true
					}
					],
					"users":10,
					"resourceId":1,
					"sendLoginInfo":true,
					"printLoginInfo":false,
					"emailMandatory":false,
					"mobilePhoneMandatory":false
					},
					"type":1
					}

/api/v1/community	PUT	Updates the current community properties	{community JSON}	Community[communityNotificationChannels=[NotificationChannel[type=1, enabled=true], NotificationChannel[type=2, enabled=false]], updateDate=Fri May 16 11:57:19 CEST 2014, permission=0,]	1418804141761
/api/v1/community/notificationsettings	GET	Returns the notification settings of the current community			{ "rid":1,"communityNotificationChannels": [{ "type":1,"enabled":true }, { "type":2,"enabled":true }], "normalPriority": { "channels": [{ "type":1,"enabled":true }, { "type":2,"enabled":false }], "delivery":1,"days":0 } ,"highPriority": { "channels": [{ "type":1,"enabled":true }, { "type":2,"enabled":false }],"delivery":1,"days":0 } ,"editable":false,"defaultSettings":true }
/api/v1/community/notificationsettings	PUT	Returns the notification settings of the current community	{notification settings JSON}	{ "communityNotificationChannels": [{ "type":1,"enabled":true }, { "type":2,"enabled":true }] ,"normalPriority": { "channels": [{ "type":1,"enabled":false }, { "type":2,"enabled":true }] ,"delivery":1,"days":0 } ,"highPriority": { "channels": [{ "type":1,"enabled":true }, { "type":2,"enabled":true }]	updateDate=[1418820555591]

				,"delivery":1,"days":0 }
				,"editable":true,"defaultSettings":true }	

Rooms

URL	Method	Description	Parameters	Example	Re
/api/v1/rooms	GET	Retrieves the list of rooms under the root			[{ Pat { } Pat des 1",
/api/v1/rooms	POST	Creates a new room under the root	{Room JSON}	Room[rid=50,pid=49,cid=1,subtype=4,authorId=10,name=NewName,permissions=Permissions[items=[Permission[level=5,inheritance=0,rid=10,type=2], Permission[level=5,inheritance=0,rid=30,type=3]]],permission=0]	
/api/v1/rooms/properties	GET	Get the properties of the rooms root object			{ " des Co
/api/v1/rooms/properties	PUT/POST	Modifies the properties of the rooms root object	{Room JSON}	Room[rid=49,subtype=4,updateDate=Fri May 16 12:08:51 CEST 2014,permission=0]	
/api/v1/rooms/config	GET	Returns the configuration needed for creating a new object under the rooms root object			
/api/v1/rooms/{id}	GET	Retrieves the properties of the room with the given id			{ " des Co ,"p
/api/v1/rooms/{id}	PUT/POST	Modifies the properties of the room with the given id	{Room JSON}	Room[rid=50,pid=49,cid=1,subtype=4,authorId=10,updateDate=Fri May 16 12:08:51 CEST 2014,name=NewName,permissions=Permissions[items=[Permission[level=5,rid=10,type=2], Permission[level=5,rid=30,type=3], Permission[level=3,rid=1,type=1]]],permission=0]	
/api/v1/rooms/{id}	DELETE	Deletes the given room			
/api/v1/rooms/{id}/config	GET	Returns the configuration needed for creating a new object under the given room			
/api/v1/rooms/{id}/rooms	GET	Retrieves the list of rooms under the room with the given id			{ " 2", }, { 3", }]
/api/v1/rooms/{id}/rooms	POST	Creates a new room under the room with the given id	{Room JSON}	Room[rid=50,pid=50,cid=1,subtype=4,authorId=10,name=NewName,permissions=Permissions[items=[Permission[level=5,rid=10,type=2]]],permission=0]	
/api/v1/tree	GET	Returns the community rooms tree	write=[true]		
/api/v1/rooms/{id}/tree	GET	Returns the rooms tree below the given room	write=[true]		{ " 2", "ric
/api/v1/rooms/{id}/copy	POST	Copies the room with the given id.	{Room JSON}		

Files

URL	Method	Description	Parameters	Example
/api/v1/{resourceType}/{pid}/files	GET	Retrieves the list of files under the object (room/user/group) with the given pid		
/api/v1/{resourceType}/{pid}/files	POST	Creates a new file under the object (room/user/group) with the given pid	{File JSON}	File[maxVersions=2,pid=50,cid=1,subtype=2,description=Resource description,permissions=Permissions[items=[Permission[level=5,rid=10,type=2]]],perm
/api/v1/files/{id}	GET	Retrieves the properties of the files with the given id and the last version		

/api/v1/files/{id}	PUT	Make an update on the file resource with the given id	File Json	File[maxVersions=2,secView=false,rid=1418893180330001,pid=50,cid=1,subtype=2,up Dec 18 09:59:40 CET 2014,name=ChangedName.txt,description=Resource description,p
/api/v1/files/{id}	DELETE	Delete the file with the given id		
/api/v1/files/{id}/versions/{vid}	DELETE	Delete the version vid of the file with the given id		
/api/v1/files/{id}/versions	GET	Get the list of the version of this file with the given id		
/api/v1/files/{id}/versions	POST	Upload a new version of the given file with given id	{Version JSON}	Version[versionDescription=version 2 description,resourceId=1418894671354001]
/api/v1/files/{id}/download	GET	Download of the file with the given id		
/api/v1/files/{id}/versions/{vid}/download	GET	Download the version vid of the file with the given id		
/api/v1/files/{id}/secprint	GET	Download the secprint of the file with the given id		
/api/v1/files/{id}/versions/{vid}/secprint	GET	Download the secprint of the version vid of the file with the given id		
/api/v1/files/{id}/secview	GET	Download the secview (JSON) of the file with the given id		
/api/v1/files/{id}/versions/{vid}/secview	GET	Download the secview (JSON) of the version vid of the file with the given id		
/api/v1/files/{id}/image	GET	Download the image (for secview) of the file with the given id	page, maxHeight, maxWidth, base64	page=[1], maxHeight=[300], maxWidth=[200]
/api/v1/files/{id}/versions/{vid}/image	GET	Download the image (for secview) of the version vid of the file with the given id	page, maxHeight, maxWidth, base64	
/api/v1/files/zip	GET	Creates a zip file with the given resources ids	List of rid, base64	
/api/v1/files/{id}/copy	POST	Copies the file with the given id	{File JSON}	

Users

URL	Method	Description	Parameters	Example
/api/v1/users	GET	Retrieve the list of users		
/api/v1/users	POST	Creates a new user	{User JSON}	{ "permissions": { "items": [{ "rid":1,"type":1,"displayName":"All Community","level":3,"inher "rid":30,"type":3,"displayName":"Community Managers","level":5,"inheritance":0 }] }, "authenticationMethod":1,"username":"uname","firstName":"fname","lastName":"lname","emai }
/api/v1/users/config	GET	Returns the configuration needed for creating a new user		
/api/v1/users/properties	GET	Get the properties of the users root object		
/api/v1/users/properties	PUT/POST	Modifies the properties of the users root object	{User JSON}	
/api/v1/users/{id}	GET	Retrieves the properties of the user with the given id		

/api/v1/users/{id}	PUT/POST	Modifies the properties of the user with the given id	{User JSON}	User[lastName=Paperone,firstName=Zio,updateDate=Fri May 16 12:08:51 CEST 2014,permissi
/api/v1/users/{id}/config	GET	Returns the configuration needed for creating a new room for the given user		
/api/v1/users/{id}/rooms	POST	Activates the personal room for the user with the given id	{Room JSON}	
/api/v1/users/{id}/image	GET	Retrieves the image (base64) for the user with the given id		
/api/v1/users/{id}/notificationsettings	GET	Returns the notification settings for the given user		
/api/v1/users/{id}/notificationsettings	PUT	Updates the notification settings for the given user	{Notification settings JSON}	{ "rid":10,"communityNotificationChannels": [{ "type":1,"enabled":true }, { "type":2,"enabled "normalPriority": { "channels": [{ "type":1,"enabled":false }, { "type":2,"enabled":true }], "delivery":1,"days":0 }, "highPriority": { "channels": [{ "type":1,"enabled":true }, { "type":2," }], "delivery":1,"days":0 }, "editable":false,"defaultSettings":false }
/api/v1/users/{id}/enable	PUT	Enable/Disable the given user	enable=[true/false]	
/api/v1/users/{id}/logininfo	POST	Sends the login information to the given user	sendType	

Groups

URL	Method	Description	Parameters	Response
/api/v1/groups	GET	Retrieves the list of groups		
/api/v1/groups	POST	Creates a new group	{Group JSON}	
/api/v1/groups/config	GET	Returns the configuration needed for creating a new group		
/api/v1/groups/properties	GET	Get the properties of the groups root object		
/api/v1/groups/properties	PUT/POST	Modifies the properties of the groups root object	{Group JSON}	
/api/v1/groups/{id}	GET	Retrieves the properties of the group with the given id		
/api/v1/groups/{id}	PUT/POST	Modifies the properties of the group with the given id	{Group JSON}	
/api/v1/groups/{id}/config	GET	Returns the configuration needed for creating a new room for the given group		
/api/v1/groups/{id}/memberships	GET	Retrieves the memberships of the group with the given id		
/api/v1/groups/{id}/rooms	POST	Activates the personal room for the group with the given id	{Room JSON}	

Home

URL	Method	Description	Parameters	Example	Response
/api/v1/home	GET	Returns the elements in the home and it's first sub level			[{ "rid":55,"displayName":"LastUpdateRoom","authorId":10,"authorName":"User Patrick","createDate":1400234931508,"updateDate":1400234931508,"maxVersions":1,"permission":5,"ty 1","authorId":10,"authorName":"User Patrick","createDate":1400234931508,"updateDate":14002349315 description","permission":5,"type":4,"sortType":0,"children": [{ "rid":51,"displayName":"Room 2","autho Patrick","createDate":1400234931508,"updateDate":1400234931508,"maxVersions":1,"permission":5,"ty 3","authorId":10,"authorName":"User Patrick","createDate":1400234931508,"updateDate":14002349315 "rid":100,"displayName":"document.docx","authorId":10,"authorName":"User Patrick","createDate":1400234330463,"updateDate":1400234330463,"maxVersions":5,"description":"Res description","permission":5,"type":5,"sortType":0,"format":"docx","size":1024,"secView":true }, { "rid":103,"displayName":"powerpoint.pptx","authorId":10,"authorName":"User Patrick","createDate":1400234330463,"updateDate":1400234330463,"maxVersions":1,"description":"pow file","permission":5,"type":5,"sortType":0,"format":"pptx","size":1024,"secView":true }, { "rid":104,"disp document","authorId":10,"authorName":"User Patrick","createDate":1400234330463,"updateDate":1400234330463,"maxVersions":0,"permission":5,"ty }, { "rid":101,"displayName":"text.txt","authorId":10,"authorName":"User Patrick","createDate":1400234330463,"updateDate":1400234931508,"maxVersions":5,"permission":5,"ty }, null] }, { "rid":1418811188544002,"displayName":"Pizzagalli Aldo","authorId":10,"authorName":"Use Patrick","createDate":1418811188544,"updateDate":1418811188544,"maxVersions":0,"permission":5,"ty "rid":1418811188704003,"displayName":"Pizzagallis_File.txt","authorId":10,"authorName":"User Patrick","createDate":1418811188704,"updateDate":1418811188704,"maxVersions":0,"description":"Res description","permission":5,"type":5,"sortType":0,"format":"txt","size":26,"secView":true }] }, { "rid":14 Pietro","authorId":10,"authorName":"User Patrick","createDate":1418811189431,"updateDate":1418811189431,"maxVersions":0,"permission":5,"ty "rid":1418811189572006,"displayName":"Saccos_File.txt","authorId":10,"authorName":"User Patrick","createDate":1418811189572,"updateDate":1418811189572,"maxVersions":0,"description":"Res description","permission":5,"type":5,"sortType":0,"format":"txt","size":21,"secView":true }] }]
/api/v1/lastUpdates	GET	Returns a list with the Resources that are updated or created since the	filterDate*, type*	filterDate=[1] type=[4]	[{ "rid":1418804746734003, "displayName":"Created_Room", "authorId":10, "authorName":"User Patrick "updateDate":1418631946782, "maxVersions":0, "permission":5, "type":4, "updateStatus":0, "sortType": "displayName":"Updated_File", "authorId":10, , "authorName":"User Patrick", "createDate":14136171466 "description":"Resource description", "permission":5, "type":5, "updateStatus":1, "sortType":0, "size":14, "displayName":"Created_File", "authorId":10, "authorName":"User Patrick", "createDate":141863194628 "description":"Resource description", "permission":5, "type":5, "updateStatus":0, "sortType":0, "size":14,

		reference date.			

*type is optional, default value = 0 (all)

*filterDate is optional, default value = lastLogin

History

URL	Method	Description	Parameters	Response
/api/v1/history	GET	Returns the latest activities inside the current community		
/api/v1/users/{id}/activity	GET	Returns the activity of the given user		
/api/v1/{resource type}/{id}/history	GET	Returns the history of the given resource		

Permissions

URL	Method	Description	Parameters	Response
/api/v1/permissions/selection/{pid}	GET	Returns the list of selectable users and groups for an object inside the space defined by the resource with the given pid		
/api/v1/permissions/overview	POST	Returns the list of the users with relative permissions given the "Permissions" object received as parameter	{Permissions JSON}	
/api/v1/permissions/overview/{id}	GET	Returns the list of the users with relative permissions on the resource with the given ID		

Event

|/api/v1/events/id/structure | GET | Returns the structure of the event with the given id | | | Missing example |

URL	Method	Description	Parameters	Example	Response
/api/v1/events	GET	Returns a list of events			Missing example
/api/v1/events/id	GET	Returns the properties of the given event			{"rid":1421318331642001, "pid":50, "cid":1, "subtype":2, "status":0, "authorId":10, "authorName":"User Patrick", "authorShortName":"UP", "updateDate":1421318331642, "createDate":1421318331642, "visibleFrom":1421318331642, "expireDate":253402297199000, "name":"Event name", "notification":0, "notificationPriorityHigh":false, "approval":0, "approvalNotification":0, "approvalMustEverybodyVote":false, "permissions": {"items": [{ "rid":10, "displayName":"User Patrick", "shortName":"UP", "type":2, "level":5, "inheritance":0 }] }, "breadcrumb": [{ "description":"Rooms Root description", "displayName":"Rooms Root", "rid":49, "shortName":"RR", "subtype":1, "type":4, "permission":5 }, { "description":"Room description", "displayName":"Room 1", "rid":50, "shortName":"R1", "subtype":4, "type":4, "permission":5 }], "permission":5, "sortType":0, "manualSortIndex":0, "maxVersions":20, "maxLifetime":0, "showInWelcome":true, "location":"Manno", "fromDate":1416259800000, "showMap":false, "toDate":1416267000000, "allDayEvent":false, "showAsFree":true, "reminder":1, "type":7, "displayName":"Event name", "shortName":"EN" }
/api/v1/events/id	PUT/POST	Update the properties of the event identified by the given id	Event json	Missing example	
/api/v1/events/id	DELETE	Delete the given event			
/api/v1/events/invitations	GET	Returns the list of invitations to be confirmed			Missing example
/api/v1/events/id/invitations	PUT/POST	Updates the invitation status at the event with the given id for the current user			
/api/v1/events/pid/config	GET	Returns the configuration needed for creating resource under the given event			Missing example

/api/v1/rooms/id/events	GET	Get a list with the Events in the room with the given id			Missing example
/api/v1/rooms/id/events	POST	Insert a new event under the one identified by the given id		Missing example	
/api/v1/events/id/tree	GET				{ "rid":120, "displayName":"apero", "type":7, "subtype":2, "permission":5, "children": [{ "rid":1421318576748001, "displayName":"NewName", "type":4, "subtype":6, "permission":5 }] }

Discussion

Comments

Community

Timeline

History

Activity

Other important things

updateDate deve essere mandato in fase di update per poter aggiornare una risorsa

Enums

- Type (ResourceType) * 1 = COMMUNITY * 2 = USER * 3 = GROUP * 4 = ROOM * 5 = FILE * 6 = DISCUSSION * 7 = EVENT * 8 = TASK * 9 = SURVEY
- Subtype * Community * 1 = Productive * 2 = Intern * 3 = Trial * Users * 1 = Root * 2 = Member * 3 = Timed guest * 4 = One-Time guest * Groups * 1 = Root * 2 = Group * Rooms * 1 = Root * 2 = User personal room * 3 = Group room * 4 = General * 5 = Context * 6 = Event folder * Files * 2 = File * 3 = Shortcut * 4 = Note * 5 = External link * Events * 2 = Event * Discussions * 2 = Discussion
- FilterDate for lastUpdate * 1 = 3 days * 2 = 7 days * 3 = 1 month * defaut = since the last login
- Period for event * 1 = 1 week * 2 = 2 weeks * 3 = 1 month * defaut = 1 week
- PermissionLevel * -1 = ACCESS_DENIED * 1 = SEC_VIEW * 2 = SEC_PRINT * 3 = ACCESS * 4 = CONTRIBUTE * 5 = MODIFY_AND_DELETE