Package ‘susographql’

January 17, 2024

Type Package

Title Comprehensive Interface to the Survey Solutions 'GraphQL' API

Version 0.1.6

Description Provides a complete suite of tools for interacting with the Survey Solutions 'GraphQL' API <https://demo.mysurvey.solutions/graphql/>. This package encompasses all currently available queries and mutations, including the latest features for map uploads. It is built on the modern 'httr2' package, offering a streamlined and efficient interface without relying on external 'GraphQL' client packages. In addition to core API functionalities, the package includes a range of helper functions designed to facilitate the use of available query filters.

License GPL (>= 3)

Encoding UTF-8

Imports curl (>= 5.1.0), glue, httr2, jsonlite, readr, stringr, data.table, withr, cli (>= 3.0.0), rlang (>= 1.1.0), lubridate

Depends R (>= 3.6)

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

URL https://michael-cw.github.io/susographql/

BugReports https://github.com/michael-cw/susographql/issues

Config/testthat/edition 3

NeedsCompilation no

Author Michael Wild [cre, aut, cph]

Maintainer Michael Wild <mwild@worldbank.org>

Repository CRAN

Date/Publication 2024-01-17 19:10:02 UTC
R topics documented:

susoop_num ......................................................... 2
susoop_str ........................................................... 3
suso_gql_addassignmentcalendarevent .......................... 4
suso_gql_addinterviewcalendarevent ............................ 5
suso_gql_addusertomap ............................................. 6
suso_gql_assignments .............................................. 7
suso_gql_deletecalendarevent ..................................... 8
suso_gql_deletemap ............................................... 9
suso_gql_deleteuserfrommap ..................................... 10
suso_gql_interviews .............................................. 11
suso_gql_mapreport ............................................... 14
suso_gql_maps ................................................... 16
suso_gql_pwcheck ................................................ 18
suso_gql_questionnaireitems .................................... 19
suso_gql_questionnaires ......................................... 20
suso_gql_questions ............................................... 21
suso_gql_updatecalendarevent .................................. 23
suso_gql_uploadmap .............................................. 24
suso_gql_users .................................................. 26
transformers.numeric ........................................... 27
transformers.string .............................................. 29

Index 31

susoop_num List object for Survey Solutions GraphQL numeric operator selection

Description

A list of the available transformers

Usage

susoop_num

Format

An object of class list of length 12.

Details

Allows the user to select the operator for the required filter.

Value

A named list with the operator and the value to be passed on as input to the filter.
Examples

# equal to 3
susoop_num$eq(3)

# not equal to 3
susoop_num$neq(3)

---

**susoop_str**

*List object for Survey Solutions GraphQL character operator selection*

**Description**

A list of the available transformers

**Usage**

`susoop_str`

**Format**

An object of class list of length 10.

**Details**

Allows the user to select the operator for the required filter.

**Value**

A named list with the operator and the value to be passed on as input to the filter.

**Examples**

# equal to 3
susoop_str$contains("area10")

# not equal to 3
susoop_str$startsWith("area")
suso_gql_addassignmentcalendarevent

Add a calendar event to an Assignment

Description
Add a calendar event to an assignment

Usage
suso_gql_addassignmentcalendarevent(
   endpoint = NULL,
   workspace = NULL,
   user = NULL,
   password = NULL,
   token = NULL,
   assignmentId = NULL,
   comment = NULL,
   newStart = NULL,
   startTimezone = "UTC"
)

Arguments
- endpoint: GraphQL endpoint of your server
- workspace: Server Workspace, if NULL uses default
- user: your API username
- password: your API user password
- token: If Survey Solutions server token is provided apiUser and apiPass will be ignored
- assignmentId: assignment id
- comment: a comment string
- newStart: new start date, format must be: 2024-01-16 01:41:14
- startTimezone: time zone of the tablet device, use OlsonNames

Value
if successfull, returns a list with the executed mutation

Examples

## Requires Survey Solutions Server and API credentials

Create a calendar event for assignment 208 on 2024-01-16 01:41:14, in timezone Chile/Continental
suso_gql_addassignmentcalendar(event = ep, user = usr, password = pass, workspace = ws, assignmentId = 208, newStart = "2024-01-16 01:41:14", startTimezone = "Chile/Continental", comment = "Lets Go!")

suso_gql_addinterviewcalendar(event)

Add a calendar event to an Interview

Description

Add a calendar event to an Interview

Usage

suso_gql_addinterviewcalendar(event = NULL, endpoint = NULL, workspace = NULL, user = NULL, password = NULL, token = NULL, interviewId = NULL, comment = NULL, newStart = NULL, startTimezone = "UTC")

Arguments

endpoint GraphQL endpoint of your server
workspace Server Workspace, if NULL uses default
user your API username
password your API user password
token If Survey Solutions server token is provided apiUser and apiPass will be ignored
interviewId the interviewId
comment a comment string
newStart new start date, format must be: 2024-01-16 01:41:14
startTimezone time zone of the tablet device, use OlsonNames

Value

if successfull, returns a list with the executed mutation
suso_gql_addusertomap

Examples

```r
# Requires Survey Solutions Server and API credentials

# Create a calendar event for a selected interview on 2024-01-16 01:41:14, # in timezone UTC

date = as.POSIXct("2024-01-16 01:41:14", tzone = "UTC")
suso_gql_addinterviewcalendarevent(endpoint = ep, user = usr,
password = pass, workspace = ws, interviewId = "bac4d1525b541dd86a003dc7f854e35",
newStart = date, startTimezone = "UTC", comment = "Lets Go!")
```

suso_gql_addusertomap  Assigns a map to a user

Description

Allows the user to assign a map to an interviewer to be used in CAPI data collection.

Usage

```r
suso_gql_addusertomap(
  endpoint = NULL,
  workspace = NULL,
  user = NULL,
  password = NULL,
  token = NULL,
  fileName = NULL,
  userName = NULL
)
```

Arguments

- **endpoint**  GraphQL endpoint of your server
- **workspace**  Server Workspace, if NULL uses default
- **user**  your API username
- **password**  your API user password
- **token**  If Survey Solutions server token is provided apiUser and apiPass will be ignored
- **fileName**  the name of the map file on the server
- **userName**  the name of the interviewer to whom the map will be assigned to

Value

if successfull, returns a list with the executed mutation
Examples

```r
## Requires Survey Solutions Server and API credentials

# Assign map seg_168_ALL.tif to user int0073

suso_gql_addusertomap(endpoint = ep, user = usr,
password = pass, workspace = ws,
fileName = "seg_168_ALL.tif", userName = "int0073")
```

## Get all assignments

**Description**

Allows the user to retrieve filtered or unfiltered assignment data.

**Usages**

```r
suso_gql_assignments(
  endpoint = NULL,
  workspace = NULL,
  user = NULL,
  password = NULL,
  token = NULL,
  archived = FALSE,
  id = NULL,
  questionnaireId = NULL,
  version = NULL,
  responsibleId = NULL,
  webMode = FALSE,
  take = NULL,
  skip = NULL
)
```

**Arguments**

- **endpoint**: GraphQL endpoint of your server
- **workspace**: Server Workspace, if NULL uses default
- **user**: your API username
- **password**: API password
- **token**: If Survey Solutions server token is provided `apiUser` and `apiPass` will be ignored
- **archived**: Boolean if the assignment is archived or not
- **id**: Assignment ID
suso_gql_deletecalendarevent

**Description**

Delete a calendar event

**Usage**

```r
suso_gql_deletecalendarevent(
    endpoint = NULL,
    workspace = NULL,
    user = NULL,
    password = NULL,
    token = NULL,
    publicKey = NULL
)
```
Arguments

- **endpoint**: GraphQL endpoint of your server.
- **workspace**: Server Workspace, if NULL uses default.
- **user**: your API username.
- **password**: your API user password.
- **token**: If Survey Solutions server token is provided, apiUser and apiPass will be ignored.
- **publicKey**: the publicKey of the calendar event to be deleted.

Value

if successfull, returns a list with the executed mutation

Examples

```python
## Requires Survey Solutions Server and API credentials

# Delete a calendar event for a specific event
# in timezone Chile/Continental

suso_gql_deletecalendarevent(endpoint = ep, user = usr, password = pass, workspace = ws, publicKey = "ea8b49c0-0d16-4fec-9029-72278e94a233")
```

Description

Allows the user to delete a map.

Usage

```python
suso_gql_deletemap(
    endpoint = NULL,
    workspace = NULL,
    user = NULL,
    password = NULL,
    token = NULL,
    fileName = NULL
)
```
suso_gql_deleteuserfrommap

Arguments

- endpoint: GraphQL endpoint of your server
- workspace: Server Workspace, if NULL uses default
- user: your API username
- password: your API user password
- token: If Survey Solutions server token is provided apiUser and apiPass will be ignored
- fileName: the name of the map file on the server

Value

if successfull, returns a list with the executed mutation

Examples

```r
## Requires Survey Solutions Server and API credentials

# Delete map seg_168_ALL.tif
suso_gql_deleteuserfrommap(endpoint = ep, user = usr, password = pass, workspace = ws, fileName = "seg_168_ALL.tif")
```

Description

Allows the user to delete an interviewer from a map to be used in CAPI data collection.

Usage

```r
suso_gql_deleteuserfrommap(
  endpoint = NULL,
  workspace = NULL,
  user = NULL,
  token = NULL,
  password = NULL,
  fileName = NULL,
  userName = NULL
)
```
suso_gql_interviews

Arguments

- **endpoint**: GraphQL endpoint of your server
- **workspace**: Server Workspace, if NULL uses default
- **user**: your API username
- **token**: If Survey Solutions server token is provided apiUser and apiPass will be ignored
- **password**: your API user password
- **fileName**: the name of the map file on the server
- **userName**: the name of the interviewer to whom the map will be assigned to

Value

- if successfull, returns a list with the executed mutation

Examples

```r
## Requires Survey Solutions Server and API credentials

# Delete map seg_168_ALL.tif from user int0073

suso_gql_deleteuserfrommap(endpoint = ep, user = usr, 
password = pass, workspace = ws, 
fileName = "seg_168_ALL.tif", userName = "int0073")
```

---

suso_gql_interviews _Get all interviews_

Description

- Allows the user to retrieve filtered or unfiltered interview data.

Usage

```r
suso_gql_interviews(
    endpoint = NULL, 
    workspace = NULL, 
    user = NULL, 
    password = NULL, 
    token = NULL, 
    assignmentId = NULL, 
    clientKey = NULL, 
    createdDate = NULL, 
    errorsCount = NULL, 
    identifyingData = NULL,
```
interviewMode = NULL,
notAnsweredCount = NULL,
questionnaireId = NULL,
questionnaireVersion = NULL,
questionnaireVariable = NULL,
responsibleName = NULL,
responsibleRole = NULL,
status = NULL,
supervisorName = NULL,
sortby_updateDateUtc = NULL,
sortby_notAnsweredCount = NULL,
sortby_createdDate = NULL,
sortby_errorsCount = NULL,
sortby_assignmentId = NULL,
sortby_status = NULL,
sortby_receivedByInterviewerAtUtc = NULL,
sortby_responsibleName = NULL,
take = NULL,
skip = NULL
)

Arguments

endpoint  GraphQL endpoint of your server
workspace  Server Workspace, if NULL uses default
user  your API username
password  API password
token  If Survey Solutions server token is provided apiUser and apiPass will be ignored
assignmentId  Assignment ID
clientKey  Interview key
createdDate  Creation data of the interview
errorsCount  number of errors
identifyingData  Pre-loaded identifying data
interviewMode  Interview mode (CAWI or CAPI)
notAnsweredCount  number of unanswered questions
questionnaireId  the questionnaire id
questionnaireVersion  the questionnaire version
questionnaireVariable  the variable for the questionnaire
responsibleName  Name of the person responsible
suso_gql_interviews

responsibleRole
Role of the person responsible

status
Role of the interview

supervisorName
Name of the supervisor of the responsible user

sortby_updateDateUtc
sort by updated at UTC time, either ASC for ascending or DESC for descending

sortby_notAnsweredCount
sort by errors count, either ASC for ascending or DESC for descending

sortby_createdDate
sort by creation date, either ASC for ascending or DESC for descending

sortby_errorsCount
sort by number of errors, either ASC for ascending or DESC for descending

sortby_assignmentId
sort by assignmentId, either ASC for ascending or DESC for descending

sortby_status
sort by interview status, either ASC for ascending or DESC for descending

sortby_receivedByInterviewerAtUtc
sort by date the table received the interview, either ASC for ascending or DESC for descending

sortby_responsibleName
sort by the current responsible user, either ASC for ascending or DESC for descending

take
take the specified integer number of interviews

skip
skip the first integer number of interviews

Value
if successfull, returns a list with the (filtered) responses

Examples

## Requires Survey Solutions Server and API credentials

# Get all interviews without filter
suso_gql_interviews(endpoint = ep, user = usr, password = pass, workspace = "primary")

# Select interviews which are not completed yet
suso_gql_interviews(endpoint = ep, user = usr, password = pass, workspace = "primary", status = susoop_str$neq("COMPLETED"))

# Select interviews which have 3 or more errors
suso_gql_assignments(endpoint = ep, user = usr, password = pass, workspace = "primary", errorsCount = susoop_num$gte(3))

# Select interviews which have less than 3 errors
suso_gql_assignments(endpoint = ep, user = usr, password = pass, workspace = "primary", errorsCount = susoop_num$ngt(3))
suso_gql_mapreport  Get the map report

Description

Allows the user to retrieve filtered or unfiltered map report data.

Usage

```r
suso_gql_mapreport(
    endpoint = NULL,
    workspace = NULL,
    user = NULL,
    password = NULL,
    token = NULL,
    questionnaireId = NULL,
    questionnaireVersion = NULL,
    variable = NULL,
    zoom = 1,
    clientMapWidth = 0,
    west = -180,
    east = 180,
    north = 90,
    south = -90,
    assignmentId = NULL,
    clientKey = NULL,
    createdDate = NULL,
    errorsCount = NULL,
    identifyingData = NULL,
    interviewMode = NULL,
    notAnsweredCount = NULL,
    questionnaireVariable = NULL,
    responsibleName = NULL,
    responsibleRole = NULL,
    status = NULL,
    supervisorName = NULL
)
```

Arguments

- `endpoint`  GraphQL endpoint of your server
workspace: Server Workspace, if NULL uses default
user: your API username
password: API password
token: If Survey Solutions server token is provided apiUser and apiPass will be ignored
questionnaireId: the questionnaire id
questionnaireVersion: the questionnaire version
variable: Variable(s) of interest
zoom: Zoom of the map report
clientMapWidth: width of the map report
west: coordinates for bounding box
east: coordinates for bounding box
north: coordinates for bounding box
south: coordinates for bounding box
assignmentId: Assignment ID
clientKey: Interview key
createdDate: Creation data of the interview
errorsCount: number of errors
identifyingData: Pre-loaded identifying data
interviewMode: Interview mode (CAWI or CAPI)
notAnsweredCount: number of unanswered questions
questionnaireVariable: the variable for the questionnaire
responsibleName: Name of the person responsible
responsibleRole: Role of the person responsible
status: of the interview
supervisorName: Name of the supervisor of the responsible user

**Value**

if successfull, returns a list with the (filtered) responses
Examples

```r
## Requires Survey Solutions Server and API credentials
questlist <- suso_gql_questionnaires(
    endpoint = ep, user = usr, 
    password = pass, 
    workspace = "primary")

id <- questlist$questionnaires$nodes$questionnaireId[1]

v <- questlist$questionnaires$nodes$version[1]

# Get map report for GPS question start_location
suso_gql_mapreport(endpoint = ep, user = usr, 
    password = pass, workspace = ws, 
    questionnaireId = id, questionnaireVersion = v, variable = "start_location")
```

---

### suso_gql_maps

*Get all maps*

**Description**

Allows the user to retrieve filtered or unfiltered map data.

**Usage**

```r
suso_gql_maps(
    endpoint = NULL, 
    workspace = NULL, 
    user = NULL, 
    password = NULL, 
    token = NULL, 
    fileName = NULL, 
    importedDateUtc = NULL, 
    size = NULL, 
    userName = NULL, 
    sortby_filename = NULL, 
    sortby_importeddateutc = NULL, 
    sortby_size = NULL, 
    take = NULL, 
    skip = NULL
)
```
suso_gql_maps

Arguments

- **endpoint**: GraphQL endpoint of your server
- **workspace**: Server Workspace, if NULL uses default
- **user**: your API username
- **password**: API password
- **token**: If Survey Solutions server token is provided *apiUser* and *apiPass* will be ignored
- **fileName**: name of the map on the server
- **importDateUtc**: Import date of map
- **size**: Size of the map
- **userName**: User name to whom the map(s) are assigned to.
- **sortby_filename**: sort maps by file name, either ASC for ascending or DESC for descending
- **sortby_importeddateutc**: sort maps by import date in utc, either ASC for ascending or DESC for descending
- **sortby_size**: sort by map size, either ASC for ascending or DESC for descending
- **take**: take the specified integer number of maps
- **skip**: skip the first integer number of maps

Value

if successfull, returns a list with the (filtered) responses

Examples

```r
## Requires Survey Solutions Server and API credentials

# Get all maps without filter
suso_gql_maps(endpoint = ep, user = usr, password = pass, workspace = ws)

# Get only boundary files (.shp)

suso_gql_maps(endpoint = ep, user = usr, password = pass, workspace = ws, fileName = susoop_str$endsWith(".shp"))
```
suso_gql_pwcheck  

Utility function to check if credentials are correct

Description

This function returns a 200 status code if credentials are correct and a 400 code otherwise.

Usage

suso_gql_pwcheck(
    endpoint = NULL,
    user = NULL,
    password = NULL,
    workspace = NULL,
    token = NULL
)

Arguments

- **endpoint**: Survey Solutions GraphQL
- **user**: API user
- **password**: API password
- **workspace**: endpoint workspace Name, if nothing provided, defaults to primary
- **token**: If Survey Solutions endpoint token is provided user and password will be ignored

Value

200 code if correct, 400 if incorrect.

Examples

# With default credentials returns 400
suso_gql_pwcheck()
suso_gql_questionnaireitems

*Get all Questions in a Questionnaires*

**Description**

Allows the user to retrieve all questions in a questionnaire.

**Usage**

```r
suso_gql_questionnaireitems(
    endpoint = NULL,
    workspace = NULL,
    user = NULL,
    password = NULL,
    token = NULL,
    id = NULL,
    version = NULL,
    variable = NULL,
    scope = NULL,
    identifying = NULL,
    title = NULL,
    includedInReportingAtUtc = NULL
)
```

**Arguments**

- `endpoint` : GraphQL endpoint of your server
- `workspace` : Server Workspace, if NULL uses default
- `user` : your API username
- `password` : API password
- `token` : If Survey Solutions server token is provided `apiUser` and `apiPass` will be ignored
- `id` : Questionnaire ID
- `version` : Questionnaire version
- `variable` : Get questions for a specific variable
- `scope` : Get questions for a specific scope
- `identifying` : If TRUE only identifying questions are exported
- `title` : the text of the questions
- `includedInReportingAtUtc` : time of the last reporting

**Value**

if successfull, returns a list with the (filtered) responses
Examples

```r
## Requires Survey Solutions Server and API credentials
questlist<-suso_gql_questionnaires(
    endpoint = ep, user = usr,
    password = pass,
    workspace = "primary")

id<-questlist$questionnaires$nodes$questionnaireId[1]
v<-questlist$questionnaires$nodes$version[1]

# Get all questions from a questionnaire/version without filter
suso_gql_questionnaireitems(endpoint = ep, user = usr,
    password = pass,
    workspace = ws,
    id = id,
    version = v)

# Select only identifying questions
suso_gql_questionnaireitems(endpoint = ep, user = usr,
    password = pass,
    workspace = ws,
    id = id,
    version = v,
    identifying = TRUE)

# Select only questions which have not interviewer scope
suso_gql_questionnaireitems(endpoint = ep, user = usr,
    password = pass,
    workspace = ws,
    id = id,
    version = v,
    scope = susoop_str$neq("INTERVIEWER"))
```

---

`suso_gql_questionnaires`

*Get all Questionnaires*

**Description**

Get all Questionnaires in the workspace or only a selected one/versions.

**Usage**

```r
suso_gql_questionnaires(
    endpoint = NULL,
```
suso_gql_questions

workspace = NULL,
user = NULL,
password = NULL,
token = NULL,
id = NULL,
version = NULL,
take = NULL,
skip = NULL
)

Arguments

endpoint  GraphQL endpoint of your server
workspace  Server Workspace, if NULL uses default
user  your API username
password  API password
token  If Survey Solutions server token is provided apiUser and apiPass will be ignored
id  Questionnaire ID
version  Questionnaire version
take  take the specified integer number of questionnaires
skip  skip the first integer number of questionnaires

Value

if successfull, returns a list with the responses

Examples

## Requires Survey Solutions Server and API credentials

# Get all questionnaires in workspace ws

suso_gql_questionnaires(endpoint = ep, user = usr,
password = pass, workspace = ws)

suso_gql_questions  Get all Questions in a Questionnaires

Description

Allows the user to retrieve all questions in a questionnaire.
suso_gql_questions

Usage

suso_gql_questions(
    endpoint = NULL,
    workspace = NULL,
    user = NULL,
    password = NULL,
    token = NULL,
    id = NULL,
    version = NULL,
    variable = NULL,
    scope = NULL,
    identifying = NULL
)

Arguments

endpoint      GraphQL endpoint of your server
workspace     Server Workspace, if NULL uses default
user          your API username
password      API password
token         If Survey Solutions server token is provided apiUser and apiPass will be ignored
id            Questionnaire ID
version       Questionnaire version
variable      Get questions for a specific variable
scope         Get questions for a specific scope
identifying   If TRUE only identifying questions are exported

Value

if successfull, returns a list with the (filtered) responses

Examples

## Requires Survey Solutions Server and API credentials

questlist<-suso_gql_questionnaires(
    endpoint = ep, user = usr,
    password = pass,
    workspace = "primary")

id<-questlist$questionnaires$nodes$questionnaireId[1]
v<-questlist$questionnaires$nodes$version[1]

# Get all questions from a questionnaire/version without filter
suso_gql_questions(endpoint = ep, user = usr,
**suso_gql_updatecalendarevent**

Update a calendar event

### Description

Update a calendar event

### Usage

```r
suso_gql_updatecalendarevent(
  endpoint = NULL,
  workspace = NULL,
  user = NULL,
  password = NULL,
  token = NULL,
  publicKey = NULL,
  comment = NULL,
  newStart = NULL,
  startTimezone = "UTC"
)
```

### Arguments

- **endpoint**: GraphQL endpoint of your server.
- **workspace**: Server Workspace, if NULL uses default.
user your API username.
password your API user password.
token If Survey Solutions server token is provided apiUser and apiPass will be ignored.
publicKey the publicKey of the calendar event to be updated.
comment a comment string.
newStart new start date, format must be: 2024-01-16 01:41:14.
startTimezone time zone of the tablet device, use OlsonNames.

Value

if successfull, returns a list with the executed mutation

Examples

## Requires Survey Solutions Server and API credentials

# Update a calendar event for a specific event to 2024-01-18 01:41:14,
# in timezone Chile/Continental

suso_gql_updatecalendarevent(endpoint = ep, user = usr,
password = pass, workspace = ws, publicKey = "ea8b49c0-0d16-4fec-9029-72278e94a233",
newStart = "2024-01-16 01:41:14",
startTimezone = "Chile/Continental", comment = "Lets Go Again!")

---

suso_gql_uploadmap   Upload map to server

Description

Allows the user to upload 1 or several zip files with base maps and shapefiles to the server.

Usage

suso_gql_uploadmap(    
    endpoint = NULL,    
    workspace = NULL,    
    user = NULL,    
    token = NULL,    
    password = NULL,    
    path_to_zip = NULL,    
    usePar = FALSE,    
    n_par = 10    
)
Arguments

- **endpoint**: GraphQL endpoint of your server.
- **workspace**: Server Workspace, if NULL uses default.
- **user**: your API username.
- **token**: If Survey Solutions server token is provided apiUser and apiPass will be ignored.
- **password**: API password.
- **path_to_zip**: path to a single zip file or path to a directory of zip files if usePar=TRUE.
- **usePar**: if TRUE the requests will be performed in parallel EXPERIMENTAL!
- **n_par**: number of parallel requests, required if usePar = TRUE EXPERIMENTAL!

Details

If usePar = TRUE a path to a directory needs to be provided containing the zipped map files. This feature may be useful when uploading a large number of maps, however be careful, with the number of parallel requests and the size of each zip file, as this may overload the server. Initial testing and gradually increasing the n_par parameter is therefore strongly recommended.

Value

a list with details on successfully processed maps, if usePar = TRUE, the node element of that list will be a data.table with all the successfully processed uploads.

Examples

```r
## Requires Survey Solutions Server and API credentials

# Upload a single zip file containing maps

suso_gql_uploadmap(endpoint = ep, user = usr, password = pass, workspace = ws, path_to_zip = "./dev/shapes_for_test.zip", usePar = FALSE)

# Upload a directory with zip files containing maps

suso_gql_uploadmap(endpoint = ep, user = usr, password = pass, workspace = ws, path_to_zip = "./dev/allzipforupload/", usePar = TRUE)
```
suso_gql_users

Get user information

Description

Allows the user to retrieve filtered or unfiltered user data.

Usage

suso_gql_users(
    endpoint = NULL,
    user = NULL,
    password = NULL,
    token = NULL,
    userName = NULL,
    fullName = NULL,
    isArchived = NULL,
    isLocked = NULL,
    creationDate = NULL,
    email = NULL,
    phoneNumber = NULL,
    id = NULL,
    role = NULL,
    sortby_userName = NULL,
    sortby_role = NULL,
    sortby_creationDate = NULL,
    take = NULL,
    skip = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>endpoint</td>
<td>GraphQL endpoint of your server</td>
</tr>
<tr>
<td>user</td>
<td>your API username</td>
</tr>
<tr>
<td>password</td>
<td>API password</td>
</tr>
<tr>
<td>token</td>
<td>If Survey Solutions server token is provided apiUser and apiPass will be ignored</td>
</tr>
<tr>
<td>userName</td>
<td>specific user name</td>
</tr>
<tr>
<td>fullName</td>
<td>fullName</td>
</tr>
<tr>
<td>isArchived</td>
<td>isArchived</td>
</tr>
<tr>
<td>isLocked</td>
<td>isLocked</td>
</tr>
<tr>
<td>creationDate</td>
<td>user creation date</td>
</tr>
<tr>
<td>email</td>
<td>user email</td>
</tr>
<tr>
<td>phoneNumber</td>
<td>user phoneNumber</td>
</tr>
<tr>
<td>id</td>
<td>user id</td>
</tr>
</tbody>
</table>
role user role
sortByUserName sort users by user name, either ASC for ascending or DESC for descending
sortByRole sort users by role, either ASC for ascending or DESC for descending
sortByCreationDate sort users by utc creation date, either ASC for ascending or DESC for descending
take take the specified integer number of maps
skip skip the first integer number of maps

Details

ATTENTION: This requires admin credentials, regular API credentials won’t work

Value

if successfull, returns a list with the (filtered) responses

Examples

## Requires Survey Solutions Server and ADMIN credentials!!

# Get all users without filter

suso_gql_users(endpoint = ep, user = adminuser, 
password = adminpass, sortByUserName = "ASC")

# Get only supervisors

suso_gql_users(endpoint = ep, user = adminuser, 
password = adminuser, , sortByUserName = "ASC", 
role = "SUPERVISOR")

transformers.numeric Utility Functions for numeric operator selection

Description

Can be used in filters ("where") for operator selection. If none is selected, operator always defaults to ‘eq()’. The functions below are valid for the corresponding inputs ComparableInt64OperationFilterInput and ComparableNullableOfInt32OperationFilterInput.
Usage

- `eq(value_set)`
- `neq(value_set)`
- `inbetw(value_set)`
- `ninbetw(value_set)`
- `gt(value_set)`
- `ngt(value_set)`
- `gte(value_set)`
- `ngte(value_set)`
- `lt(value_set)`
- `nlt(value_set)`
- `lte(value_set)`
- `nlte(value_set)`

Arguments

- `value_set` the parameter set for the operator

Details

Also see the `susooq_num` selector list, which allows you, to just select the function from a named list.

Value

a list with a single named element (operator name) to be handed over to the filter.

Functions

- `eq()`: equal
- `neq()`: not equal
- `inbetw()`: inbetween (requires numeric vector with upper and lower bound, i.e. c(1, 5))
- `ninbetw()`: not inbetween (requires numeric vector with upper and lower bound, i.e. c(1,5))
- `gt()`: greater than
- `ngt()`: not greater than
- `gte()`: greater than or equal
transformers.string

- `ngte()`: not greater than or equal
- `lt()`: lower than
- `nlt()`: not lower than
- `lte()`: lower than or equal
- `nlte()`: not lower than or equal

**Examples**

```r
# set filter equal to 0
eq(0)

# set filter to a value from 0 to 5
inbetw(c(0, 5))
```

---

**Utility Functions for string operator selection**

**Description**

Can be used in filters ("where") for operator selection. If none is selected, operator always defaults to ‘eq()’. The functions below are valid for the corresponding inputs `ComparableInt64OperationFilterInput` and `ComparableNullableOfInt32OperationFilterInput`.

**Usage**

```r
contains(value_set)
endsWith(value_set)
ncontains(value_set)
nendsWith(value_set)
nstartsWith(value_set)
startsWith(value_set)
inclu(value_set)
ninclu(value_set)
```

**Arguments**

- `value_set` the parameter set for the operator
Details

Also see the `susoop_str` selector list, which allows you, to just select the function from a named list.

Value

a list with a single named element (operator name) to be handed over to the filter.

Functions

- `contains()`: contains
- `endsWith()`: ends with
- `ncontains()`: not contains
- `nendsWith()`: not ends with
- `nstartsWith()`: not starts with
- `startsWith()`: starts with
- `inclu()`: in
- `ninclu()`: not in

Examples

```r
# set filter so that the string contains "area"
contains("area")

# set filter to string ending with .shp
endsWith(".shp")
```
Index

* datasets
  - susoop_num, 2
  - susoop_str, 3
  
  contains (transformers.string), 29
  
  endsWith (transformers.string), 29
  
  eq (transformers.numeric), 27
  
  gt (transformers.numeric), 27
  
  gte (transformers.numeric), 27
  
  inbetw (transformers.numeric), 27
  
  inclu (transformers.string), 29
  
  lt (transformers.numeric), 27
  
  lte (transformers.numeric), 27
  
  ncontains (transformers.string), 29
  
  nendsWith (transformers.string), 29
  
  neq (transformers.numeric), 27
  
  ngt (transformers.numeric), 27
  
  ngte (transformers.numeric), 27
  
  ninbetw (transformers.numeric), 27
  
  ninclu (transformers.string), 29
  
  nlt (transformers.numeric), 27
  
  nlte (transformers.numeric), 27
  
  nstartsWith (transformers.string), 29
  
  OlsonNames, 4, 5, 24
  
  startsWith (transformers.string), 29
  
  suso_gql_addassignmentcalendarevent, 4
  
  suso_gql_addinterviewcalendarevent, 5
  
  suso_gql_addusertomap, 6
  
  suso_gql_assignments, 7
  
  suso_gql_deletecalendarevent, 8
  
  suso_gql_deletemap, 9
  
  suso_gql_deleteuserfrommap, 10
  
  suso_gql_interviews, 11
  
  suso_gql_mapreport, 14
  
  suso_gql_maps, 16
  
  suso_gql_pwcheck, 18
  
  suso_gql_questionnaireitems, 19
  
  suso_gql_questionnaires, 20
  
  suso_gql_questions, 21
  
  suso_gql_updatecalendarevent, 23
  
  suso_gql_uploadmap, 24
  
  suso_gql_users, 26
  
  susoop_num, 2, 28
  
  susoop_str, 3, 30
  
  transformers.numeric, 27
  
  transformers.string, 29

31