TREX

Generated by Doxygen 1.8.17

Chapter 1

TREX

Command line tool for provisioning via xCat in compliance with the batch scheduler

1.1 Dependencies

- libcurl
- rapidjson

1.2 Compilation

```
./cmakeHelper.sh
cd cbuild
make -j $(nproc)
```

1.3 Doxygen

```
cd doc && doxygen Doxyfile
```

1.3.1 for HTML documentation:

doc/build/html/index.html

1.3.2 for PDF documentation:

```
make -C build/latex
```

generates doc/build/latex/refman.pdf

1.4 Usage

```
USAGE:
        ./TREX deploy <nodes> [--group] [--image <image>] [--prescripts <prescripts>]
               [--postbootscripts <postbootscripts>] [--postscripts <postscripts>] [--provmethod
               <provmethod>] [-h] [--json] [-b (slurm|pbs)] [-l <path>]
        ./TREX nodes [<nodes>] [-h] [--json] [-b (slurm|pbs)] [-l <path>]
        ./TREX state [<nodes>] [--state <state>] [--reason <reason>] [-h] [--json] [-b (slurm|pbs)]
               [-l <path>]
        ./TREX jobs [<jobIDs>] [-h] [--json] [-b (slurm|pbs)] [-l <path>]
        ./TREX queues [<queues>] [-h] [--json] [-b (slurm|pbs)] [-l <path>]
        ./TREX images [<images>] [-h] [--json] [-b (slurm|pbs)] [-l <path>]
        ./TREX bootstate [<nodes>] [-h] [--json] [-b (slurm|pbs)] [-l <path>]
        ./TREX reboot <nodes> [-h] [--json] [-b (slurm|pbs)] [-l <path>]
PARAMETERS:
        COMMANDS
            deploy <nodes> --group [--image <image>] [--prescripts <prescripts>] [--postbootscripts
            <postbootscripts>] [--postscripts <postscripts>] [--provmethod <provmethod>]
                    Deploy <image> on <nodes/groups>
            nodes <nodes>
                    Get node information [of <nodes>]
            state <nodes> [--state <state>] [--reason <reason>]
                    Get/Set state [of <nodes>]
            jobs <jobIDs>
                    Get job info [of <jobIDs>]
            queues <queues>
                    Get queue information [of <queues>]
            images <images>
                    Get information for available images [<images>]
            bootstate <nodes>
                    Get bootstate [of <nodes>]
            reboot <nodes>
                    Reboot <nodes>
        OPTIONS
            -h, --help
                    Shows this help message
            --json  Output as json
            -b, --batch (slurm|pbs)
                    Batch System
            -l, --loginFile <path>
                    Path for login data
```

Chapter 2

Third Party Dependencies

2.1 clipp - command line interfaces for modern C++

https://github.com/muellan/clipp; Copyright (c) 2017 André Müller; foss@andremueller-online.de

License: MIT

2.2 Licenses

2.2.1 MIT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 6

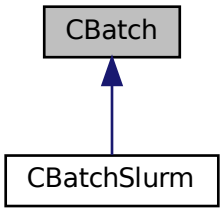
Class Documentation

6.1 CBatch Class Reference

Interface.

`#include <CBatch.h>`

Inheritance diagram for CBatch:



Protected Member Functions

- virtual int **get_jobs** (const std::vector< std::string > &, std::string &)=0
- virtual int **get_nodes** (const std::vector< std::string > &, std::string &)=0
- virtual int **get_queues** (const std::vector< std::string > &, std::string &)=0
- virtual int **get_node_states** (const std::vector< std::string > &, std::string &)=0
- virtual int **set_node_state** (const std::vector< std::string > &, std::string, std::string)=0

6.1.1 Detailed Description

Interface.

The documentation for this class was generated from the following files:

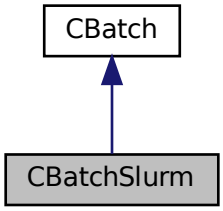
- /home/ntippman/projects/trex/src/batchExchange/CBatch.h
- /home/ntippman/projects/trex/src/batchExchange/CBatch.cpp

6.2 CBatchSlurm Class Reference

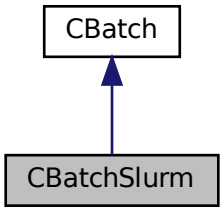
Slurm Class.

`#include <CBatchSlurm.h>`

Inheritance diagram for CBatchSlurm:



Collaboration diagram for CBatchSlurm:



Public Member Functions

- CBatchSlurm (std::string, std::string, std::string, std::string, bool)
  Constructor.
- virtual ~CBatchSlurm ()
  Destructor.
- int login ()
  Login session.
- int logout ()
  Logout session.
- int get_jobs (const std::vector< std::string > &, std::string &)
  Get job information.
- int get_nodes (const std::vector< std::string > &, std::string &)

Get node information.

- int get_queues (const std::vector< std::string > &, std::string &)

  Get queue information.
- int get_node_states (const std::vector< std::string > &, std::string &)

  Get node states.
- int set_node_state (const std::vector< std::string > &, std::string, std::string)

  Set node information.
- int **drain_nodes** (std::vector< std::string > &, const std::string &)
- int **drained** (std::vector< std::string > &, unsigned int &)

Additional Inherited Members

6.2.1 Detailed Description

Slurm Class.

6.2.2 Member Function Documentation

6.2.2.1 get_jobs()

```
int CBatchSlurm::get_jobs (
            const std::vector< std::string > & filter,
            std::string & output )  [virtual]
```

Get job information.

Parameters

jobs	Job Id(s)
output	output

Returns

0 Success

1 Error

Implements CBatch.

6.2.2.2 get_node_states()

```
int CBatchSlurm::get_node_states (
            const std::vector< std::string > & filter,
            std::string & output )  [virtual]
```

Get node states.

Parameters

filter	Nodes(s)
output	output

Returns

0 Success

1 Error

Implements CBatch.

6.2.2.3 get_nodes()

```
int CBatchSlurm::get_nodes (
            const std::vector< std::string > & filter,
            std::string & output )  [virtual]
```

Get node information.

Parameters

filter	Nodes(s)
output	output

Returns

0 Success

1 Error

Implements CBatch.

6.2.2.4 get_queues()

```
int CBatchSlurm::get_queues (
            const std::vector< std::string > & filter,
            std::string & output )  [virtual]
```

Get queue information.

Parameters

filter	Queue(s)
output	output

Returns

0 Success

1 Error

Implements CBatch.

6.2.2.5 login()

```
int CBatchSlurm::login ( )
```

Login session.

Returns

0 Success

1 Error

6.2.2.6 logout()

```
int CBatchSlurm::logout ( )
```

Logout session.

Returns

0 Success

1 Error

6.2.2.7 set_node_state()

```
int CBatchSlurm::set_node_state (
            const std::vector< std::string > & nodes,
            std::string state,
            std::string reason )  [virtual]
```

Set node information.

Parameters

queues	nodes list of nodes
state	State to be set
reason	Reason for state change

Returns

0 Success

1 Error

Implements CBatch.

The documentation for this class was generated from the following files:

- /home/ntippman/projects/trex/src/batchExchange/CBatchSlurm.h
- /home/ntippman/projects/trex/src/batchExchange/CBatchSlurm.cpp

6.3 CXCat Class Reference

Public Member Functions

- CXCat (std::string, std::string, std::string, std::string, bool)
  Constructor.
- ~CXCat ()
  Destructor.
- int login ()
  Login session.
- int logout ()
  Logout session.
- int get_os_images (const std::vector< std::string > &, std::string &)
  Get os images.
- int get_os_image_names (std::vector< std::string > &)
  Get names of os images.
- int get_bootstate (const std::vector< std::string > &, std::string &)
  Get os images.
- int get_nodes (std::string &)
  Get information of all available nodes.
- int set_os_image (const std::vector< std::string > &, std::string)
  Set os image.
- int reboot_nodes (const std::vector< std::string > &)
  Reboot nodes.
- int set_group_attributes (std::string, const std::string &)
  Set attributes of group.
- int set_node_attributes (const std::vector< std::string > &, const std::string &)
  Set attributes of nodes.
- int get_group_members (std::string group, std::vector< std::string > &output)
  Get all members of group.
- int get_group_names (std::vector< std::string > &)
  Get names of all groups.
- int get_group (std::string, std::string &)
  Get attributes of group.

6.3.1 Member Function Documentation

6.3.1.1 get_bootstate()

```
int CXCat::get_bootstate (
            const std::vector< std::string > & filter,
            std::string & output )
```

Get os images.

Parameters

filter	filter
output	output

Returns

0 Success

1 Error

6.3.1.2 get_group()

```
int CXCat::get_group (
            std::string group,
            std::string & output )
```

Get attributes of group.

Parameters

group	group
output	output

Returns

0 Success

1 Error

6.3.1.3 get_group_members()

```
int CXCat::get_group_members (
            std::string group,
            std::vector< std::string > & output )
```

Get all members of group.

Parameters

group	group
output	output

Returns

0 Success

1 Error

6.3.1.4 get_group_names()

```
int CXCat::get_group_names (
            std::vector< std::string > & output )
```

Get names of all groups.

Parameters

output	output

Returns

0 Success

1 Error

6.3.1.5 get_nodes()

```
int CXCat::get_nodes (
            std::string & output )
```

Get information of all available nodes.

Parameters

output	output

Returns

0 Success

1 Error

6.3.1.6 get_os_image_names()

```
int CXCat::get_os_image_names (
            std::vector< std::string > & output )
```

Get names of os images.

Parameters

output	output

Returns

0 Success

1 Error

6.3.1.7 get_os_images()

```
int CXCat::get_os_images (
            const std::vector< std::string > & filter,
            std::string & output )
```

Get os images.

Parameters

filter	filter
output	output

Returns

0 Success

1 Error

6.3.1.8 login()

```
int CXCat::login ( )
```

Login session.

Returns

0 Success

1 Error

6.3.1.9 logout()

```
int CXCat::logout ( )
```

Logout session.

Returns

0 Success

1 Error

6.3.1.10 reboot_nodes()

```
int CXCat::reboot_nodes (
            const std::vector< std::string > & filter )
```

Reboot nodes.

Parameters

filter	filter

Returns

0 Success

1 Error

6.3.1.11 set_group_attributes()

```
int CXCat::set_group_attributes (
            std::string group,
            const std::string & attributes )
```

Set attributes of group.

Parameters

group	name of group
attributes	json attributes

Returns

0 Success

1 Error

6.3.1.12 set_node_attributes()

```
int CXCat::set_node_attributes (
            const std::vector< std::string > & nodes,
            const std::string & attributes )
```

Set attributes of nodes.

Parameters

nodes	list of nodes
attributes	json attributes

Returns

0 Success

1 Error

6.3.1.13 set_os_image()

```
int CXCat::set_os_image (
            const std::vector< std::string > & filter,
            std::string osImage )
```

Set os image.

Parameters

filter	filter
output	output

Returns

0 Success

1 Error

The documentation for this class was generated from the following files:

- /home/ntippman/projects/trex/src/batchExchange/CXCat.h
- /home/ntippman/projects/trex/src/batchExchange/CXCat.cpp

6.4 HttpSession Class Reference

Public Member Functions

- HttpSession (const std::string &usernameIn, const std::string &passwordIn, const std::string &server↩ AddressIn, const std::string &serverPortIn)

  Constructor.
- ~HttpSession ()

  Destructor.
- void ssl_verify (bool verfiy)

  Simple setter if ssl certificate should be checked or not.
- int set_token_type (int tokenTypeIn, const SessionTokenTypes *sessionTokenTypes=nullptr)

  Sets keys to extract session token more easyly.
- void set_date_parse_descr (const std::string formatDesc)

  Setter for data format.
- void set_login_path (const std::string loginPathIn)

  Defines the URL path for login.
- void set_logout_path (const std::string logoutPathIn)

  Defines the URL path for logout.
- std::string get_access_token ()

  Get access token.
- int login ()

  Gets authentification token.
- int logout ()

  Frees authentification token.

6.4.1 Constructor & Destructor Documentation

6.4.1.1 HttpSession()

```
HttpSession::HttpSession (
            const std::string & usernameIn,
            const std::string & passwordIn,
            const std::string & serverAddressIn,
            const std::string & serverPortIn )
```

Constructor.

Constructor with necessary parameters for function. Initililizes libcurl handle too.

Parameters

usernameIn	login name of user
passwordIn	login password of user
server↩ AddressIn	url of server
serverPortIn	port of server

6.4.1.2 ∼HttpSession()

```
HttpSession::∼HttpSession ( )
```

Destructor.

Frees session if not done yet and libcurl ressources.

6.4.2 Member Function Documentation

6.4.2.1 get_access_token()

```
std::string HttpSession::get_access_token ( )
```

Get access token.

Returns

the access token itself

6.4.2.2 login()

```
int HttpSession::login ( )
```

Gets authentification token.

Returns

0-100 error codes from libcurl >100 http error codes -1 no libcurl handle -2 no username or password -3 no host address or port -4 could not receive access token -5 already logged in -10 could'n extract token -100 unknown session type

Parameters

loginPathIn	REST path to login endpoint (leading / is needed e.g. "/token")

6.4.2.3 logout()

```
int HttpSession::logout ( )
```

Frees authentification token.

Returns

0-100 error codes from libcurl >100 http error codes

6.4.2.4 set_date_parse_descr()

```
void HttpSession::set_date_parse_descr (
            const std::string formatDesc )
```

Setter for data format.

Must have a format like it is described in strtime. https://www.cplusplus.com/reference/ctime/strftime/

Parameters

formatDesc	strtime like date description

6.4.2.5 set_login_path()

```
void HttpSession::set_login_path (
            const std::string loginPathIn )
```

Defines the URL path for login.

Parameters

login↩ PathIn	URL path (leading / is needed)

6.4.2.6 set_logout_path()

```
void HttpSession::set_logout_path (
            const std::string logoutPathIn )
```

Defines the URL path for logout.

Parameters

logout↩ PathIn	URL path (leading / is needed)

6.4.2.7 set_token_type()

```
int HttpSession::set_token_type (
            int tokenTypeIn,
            const SessionTokenTypes * sessionTokenTypes = nullptr )
```

Sets keys to extract session token more easyly.

Parameters

tokenTypeIn	defines which type of token should be used
sessionTokenTypes	struct with json keys for token attributes

6.4.2.8 ssl_verify()

```
void HttpSession::ssl_verify (
            bool verfiy )
```

Simple setter if ssl certificate should be checked or not.

Parameters

verfiy	true if ssl certificate should be checked

The documentation for this class was generated from the following files:

- /home/ntippman/projects/trex/src/batchExchange/httpSession.h
- /home/ntippman/projects/trex/src/batchExchange/httpSession.cpp

6.5 utils::loginData Struct Reference

Public Attributes

- std::string **username**
- std::string **password**
- std::string **host**
- std::string **port**

The documentation for this struct was generated from the following file:

- /home/ntippman/projects/trex/src/batchExchange/utils.h

6.6 RestClient Class Reference

Public Member Functions

- RestClient (const int)
  Constructor.
- ~RestClient ()
  Destructor.
- int login ()
  Loggs in if needed.
- int logout ()
  Removes login tokens and frees sessions.
- void set_user_credentials (const std::string &, const std::string &)
  Simple setter for username an password.
- void set_host_config (const std::string &, const std::string &)
  Simple setter for server settings.
- void ssl_verify (bool)
  Simple setter if ssl certificate should be checked or not.
- void useragent (const std::string &)
  Simple setter user agent.
- int get (const std::string &, std::string &, std::string &)
  Represents http get request.
- int post (const std::string &, const std::string &, std::string &, std::string &)
  Represents http post request.
- int del (const std::string &, std::string &, std::string &)
  Represents http delete request.
- int patch (const std::string &, const std::string &, std::string &, std::string &)
  Represents http patch request.
- int put (const std::string &, const std::string &, std::string &, std::string &)
  Represents http put request.
- int **call** (std::string, const std::string &, std::string &, const std::string &="")
- long get_last_http_code ()
  Getter for http return code of last request.
- double get_last_request_time ()
  Getter for execution time of last request.
- std::string get_last_url ()
  Getter for url of last request.

6.6.1 Constructor & Destructor Documentation

6.6.1.1 RestClient()

```
RestClient::RestClient (
            const int authTypeIn )
```

Constructor.

Constructor with necessary parameters for function. Initililizes libcurl handle too.

Parameters

auth↩ TypeIn	type of http connection (e.g. no login, token based, ...)

6.6.1.2 ~RestClient()

```
RestClient::~RestClient ( )
```

Destructor.

Frees session if not done yet and libcurl ressources.

6.6.2 Member Function Documentation

6.6.2.1 del()

```
int RestClient::del (
            const std::string & restPath,
            std::string & response,
            std::string & header )
```

Represents http delete request.

Returns

0-100 error codes from libcurl >100 http error codes -1 no libcurl handle

6.6.2.2 get()

```
int RestClient::get (
            const std::string & restPath,
            std::string & response,
            std::string & header )
```

Represents http get request.

Returns

0-100 error codes from libcurl >100 http error codes -1 no libcurl handle

6.6.2.3 get_last_http_code()

```
long RestClient::get_last_http_code ( )
```

Getter for http return code of last request.

Returns

code of last request

6.6.2.4 get_last_request_time()

```
double RestClient::get_last_request_time ( )
```

Getter for execution time of last request.

Returns

execution time of last request

6.6.2.5 get_last_url()

```
std::string RestClient::get_last_url ( )
```

Getter for url of last request.

Returns

url of last request

6.6.2.6 login()

```
int RestClient::login ( )
```

Loggs in if needed.

Returns

0-100 error codes from libcurl >100 http error codes -1 no libcurl handle -2 no username or password -3 no host address or port -4 could not receive access token -5 already logged in -100 unknown authentication method

6.6.2.7 patch()

```
int RestClient::patch (
            const std::string & restPath,
            const std::string & postData,
            std::string & response,
            std::string & header )
```

Represents http patch request.

Returns

0-100 error codes from libcurl >100 http error codes -1 no libcurl handle

6.6.2.8 post()

```
int RestClient::post (
            const std::string & restPath,
            const std::string & postData,
            std::string & response,
            std::string & header )
```

Represents http post request.

Returns

0-100 error codes from libcurl >100 http error codes -1 no libcurl handle

6.6.2.9 put()

```
int RestClient::put (
            const std::string & restPath,
            const std::string & postData,
            std::string & response,
            std::string & header )
```

Represents http put request.

Returns

0-100 error codes from libcurl >100 http error codes -1 no libcurl handle

6.6.2.10 set_host_config()

```
void RestClient::set_host_config (
            const std::string & serverAddressIn,
            const std::string & serverPortIn )
```

Simple setter for server settings.

Parameters

serverAddressIn	server address or dns name
serverPortIn	port to communicate with (e.g. 80 for http, 443 for https)

6.6.2.11 set_user_credentials()

```
void RestClient::set_user_credentials (
            const std::string & usernameIn,
            const std::string & passwordIn )
```

Simple setter for username an password.

Parameters

usernameIn	username for login
passwordIn	password for login

6.6.2.12 ssl_verify()

```
void RestClient::ssl_verify (
            bool sslVerifyIn )
```

Simple setter if ssl certificate should be checked or not.

Parameters

sslVerifyIn	true if ssl certificate should be checked

6.6.2.13 useragent()

```
void RestClient::useragent (
            const std::string & useragent )
```

Simple setter user agent.

Sets the name how libcurl authenticated against the server.

Parameters

useragent	authentication name

The documentation for this class was generated from the following files:

- /home/ntippman/projects/trex/src/batchExchange/restClient.h
- /home/ntippman/projects/trex/src/batchExchange/restClient.cpp

6.7 SessionToken Class Reference

Public Member Functions

- SessionToken ()
  Constructor.
- ~SessionToken ()
  Destructor.
- void set_keys_token_type (const std::vector< std::string > &keysTokenTypeIn)
  Set token type.
- void set_keys_access_token (const std::vector< std::string > &keysAccessTokenIn)
  Set access token.
- void set_keys_refresh_token (const std::vector< std::string > &keysRefreshTokenIn)
  Set refresh token.
- void set_keys_expire_time (const std::vector< std::string > &keysExpireTimeIn)
  Set expire time.
- void set_keys_expire_date (const std::vector< std::string > &keysExpireDateIn)
  Set expire date.
- void set_keys_by_token_type (const SessionTokenTypes token)
  Sets keys to extract session token more easyly.
- void set_date_parse_descr (std::string formatDesc)
  Setter for data format.
- int read_token (const std::string &token)
  Wrapper for reading token data from json object.
- bool token_expired ()
  Resturns if token is expired.
- int token_time_to_expire ()
  Delivers time in seconds when how long the token is valid.
- std::string get_token_type ()
  Get token type.
- std::string get_access_token ()
  Get access token.
- std::string get_refresh_token ()
  Get refresh token.
- int get_expire_time ()
  Get expire time of token.
- time_t get_expire_date ()
  Get expire date.

6.7.1 Member Function Documentation

6.7.1.1 get_access_token()

```
std::string SessionToken::get_access_token ( )
```

Get access token.

Returns

access token

6.7.1.2 get_expire_date()

```
time_t SessionToken::get_expire_date ( )
```

Get expire date.

Returns

expire date of the access token as unix timestamp

6.7.1.3 get_expire_time()

```
int SessionToken::get_expire_time ( )
```

Get expire time of token.

Returns

expire time of the access token in seconds

6.7.1.4 get_refresh_token()

```
std::string SessionToken::get_refresh_token ( )
```

Get refresh token.

Returns

a refresh token to request a new access token

6.7.1.5 get_token_type()

```
std::string SessionToken::get_token_type ( )
```

Get token type.

Returns

Type of the token

6.7.1.6 read_token()

```
int SessionToken::read_token (
            const std::string & token )
```

Wrapper for reading token data from json object.

Parameters

token	the token data as json containted in a string

Returns

error code

6.7.1.7 set_date_parse_descr()

```
void SessionToken::set_date_parse_descr (
            std::string formatDesc )
```

Setter for data format.

Must have a format like it is described in strtime. https://www.cplusplus.com/reference/ctime/strftime/

Parameters

formatDesc	strtime like date description

6.7.1.8 set_keys_access_token()

```
void SessionToken::set_keys_access_token (
            const std::vector< std::string > & keysAccessTokenIn )
```

Set access token.

Parameters

keysAccess↩ TokenIn	contains keys of json file where to find the access token ["token", "accessToken"]

6.7.1.9 set_keys_by_token_type()

```
void SessionToken::set_keys_by_token_type (
            const SessionTokenTypes token )
```

Sets keys to extract session token more easyly.

Parameters

token	struct with json keys for token attributes

6.7.1.10 set_keys_expire_date()

```
void SessionToken::set_keys_expire_date (
            const std::vector< std::string > & keysExpireDateIn )
```

Set expire date.

Parameters

keysExpire↩ DateIn	contains keys of json file where to find the expire date ["token", "date"]

6.7.1.11 set_keys_expire_time()

```
void SessionToken::set_keys_expire_time (
            const std::vector< std::string > & keysExpireTimeIn )
```

Set expire time.

Parameters

keysExpire↩ TimeIn	contains keys of json file where to find the expire time ["token", "time"]

6.7.1.12 set_keys_refresh_token()

```
void SessionToken::set_keys_refresh_token (
            const std::vector< std::string > & keysRefreshTokenIn )
```

Set refresh token.

Parameters

keysRefresh↩ TokenIn	contains keys of json file where to find the refresh token ["token", "refreshToken"]

6.7.1.13 set_keys_token_type()

```
void SessionToken::set_keys_token_type (
            const std::vector< std::string > & keysTokenTypeIn )
```

Set token type.

Parameters

keysToken↩ TypeIn	contains keys of json file where to find the token type ["token", "tokenType"]

6.7.1.14 token_expired()

```
bool SessionToken::token_expired ( )
```

Resturns if token is expired.

Returns

simple bool true == expired

6.7.1.15 token_time_to_expire()

```
int SessionToken::token_time_to_expire ( )
```

Delivers time in seconds when how long the token is valid.

Returns

time in seconds

The documentation for this class was generated from the following files:

- /home/ntippman/projects/trex/src/batchExchange/httpSession.h
- /home/ntippman/projects/trex/src/batchExchange/httpSession.cpp

6.8 SessionTokenTypes Struct Reference

Public Attributes

- std::vector< std::string > **keysTokenType**
- std::vector< std::string > **keysAccessToken**
- std::vector< std::string > **keysRefreshToken**
- std::vector< std::string > **keysExpireTime**
- std::vector< std::string > **keysExpireDate**

The documentation for this struct was generated from the following file:

- /home/ntippman/projects/trex/src/batchExchange/sessionTokenTypes.h

Chapter 7

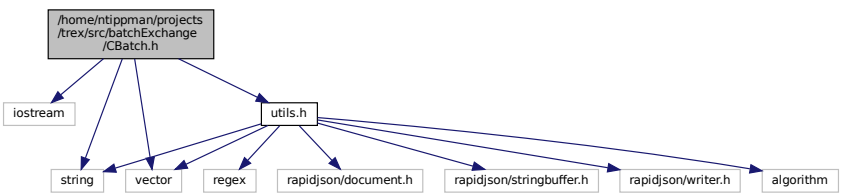
File Documentation

7.1 /home/ntippman/projects/trex/src/batchExchange/CBatch.h File Reference

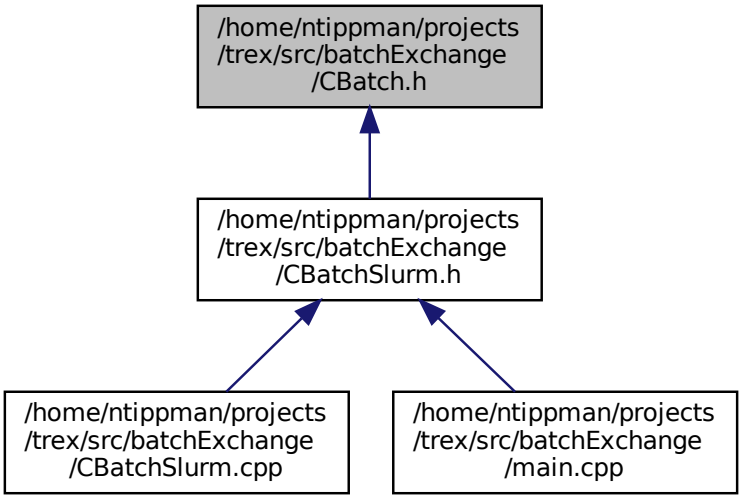
Definitions for CBatch.h.

```
#include <iostream>
#include <string>
#include <vector>
#include "utils.h"
```

Include dependency graph for CBatch.h:



This graph shows which files directly or indirectly include this file:



Classes

- class CBatch

  Interface.

7.1.1 Detailed Description

Definitions for CBatch.h.

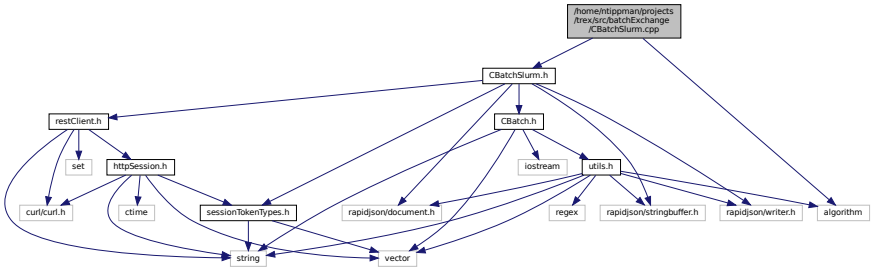
Interface definition for inheriting batch classes.

7.2 /home/ntippman/projects/trex/src/batchExchange/CBatchSlurm.cpp File Reference

CBatch Slurm implementation.

```
#include "CBatchSlurm.h"
#include <algorithm>
```

Include dependency graph for CBatchSlurm.cpp:



7.2.1 Detailed Description

CBatch Slurm implementation.

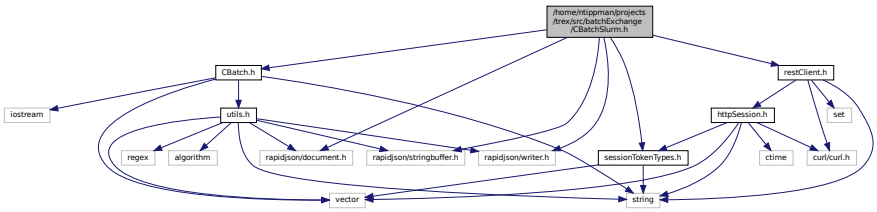
Slurm currently does not provide a way to query multiple specified resources with a single call. Therefore the whole collection is queried and then filtered.

7.3 /home/ntippman/projects/trex/src/batchExchange/CBatchSlurm.h File Reference

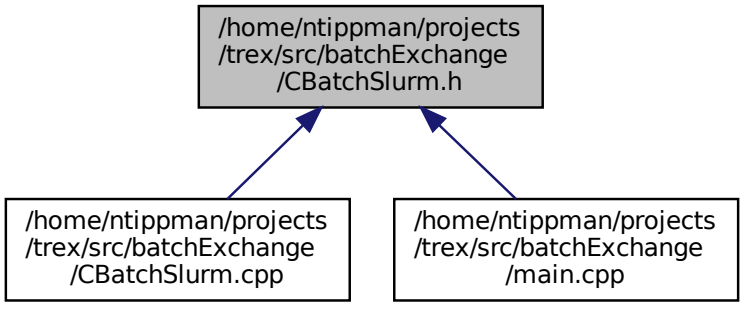
Header for CBatchSlurm.cpp.

```
#include "CBatch.h"
#include "rapidjson/document.h"
#include "rapidjson/stringbuffer.h"
#include "rapidjson/writer.h"
#include "restClient.h"
#include "sessionTokenTypes.h"
```

Include dependency graph for CBatchSlurm.h:



This graph shows which files directly or indirectly include this file:



Classes

- class CBatchSlurm

  Slurm Class.

7.3.1 Detailed Description

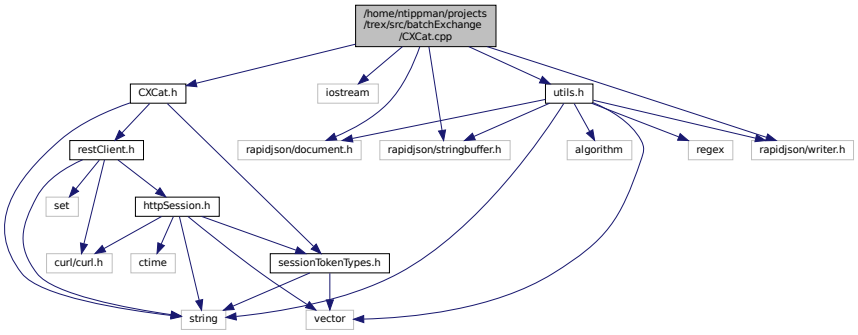
Header for CBatchSlurm.cpp.

7.4 /home/ntippman/projects/trex/src/batchExchange/CXCat.cpp File Reference

CXCat implementation.

```
#include "CXCat.h"
#include <iostream>
#include "rapidjson/document.h"
#include "rapidjson/stringbuffer.h"
#include "rapidjson/writer.h"
#include "utils.h"
```

Include dependency graph for CXCat.cpp:



7.4.1 Detailed Description

CXCat implementation.

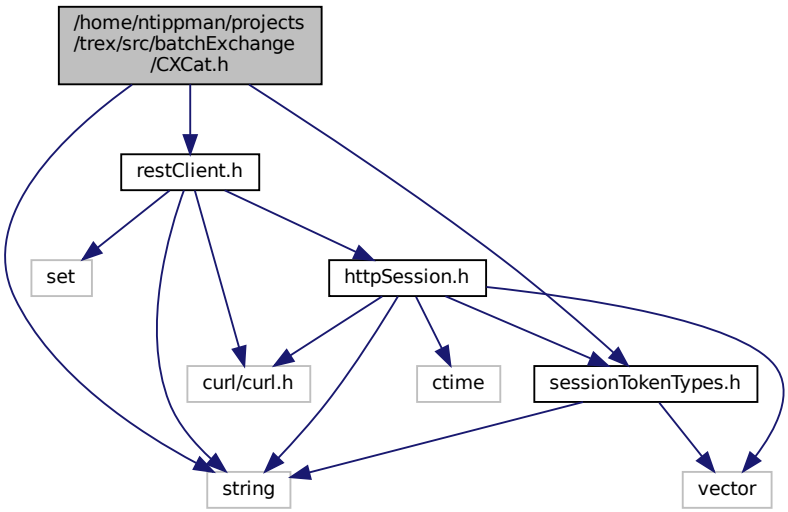
7.5 /home/ntippman/projects/trex/src/batchExchange/CXCat.h File Reference

Header for CXCat.

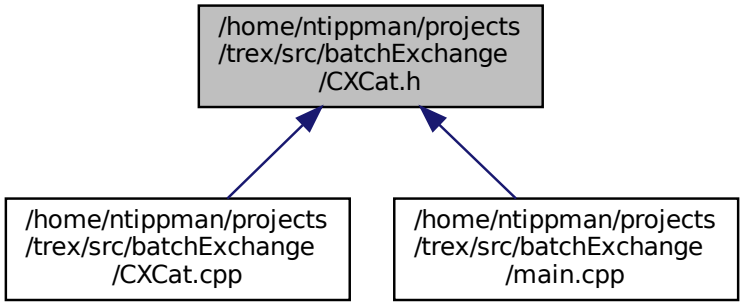
```
#include <string>
#include "restClient.h"
```

```
#include "sessionTokenTypes.h"
```

Include dependency graph for CXCat.h:



This graph shows which files directly or indirectly include this file:



Classes

- class CXCat

7.5.1 Detailed Description

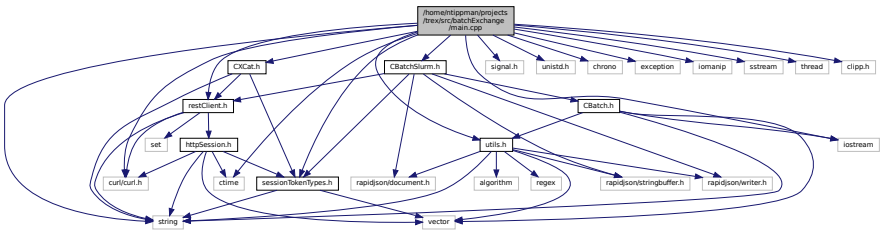
Header for CXCat.

7.6 /home/ntippman/projects/trex/src/batchExchange/main.cpp File Reference

CLI.

```
#include <curl/curl.h>
#include <signal.h>
#include <unistd.h>
#include <chrono>
#include <ctime>
#include <exception>
#include <iomanip>
#include <iostream>
#include <sstream>
#include <string>
#include <thread>
#include "CBatchSlurm.h"
#include "CXCat.h"
#include "clipp.h"
#include "restClient.h"
#include "sessionTokenTypes.h"
#include "utils.h"
```

Include dependency graph for main.cpp:



Macros

- #define **DRAIN_SLEEP** 3000

Functions

- bool **canceled** (false)
- void sigHandler (int signal)

  Handle caught signal.
- int **main** (int argc, char **argv)

7.6.1 Detailed Description

CLI.

7.6.2 Function Documentation

7.6.2.1 sigHandler()

```
void sigHandler (
            int signal )
```

Handle caught signal.

This function is called when SIGINT is caught.

Parameters

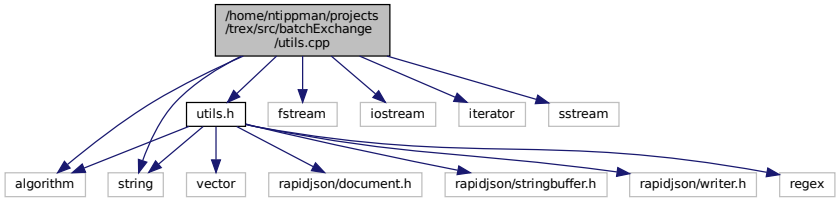
signal	Number of signal

7.7 /home/ntippman/projects/trex/src/batchExchange/utils.cpp File Reference

Collection of helper functions.

```
#include "utils.h"
#include <algorithm>
#include <fstream>
#include <iostream>
#include <iterator>
#include <sstream>
#include <string>
```

Include dependency graph for utils.cpp:



7.7.1 Detailed Description

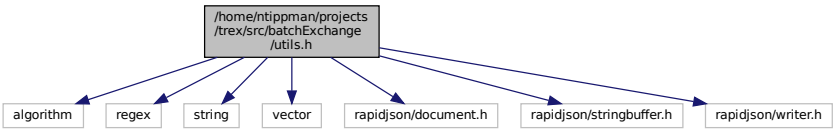
Collection of helper functions.

7.8 /home/ntippman/projects/trex/src/batchExchange/utils.h File Reference

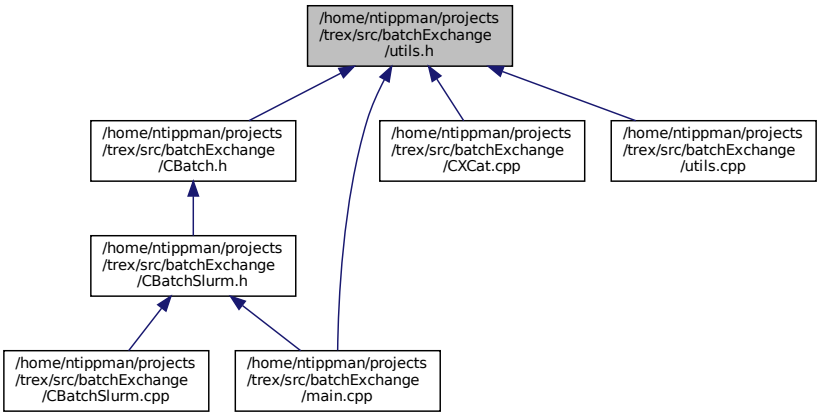
Header for utils.cpp.

```
#include <algorithm>
#include <regex>
#include <string>
#include <vector>
#include "rapidjson/document.h"
#include "rapidjson/stringbuffer.h"
#include "rapidjson/writer.h"
```

Include dependency graph for utils.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct utils::loginData

Macros

- #define **INVALID_JSON_ERROR_MSG** "Error - Response is not valid JSON"

Functions

- void utils::str_split (const std::string &input, const std::string delimiter, std::vector< std::string > &ret)

  Split string at delimiter.
- int **utils::read_login_data** (const std::string &path, utils::loginData &megware, utils::loginData &xcat, utils::loginData &slurm, bool ignoreHeader=true)
- int **utils::read_file_to_string** (std::string, std::string &)
- int **utils::write_to_file** (std::string, const std::string &)
- void **utils::erase_lines_from_start** (std::string &, int)
- void **utils::to_lower** (std::string &)
- void **utils::to_upper** (std::string &)
- std::string **utils::join_vector_to_string** (const std::vector< std::string > &, const std::string)
- int **utils::check_errors** (const std::string &)
- void utils::str_extract_regex_occurances (std::string, const std::regex &, std::vector< std::string > &)

  Extract all regex matches from string.
- bool utils::is_number (const std::string &)

  Check if string contains only numbers (integer only)
- void utils::decode_brace (const std::string &, std::vector< std::string > &)

  Decode brace notation.
- bool utils::str_match_wildcard (const std::string &, const std::string &)

  Check if string matches wildcard-string.
- bool utils::ends_with (const std::string &, const std::string &)

  Check if string ends with <suffix>
- bool utils::starts_with (const std::string &, const std::string &)

  Check if string starts with <prefix>
- bool utils::str_match_any_wildcard (const std::vector< std::string > &, const std::string &)

  Check if string matches any wildcard-string of vector.
- void **utils::rapidjson_doc_to_str** (rapidjson::Document &, std::string &)
- template<typename T >
  bool utils::vector_contains (const std::vector< T > &v, const T &e)

  Template function to check whether a vector contains the specified element.

Variables

- const std::vector< std::string > **utils::slurmNodeStates**

7.8.1 Detailed Description

Header for utils.cpp.

7.8.2 Function Documentation

7.8.2.1 decode_brace()

```
void utils::decode_brace (
            const std::string & input,
            std::vector< std::string > & ret )
```

Decode brace notation.

Parameters

input	String to be decoded
ret	Vector of decoded strings

7.8.2.2 ends_with()

```
bool utils::ends_with (
            const std::string & s,
            const std::string & suffix )
```

Check if string ends with <suffix>

Parameters

s	Input string
suffix	Suffix

Returns

0 Suffix not present

1 Suffix present

7.8.2.3 is_number()

```
bool utils::is_number (
            const std::string & s )
```

Check if string contains only numbers (integer only)

Parameters

pattern	Wildcard pattern vector
target	Target to be checked

Returns

0 NaN

1 Is number

7.8.2.4 starts_with()

```
bool utils::starts_with (
            const std::string & s,
            const std::string & prefix )
```

Check if string starts with <prefix>

Parameters

s	Input string
prefix	Prefix

Returns

0 Prefix not present

1 Prefix present

7.8.2.5 str_extract_regex_occurances()

```
void utils::str_extract_regex_occurances (
            std::string input,
            const std::regex & regex,
            std::vector< std::string > & ret )
```

Extract all regex matches from string.

Parameters

input	Input string to be checked
regex	Regex
ret	Vector of matches

7.8.2.6 str_match_any_wildcard()

```
bool utils::str_match_any_wildcard (
            const std::vector< std::string > & wildcards,
            const std::string & target )
```

Check if string matches any wildcard-string of vector.

Parameters

pattern	Wildcard pattern vector
target	Target to be checked

Returns

0 No match

1 Match

7.8.2.7 str_match_wildcard()

```
bool utils::str_match_wildcard (
            const std::string & pattern,
            const std::string & target )
```

Check if string matches wildcard-string.

Parameters

pattern	Wildcard pattern
target	Target to be checked

Returns

0 No match

1 Match

7.8.2.8 str_split()

```
void utils::str_split (
            const std::string & input,
            const std::string delimiter,
            std::vector< std::string > & ret )
```

Split string at delimiter.

Parameters

input	Input string
delimiter	Delimiter
ret	Output vector

7.8.2.9 vector_contains()

```
template<typename T >
bool utils::vector_contains (
```

```
const std::vector< T > & v,
const T & e )
```

Template function to check whether a vector contains the specified element.

Parameters

v	Vector
e	Element

Returns

0 Element not in vector

1 Element in vector

Index