

KYPO User and Group - API Reference

Table of Contents

1. Overview	1
1.1. Version information	1
1.2. Contact information	1
1.3. URI scheme	1
1.4. Tags	1
2. Resources	1
2.1. Groups	1
2.2. Roles	12
2.3. Users	18
3. Definitions	28
3.1. AddUsersToGroupDTO	28
3.2. GroupDTO	28
3.3. GroupDeletionResponseDTO	29
3.4. NewGroupDTO	29
3.5. RoleDTO	29
3.6. UpdateGroupDTO	30
3.7. UserDTO	30
3.8. UserDeletionResponseDTO	30
3.9. UserForGroupsDTO	31

1. Overview

KYPO User and Group API Reference Description.

1.1. Version information

Version : Version: 1.1.6

1.2. Contact information

Contact : Pavel Seda

Contact Email : 441048@mail.muni.cz

1.3. URI scheme

Host : localhost:8084

BasePath : /kypo2-user-and-group/api/v1

Schemes : HTTP, HTTPS

1.4. Tags

- groups
- roles
- users

2. Resources

2.1. Groups

2.1.1. Create new group.

POST /kypo2-user-and-group/api/v1/groups

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Group to be created.	NewGroupDTO

Responses

HTTP Code	Description	Schema
200	Given group created.	GroupDTO

Consumes

- `application/json`

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups
```

Request body

```
{
  "name" : "Main group",
  "description" : "Group for main users.",
  "expiration_date" : "2019-11-20T10:28:02.727Z",
  "users" : [ {
    "full_name" : "Peter Novak.",
    "given_name" : "Pavel",
    "family_name" : "Seda",
    "login" : "{ \"sub\": \"michaelsmith\", \"iss\": \"https://oidc.muni.cz/oidc/\" }",
    "mail" : "michaelsmith@mail.muni.cz.",
    "iss" : "https://oidc.muni.cz",
    "picture" : [ "string" ],
    "id" : 1
  } ],
  "group_ids_of_imported_users" : [ 0 ]
}
```

Example HTTP response

Response 200

```
{
  "name" : "Main group of organizers",
  "description" : "Organizers group for training run in June.",
  "roles" : [ {
    "id_of_microservice" : 5,
    "name_of_microservice" : "kypo2-training",
    "description" : "This role will allow you to create and delete groups.",
    "role_type" : "ROLE_USER_AND_GROUP_ADMINISTRATOR",
    "id" : 1
  } ],
  "users" : [ {
    "full_name" : "Peter Novak.",
    "given_name" : "Pavel",
    "family_name" : "Seda",
    "login" : "{\"sub\":\"michaelsmith\", \"iss\":\"https://oidc.muni.cz/oidc/\"}",
    "mail" : "michaelsmith@mail.muni.cz.",
    "iss" : "https://oidc.muni.cz",
    "picture" : [ "string" ],
    "id" : 1
  } ],
  "source" : "Internal",
  "can_be_deleted" : false,
  "expiration_date" : "2017-10-19 10:23:54+02",
  "id" : 1
}
```

2.1.2. Get groups.

GET /kypo2-user-and-group/api/v1/groups

Parameters

Type	Name	Description	Schema
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)
Body	body <i>optional</i>	Parameters for filtering the objects.	< string, < string > array > map

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups
```

Request query

```
{
  "fields" : "string",
  "page" : 0,
  "size" : 0,
  "sort" : "asc"
}
```

Request body

```
{ }
```

Example HTTP response

Response 200

```
"object"
```

2.1.3. Updates input group.

```
PUT /kypo2-user-and-group/api/v1/groups
```

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Group to be updated.	UpdateGroupDTO

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups
```

Request body

```
{
  "name" : "Main group.",
  "description" : "Group for main users.",
  "expiration_date" : "2019-11-20T10:28:02.727Z",
  "id" : 1
}
```

2.1.4. Tries to delete groups with given ids and returns groups and statuses of their deletion.

```
DELETE /kypo2-user-and-group/api/v1/groups
```

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Ids of groups to be deleted.	< integer(int64) > array

Responses

HTTP Code	Description	Schema
200	successful operation	< GroupDeletionResponseDTO > array

Consumes

- `application/json`

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups
```

Request body

```
[ 0 ]
```

Example HTTP response

Response 200

```
"array"
```

2.1.5. Assign role with given role ID to group with given ID

```
PUT /kypo2-user-and-group/api/v1/groups/{groupId}/roles/{roleId}
```

Parameters

Type	Name	Description	Schema
Path	groupId <i>required</i>	groupId	integer(int64)
Path	roleId <i>required</i>	roleId	integer(int64)

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups/0/roles/0
```

2.1.6. Cancel role with given role ID to group with given ID

```
DELETE /kypo2-user-and-group/api/v1/groups/{groupId}/roles/{roleId}
```

Parameters

Type	Name	Description	Schema
Path	groupId <i>required</i>	groupId	integer(int64)
Path	roleId <i>required</i>	roleId	integer(int64)

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups/0/roles/0
```

2.1.7. Get group with given id

```
GET /kypo2-user-and-group/api/v1/groups/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Id of group to be returned.	integer(int64)

Responses

HTTP Code	Description	Schema
200	successful operation	GroupDTO

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups/0
```

Example HTTP response

Response 200

```
{
  "name" : "Main group of organizers",
  "description" : "Organizers group for training run in June.",
  "roles" : [ {
    "id_of_microservice" : 5,
    "name_of_microservice" : "kypo2-training",
    "description" : "This role will allow you to create and delete groups.",
    "role_type" : "ROLE_USER_AND_GROUP_ADMINISTRATOR",
    "id" : 1
  } ],
  "users" : [ {
    "full_name" : "Peter Novak.",
    "given_name" : "Pavel",
    "family_name" : "Seda",
    "login" : "{\"sub\":\"michaelsmith\", \"iss\":\"https://oidc.muni.cz/oidc/\"}",
    "mail" : "michaelsmith@mail.muni.cz.",
    "iss" : "https://oidc.muni.cz",
    "picture" : [ "string" ],
    "id" : 1
  } ],
  "source" : "Internal",
  "can_be_deleted" : false,
  "expiration_date" : "2017-10-19 10:23:54+02",
  "id" : 1
}
```

2.1.8. Tries to delete group with given id and returns if it was successful.

```
DELETE /kypo2-user-and-group/api/v1/groups/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Id of group to be deleted.	integer(int64)

Responses

HTTP Code	Description	Schema
200	successful operation	GroupDeletionResponseDTO

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups/0
```

Example HTTP response

Response 200

```
{
  "status" : "SUCCESS",
  "id" : 1
}
```

2.1.9. Returns all roles of group with given id.

```
GET /kypo2-user-and-group/api/v1/groups/{id}/roles
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer(int64)

Responses

HTTP Code	Description	Schema
200	successful operation	< RoleDTO > array

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups/0/roles
```

Example HTTP response

Response 200

```
"array"
```

2.1.10. Add users to group.

```
PUT /kypo2-user-and-group/api/v1/groups/{id}/users
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Id of group to add users.	integer(int64)
Body	body <i>required</i>	Ids of members to be added and ids of groups of imported members to group.	AddUsersToGroupDTO

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Consumes

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups/0/users
```

Request body

```
{  
  "ids_of_users_to_be_add" : [ 0 ],  
  "ids_of_groups_of_imported_users" : [ 0 ]  
}
```

2.1.11. Remove users from input group.

```
DELETE /kypo2-user-and-group/api/v1/groups/{id}/users
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Id of group to remove users.	integer(int64)
Body	body <i>required</i>	Ids of members to be removed from group.	< integer(int64) > array

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/groups/0/users
```

Request body

```
[ 0 ]
```

2.2. Roles

2.2.1. Get all roles

```
GET /kypo2-user-and-group/api/v1/roles
```

Parameters

Type	Name	Description	Schema
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(,asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)
Body	body <i>optional</i>	Parameters for filtering the objects.	< string, < string > array > map

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/roles
```

Request query

```
{
  "fields" : "string",
  "page" : 0,
  "size" : 0,
  "sort" : "asc"
}
```

Request body

```
{ }
```

Example HTTP response

Response 200

```
"object"
```

2.2.2. Gets all users with given role.

```
GET /kypo2-user-and-group/api/v1/roles/users
```

Parameters

Type	Name	Description	Schema
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer
Query	roleType <i>required</i>	Type of role to get users for.	string
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)
Body	body <i>optional</i>	Parameters for filtering the objects.	< string, < string > array > map

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/roles/users
```

Request query

```
{
  "fields" : "string",
  "page" : 0,
  "roleType" : "string",
  "size" : 0,
  "sort" : "asc"
}
```

Request body

```
{ }
```

Example HTTP response

Response 200

```
"object"
```

2.2.3. Gets all users with given role and not with given ids.

```
GET /kypo2-user-and-group/api/v1/roles/users-not-with-ids
```

Parameters

Type	Name	Description	Schema
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	ids <i>required</i>	Ids of the users to be excluded from the result.	< integer(int64) > array(multi)
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer

Type	Name	Description	Schema
Query	roleType <i>required</i>	Type of role to get users for.	string
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)
Body	body <i>optional</i>	Parameters for filtering the objects.	< string, < string > array > map

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/roles/users-not-with-ids
```

Request query

```
{
  "fields" : "string",
  "ids" : 0,
  "page" : 0,
  "roleType" : "string",
  "size" : 0,
  "sort" : "asc"
}
```

Request body

```
{ }
```

Example HTTP response

Response 200

```
"object"
```

2.2.4. Get role with given id

```
GET /kypo2-user-and-group/api/v1/roles/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Id of role to be returned	integer(int64)

Responses

HTTP Code	Description	Schema
200	successful operation	RoleDTO

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/roles/0
```

Example HTTP response

Response 200

```
{
  "id_of_microservice" : 5,
  "name_of_microservice" : "kypo2-training",
  "description" : "This role will allow you to create and delete groups.",
  "role_type" : "ROLE_USER_AND_GROUP_ADMINISTRATOR",
  "id" : 1
}
```

2.2.5. Gets all users with given role.

GET /kypo2-user-and-group/api/v1/roles/{roleId}/users

Parameters

Type	Name	Description	Schema
Path	roleId <i>required</i>	Type of role to get users for.	integer(int64)
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(,asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)
Body	body <i>optional</i>	Parameters for filtering the objects.	< string, < string > array > map

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

/kypo2-user-and-group/api/v1/roles/0/users

Request query

```
{
  "fields" : "string",
  "page" : 0,
  "size" : 0,
  "sort" : "asc"
}
```

Request body

```
{ }
```

Example HTTP response

Response 200

```
"object"
```

2.3. Users

2.3.1. Gets all users.

```
GET /kypo2-user-and-group/api/v1/users
```

Parameters

Type	Name	Description	Schema
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(,asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)
Body	body <i>optional</i>	Parameters for filtering the objects.	< string, < string > array > map

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users
```

Request query

```
{
  "fields" : "string",
  "page" : 0,
  "size" : 0,
  "sort" : "asc"
}
```

Request body

```
{ }
```

Example HTTP response

Response 200

```
"object"
```

2.3.2. Tries to delete users with given ids and returns users and statuses of their deletion.

```
DELETE /kypo2-user-and-group/api/v1/users
```

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Ids of users to be deleted.	< integer(int64) > array

Responses

HTTP Code	Description	Schema
200	successful operation	< UserDeletionResponseDTO > array

Consumes

- `application/json`

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users
```

Request body

```
[ 0 ]
```

Example HTTP response

Response 200

```
"array"
```

2.3.3. Gets users in given groups.

```
GET /kypo2-user-and-group/api/v1/users/groups
```

Parameters

Type	Name	Description	Schema
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	ids <i>required</i>	Ids of groups where users are assigned.	< integer(int64) > array(multi)
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)
Body	body <i>optional</i>	Parameters for filtering the objects.	< string, < string > array > map

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users/groups
```

Request query

```
{
  "fields" : "string",
  "ids" : 0,
  "page" : 0,
  "size" : 0,
  "sort" : "asc"
}
```

Request body

```
{ }
```

Example HTTP response

Response 200

```
"object"
```

2.3.4. Gets users with given ids.

```
GET /kypo2-user-and-group/api/v1/users/ids
```

Parameters

Type	Name	Description	Schema
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	ids <i>required</i>	Ids of users to be obtained.	< integer(int64) > array(multi)
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(,asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)
Body	body <i>optional</i>	Parameters for filtering the objects.	< string, < string > array > map

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users/ids
```

Request query

```
{
  "fields" : "string",
  "ids" : 0,
  "page" : 0,
  "size" : 0,
  "sort" : "asc"
}
```

Request body

```
{ }
```


Example HTTP response

Response 200

```
"object"
```

2.3.5. Returns details of user who is logged in

```
GET /kypo2-user-and-group/api/v1/users/info
```

Responses

HTTP Code	Description	Schema
200	successful operation	UserDTO

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users/info
```

Example HTTP response

Response 200

```
{
  "full_name" : "Michael Smith",
  "login" : "michaelsmith",
  "mail" : "michaelsmith@mail.muni.cz.",
  "given_name" : "Pavel",
  "family_name" : "Seda",
  "roles" : [ {
    "id_of_microservice" : 5,
    "name_of_microservice" : "kypo2-training",
    "description" : "This role will allow you to create and delete groups.",
    "role_type" : "ROLE_USER_AND_GROUP_ADMINISTRATOR",
    "id" : 1
  } ],
  "iss" : "https://oidc.muni.cz",
  "picture" : [ "string" ],
  "id" : 1
}
```

2.3.6. Gets all users except users in given group.

```
GET /kypo2-user-and-group/api/v1/users/not-in-groups/{groupId}
```

Parameters

Type	Name	Description	Schema
Path	groupId <i>required</i>	Id of group whose users do not get.	integer(int64)
Query	fields <i>optional</i>	Fields which should be returned in REST API response	string
Query	page <i>optional</i>	Results page you want to retrieve (0..N)	integer
Query	size <i>optional</i>	Number of records per page.	integer
Query	sort <i>optional</i>	Sorting criteria in the format: property(asc desc). Default sort order is ascending. Multiple sort criteria are supported.	< string > array(multi)

Responses

HTTP Code	Description	Schema
200	successful operation	object

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users/not-in-groups/0
```

Request query

```
{
  "fields" : "string",
  "page" : 0,
  "size" : 0,
  "sort" : "asc"
}
```

Example HTTP response

Response 200

```
"object"
```

2.3.7. Gets user with given id.

```
GET /kypo2-user-and-group/api/v1/users/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Id of user to be returned.	integer(int64)

Responses

HTTP Code	Description	Schema
200	successful operation	UserDTO

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users/0
```

Example HTTP response

Response 200

```
{
  "full_name" : "Michael Smith",
  "login" : "michaelsmith",
  "mail" : "michaelsmith@mail.muni.cz.",
  "given_name" : "Pavel",
  "family_name" : "Seda",
  "roles" : [ {
    "id_of_microservice" : 5,
    "name_of_microservice" : "kypo2-training",
    "description" : "This role will allow you to create and delete groups.",
    "role_type" : "ROLE_USER_AND_GROUP_ADMINISTRATOR",
    "id" : 1
  } ],
  "iss" : "https://oide.muni.cz",
  "picture" : [ "string" ],
  "id" : 1
}
```

2.3.8. Tries to delete user with given screen name and returns status of its result.

```
DELETE /kypo2-user-and-group/api/v1/users/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	Screen name of user to be deleted.	integer(int64)

Responses

HTTP Code	Description	Schema
200	successful operation	UserDeletionResponseDTO

Produces

- `application/json`

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users/0
```

Example HTTP response

Response 200

```
{
  "user" : {
    "full_name" : "Michael Smith",
    "login" : "michaelsmith",
    "mail" : "michaelsmith@mail.muni.cz.",
    "given_name" : "Pavel",
    "family_name" : "Seda",
    "roles" : [ {
      "id_of_microservice" : 5,
      "name_of_microservice" : "kypo2-training",
      "description" : "This role will allow you to create and delete groups.",
      "role_type" : "ROLE_USER_AND_GROUP_ADMINISTRATOR",
      "id" : 1
    } ],
    "iss" : "https://oidc.muni.cz",
    "picture" : [ "string" ],
    "id" : 1
  },
  "status" : "SUCCESS"
}
```

2.3.9. Returns all roles of user with given id.

```
GET /kypo2-user-and-group/api/v1/users/{id}/roles
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	id	integer(int64)

Responses

HTTP Code	Description	Schema
200	successful operation	< RoleDTO > array

Example HTTP request

Request path

```
/kypo2-user-and-group/api/v1/users/0/roles
```

Example HTTP response

Response 200

```
"array"
```

3. Definitions

3.1. AddUsersToGroupDTO

Name	Description	Schema
ids_of_groups_of_imported_users <i>optional</i>	Main group identifiers whose users are to be imported into a group. Example : [0]	< integer(int64) > array
ids_of_users_to_be_add <i>optional</i>	Main identifiers of users to be added to group. Example : [0]	< integer(int64) > array

3.2. GroupDTO

Name	Description	Schema
can_be_deleted <i>optional</i>	Sign if the group can be deleted. Example : false	boolean
description <i>optional</i>	A description of the group. Example : "Organizers group for training run in June."	string
expiration_date <i>optional</i>	Time until the group is valid. Example : "2017-10-19 10:23:54+02"	string(date-time)
id <i>optional</i>	Main identifier of group. Example : 1	integer(int64)
name <i>optional</i>	A name of the group. Example : "Main group of organizers"	string
roles <i>optional</i>	Set of roles of group. Example : ["RoleDTO"]	< RoleDTO > array
source <i>optional</i>	Source of the group, whether its internal or from perun. Example : "Internal"	enum (INTERNAL, PERUN)
users <i>optional</i>	List of users in group. Example : ["UserForGroupsDTO"]	< UserForGroupsDTO > array

3.3. GroupDeletionResponseDTO

Name	Description	Schema
id <i>optional</i>	Main identifiers of deleted group. Example : 1	integer(int64)
status <i>optional</i>	Result of deleting group: 1) SUCCESS - group was deleted 2) NOT_FOUND - group has not found 3) ERROR - group could not be deleted, try it later 4) ERROR_MAIN_GROUP - group cannot be deleted due to it is one of the main group for roles (ADMINISTRATOR, USER, GUEST) Example : "SUCCESS"	enum (EXTERNAL_VALID, SUCCESS, ERROR, NOT_FOUND, ERROR_MAIN_GROUP)

3.4. NewGroupDTO

Name	Description	Schema
description <i>required</i>	A description of the group. Example : "Group for main users."	string
expiration_date <i>optional</i>	Time until the group is valid. Example : "2019-11-20T10:28:02.727Z"	string(date-time)
group_ids_of_imported_users <i>optional</i>	Main identifiers of group. Example : [0]	< integer(int64) > array
name <i>required</i>	A name of the group. Example : "Main group"	string
users <i>optional</i>	List of users who is assigned to group. Example : ["UserForGroupsDTO"]	< UserForGroupsDTO > array

3.5. RoleDTO

Name	Description	Schema
description <i>optional</i>	A description of what the user is capable of with this role. Example : "This role will allow you to create and delete groups."	string
id <i>optional</i>	Main identifier of the role. Example : 1	integer(int64)
id_of_microservice <i>optional</i>	Id of microservice which use this role. Example : 5	integer(int64)
name_of_microservice <i>optional</i>	A name of microservice which use this role. Example : "kypo2-training"	string

Name	Description	Schema
role_type <i>optional</i>	Role type of role. Example : "ROLE_USER_AND_GROUP_ADMINISTRATOR"	string

3.6. UpdateGroupDTO

Name	Description	Schema
description <i>required</i>	A description of the group. Example : "Group for main users."	string
expiration_date <i>optional</i>	Time until the group is valid. Example : "2019-11-20T10:28:02.727Z"	string(date-time)
id <i>required</i>	Main identifier of group. Example : 1	integer(int64)
name <i>required</i>	A name of the group. Example : "Main group."	string

3.7. UserDTO

Name	Description	Schema
family_name <i>optional</i>	Surname of a user. Example : "Seda"	string
full_name <i>optional</i>	Full name of the user. Example : "Michael Smith"	string
given_name <i>optional</i>	First name of a user. Example : "Pavel"	string
id <i>optional</i>	Main identifier of the user. Example : 1	integer(int64)
iss <i>optional</i>	Issuer of a user. Example : "https://oidc.muni.cz"	string
login <i>optional</i>	Login of the user. Example : "michaelsmith"	string
mail <i>optional</i>	Email of the user. Example : "michaelsmith@mail.muni.cz."	string
picture <i>optional</i>	Identicon of a user. Example : ["string"]	< string(byte) > array
roles <i>optional</i>	Roles of user assigned through the groups which user is in. Example : ["RoleDTO"]	< RoleDTO > array

3.8. UserDeletionResponseDTO

Name	Description	Schema
status <i>optional</i>	Status about result of deletion. Example : "SUCCESS"	enum (SUCCESS, EXTERNAL_VALID, ERROR, NOT_FOUND)
user <i>optional</i>	User to be deleted. Example : "UserDTO"	UserDTO

3.9. UserForGroupsDTO

Name	Description	Schema
family_name <i>optional</i>	Surname of a user. Example : "Seda"	string
full_name <i>optional</i>	Full name of the user including his titles. Example : "Peter Novak."	string
given_name <i>optional</i>	First name of a user. Example : "Pavel"	string
id <i>optional</i>	Main identifiers of the user. Example : 1	integer(int64)
iss <i>optional</i>	Issuer of a user. Example : "https://oidc.muni.cz"	string
login <i>optional</i>	Login of the user. Example : "{\"sub\":\"michaelsmith\", \"iss\": \"https://oidc.muni.cz/oidc/\"}"	string
mail <i>optional</i>	Email of the user. Example : "michaelsmith@mail.muni.cz."	string
picture <i>optional</i>	Identicon of a user. Example : ["string"]	< string(byte) > array