1 NAME

bsclient - simple interactive FTP-like client for BaseSpace data downloads

2 VERSION

This documentation refers to bsclient v0.4.1a

3 SYNOPSIS

bsclient [options]

4 DESCRIPTION

bsclient is a simple interactive shell for Illumina BaseSpace files/resources. It supports a limited set of FTP-like commands for navigating and downloading files from a BaseSpace account. The user account and privileges are encoded in an 'access token' which must be obtained from the BaseSpace Developers site (see AUTHENTICATION). **bsclient** treats BaseSpace data "levels" as pseudo-directories in order to provide an FTP-like experience. See COMMANDS for further details.

5 OPTIONS

--overwrite

Force overwrite of existing files. Default is false (existing files are skipped).

--save_token

Attempt to save encrypted access token to file. This is only honored if an existing .bsclient_token file is not found and the user has been prompted for the access token. User will be prompted for an encryption key which is used for subsequent sessions (easier than remembering a 32-character hex string!).

--timeout

Set the network timeout in seconds (default: 60). Increase this value if downloads are failing mid-way through.

--skip_verify

Skip verification of the SSL certificate validity. This does not disable encryption on the connection, but simply turns off the requirement that the certificate be matched against the database stored on the computer. Some users may need to set this flag if receiving errors about failed SSL connections.

--help

Show documentation and exit

--version

Print version string and exit

6 AUTHENTICATION

Authentication with BaseSpace is done via tokens. Therefore, in order to obtain full access to your BaseSpace account using this tool, you will need to obtain an access token with global rights to your account. Currently, this can only be done by signing up for a BaseSpace developer's account (free and easy to do). Perhaps in the future BaseSpace will allow creation of tokens with global read-only permissions for regular users, but this is currently not possible to my knowledge. This token can grant full access to your account for anyone who discovers it, so treat it like you would any other important password or passphrase.

Once you have obtained an access token (a long hexidecimal string) you can use it with **bsclient** to connect to your account. If **bsclient** does not find any stored tokens, it will prompt you to enter your access token when started. It will not, by default, store this token to disk in order to prevent access to the token from others on a public or shared machine. If you are on a private, secure computer and want to store the token to disk for future use, specify the '--save_token' flag on the command line. In this case, **bsclient** will ask for a separate passphrase to use for symmetric encryption of the access token before writing it to your home directory. This prevents casual discovery of the token should anyone gain access to your home files, with the advantage that is is probably easier to remember a passphrase of your choosing than a 32-character hexadecimal string. If your access token should change for any reason, simply delete the file at $HOME/.bsclient_token and repeat the above procedure.

7 COMMANDS

ls

List items in current level. Information displayed is pipe-separated and varies depending on the current level, as follows:

Projects: project | <ID> | <Name>

Samples: sample | <ID> | <Name> | <library_type> | <read_lens> | <passed_read_count>

Files: file | <ID> | <Name> | <size_in_bytes>

cd ***target***

Change level. Currently there is a three-level hierarchy: Projects->Samples->Files. Works with tab-autocompletion (either by ID or name), and '..' goes up one level as expected. Currently, only a single level change is supported per command (e.g. '../..' will not work).

get ***target***

Download file(s). If a file name/ID is given, will download a single file. If a sample or project name/ID is given, will recursively download all files in that sample/project within a corresponding directory structure. If a file exists and '--overwrite' was not specified, it will be skipped.

quit

Self-explanatory

8 CAVEATS AND BUGS

This software has been tested on Linux and, to a lesser extent, on Windows 8.1. While it should also work on Mac OSX, BSD, and other systems with the appropriate modules installed, it has not been tested on these systems. Input from users on these systems would be appreciated.

Please report all bugs or suggestions by email to the author below.

9 AUTHOR

Jeremy Volkening

10 COPYRIGHT AND LICENSE

Copyright 2014-2015 Jeremy Volkening <jeremy@base2bio.com>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.