Data Delivery System

SciLifeLab Data Centre

Mar 03, 2022

CONTENTS

Note: During testing on 2022-02-28 to 2022-03-07, this will work as **both documentation and as a test protocol**. The goal is to give you ideas on what aspects of the system to try out, but please also test anything else you can think of (e.g. if you feel we have missed something). Also, please make note of whether the documentation of the system has any deficiencies such as unclear, misleading or incomplete instructions.

Warning: Please do not use any sensitive data during this testing phase. Since the last testing period we have implemented a key management system for the storage of all required keys (e.g. encryption keys) and your data should be secure, however there may be bugs and/or issues which we need to solve before DDS is put into production.

The Data Delivery System (DDS) consists of a command line interface (CLI) and a very minimal web interface. The web interface will be improved on as soon as possible, but we have decided that having a working CLI and its corresponding API is highest on the priority list.

CHAPTER

ONE

HOW WILL I GET MY USER ACCOUNT?

The testing will begin with you getting an invite via email. The email will be from *dsw@scilifelab.se* during the testing (note that any emails sent to this address regarding the DDS *will not be responded to*). If you do not get an email, please have a look in the junk/spam folder. If it's not there either, please contact Ina (*email:* ina.oden.osterbo@scilifelab.uu.se, *slack:* Ina Odén Österbo) and we will look into it.

Once you get the invitation email, follow the link in the email and register your account. After this, you should have access to the system and can begin testing the different features. To be able to test the CLI (which contains most of the functionality) please follow the installation guide *below*.

Warning: Forgetting passwords in the DDS means that you will lose access to all project data. We highly recommend that you use a password management system such as LastPass or similar.

When resetting a password you can, of course, regain access to the projects you lost access to. This procedure is explained *here*.

Your account will be either a *Unit Admin*, *Unit Personnel* or a *Researcher* account. These are called the different roles which define the commands and actions you are allowed to perform in the DDS, including some administrative permissions. The roles are defined on this board.

CHAPTER

TWO

HOW TO USE THE DDS CLI

2.1 Installation

At this time, a **Python** and **pip** installation is required for the `dds-cli` installation to work. We are currently working on an executable which can install all the requirements and the CLI for you. The goal is for this executable to be ready when we release the CLI into production.

2.1.1 Uppmax

We are currently checking that installing the DDS CLI on Rackham works as expected. We will update this information as soon as possible. If we have not yet updated this section before you attempt to use the DDS CLI on **Uppmax Rackham**, feel free to try it and inform us on any issues.

Regarding Bianca: Uppmax has offered to help us out with testing that the connection to Bianca works. Instructions for this will therefore not be provided at this time. Data will be possible to deliver to Bianca when the DDS is in production. Instructions for how this will work will come at a later time.

2.1.2 PyPi - MacOS / Linux

1. To perform these steps you need to have Python version 3.8 or higher installed. It's possible that it could work with other versions, but this cannot be guaranteed.
   - To install Python, please first run

     ```
     python --version
     ```

     It's possible that this shows `Python 2.7`, in which case you should run

     ```
     python3 --version
     ```

     If this does not return `Python 3.8.x` or higher, you will need to install Python.

   Warning: Make sure you have the latest version of **pip**.

   ```
   python3 -m pip install --upgrade pip
   ```

2. The DDS CLI is available on PyPi. To install the DDS CLI, open the terminal and run

   ```
   $ pip install dds-cli
   ```

3. Once the installation has finished, test that everything is working correctly:

```
$ dds --help
```

This should display a logo, version information and a short usage message. If there are no errors when running this command, the test has succeeded and you should be able to move on to *Running the command*.

2.1.3 Windows

We are working on creating an executable which will perform all required installations. However, for now, we have made detailed instructions for how you can install the DDS CLI on Windows. The instructions can be found here.

2.2 Running the command

The main command `dds` has some options and possible customisations. A detailed list of these can be found *here*.

Some commands should not be possible to successfully run from a Researcher account. The affected commands are marked with asterisks (***). However, we ask you to try these commands anyway and report back to us if anything unexpected occurs.

2.2.1 The five group commands

The DDS CLI has the following group commands: *auth*, *user*, *project*, *data* and *ls*.

dds auth

`dds auth` and its subcommands are used for creating and managing sessions. This will enable you to use the CLI without specifying your user credentials for a certain amount of time, currently 48 hours.

See the test protocol and the command documentation *here*.

dds user

You can use the `add user` group command to manage your own and (if you have administrative permissions) other user accounts.

See the test protocol and the command documentation *here*.

dds project

The `dds project` command is for creating and managing projects. The majority of the functionalities regarding project management is only available to *Unit Admin* and *Unit Personnel* accounts.

See the test protocol and the command documentation *here*.

dds data

The `dds data` group command is used for uploading, downloading, listing and deleting data. Only **Unit Admin** and **Unit Personnel** accounts can upload and delete data. All account types can list and download.

See the test protocol and the command documentation *here*.

dds ls

The `dds ls` group command can be used for listing both projects and project contents. Calling the `dds ls` command should produce the same output as `dds project ls`, and calling `dds ls --project` should result in the same output as when calling `dds data ls`.

See the test protocol and the command documentation *here*.

CHAPTER

THREE

HOW TO TEST THE WEB INTERFACE

The DDS web interface can be found at https://delivery.scilifelab.se/. There will only be a log in page and the possibility of requesting a password change. A guide on how to test out the existing web can be found *here*.

CHAPTER

FOUR

COMMAND DOCUMENTATION AND GUIDE

4.1 DDS main CLI command

4.1.1 How to test the functionality of the main *dds* command

Note: When running the commands, remember to make a note of whether or not any information or error messages are understandable and if there's anything we need to improve on, including the documentation on this page.

Steps

1. Run the *dds* command without any options or additional commands. Simply run:

```
$ dds
```

   Note: Please let us know whether there is any additional information that you would like to see added to the help text.

2. When running other subcommands, please try out the different flags and options listed below. To see the instructions on how to use these commands, go to *dds auth*, *dds user*, *dds project*, *dds data* or *dds ls*.

dds

SciLifeLab Data Delivery System (DDS) command line interface.

Access token is saved in a .dds_cli_token file in the home directory.

```
dds [OPTIONS] COMMAND [ARGS]...
```

Options

-v, --verbose
Print verbose output to the console.

-l, --log-file <filename>
Save a log to a file.

--no-prompt
Run without any interactive features.

-tp, --token-path <token_path>
The path where the authentication token will be stored. For a normal use-case, this should not be needed.

--version
Show the version and exit.

Commands

auth
Group command for creating and managing…

data
Group command for uploading, downloading…

ls
List the projects you have access to or…

project
Group command for creating and managing…

user
Group command for managing user accounts,…

4.2 *dds auth*

Section structure

This section begins with a description and step-by-step guide to how you could test this command. You can find the different commands and their options at the *bottom* of the section.

4.2.1 How to test the *dds auth* command functionality

Note: When running the commands, remember to make a note of whether or not any information or error messages are understandable and if there's anything we need to improve on, including the documentation on this page.

Steps

1. Help: `--help`

Run

```
dds auth --help
```

Note: Please let us know whether there is any additional information that you would like to see added to the help text.

---

2. Start authenticated session: `login`

```
dds auth login
```

2.1. With incorrect credentials

> **Note:** You **should not** be granted access and you **should not** have a *.dds_cli_token* file in your home directory.

2.2. Using the correct characters in the credentials but exchanging them to lower case or upper case depending on the correct format

> **Note:** **Username:** The username should be case *insensitive*. If your username is `username` you should also be able to login with `USERNAME` or `uSeRnAmE` etc.
>
> **Password:** The password should be case *sensitive*. Only the exact characters in your password, including whether or not they are upper case or lower case, should give you access to the system.
>
> If either the username or password is incorrect, you **should not** be granted access and you **should not** have a *.dds_cli_token* file in your home directory.

2.3. With correct credentials. You should receive an email containing a one-time code and be prompted by the command line to enter this code.

(i) Fill in an incorrect one-time code.

> **Note:** You **should not** be granted access and you **should not** have a *.dds_cli_token* file in your home directory. You should be asked by the command line if you want to try again. If you choose to try again, you should not recieve a new one-time code. If you cancel the current command and run *dds auth login* again, you should also not receive a new one-time code. However, if you wait 15 minutes and then try again, you should receive a new one-time code via email. This setup is due to security reasons.

(ii) Wait 15 minutes and run the *dds auth login* command again. You should receive a new email with a new one-time code. Fill in the one-time code sent in the previous email.

> **Note:** The system should not accept an old one-time code. You **should not** be granted access and you **should not** have a *.dds_cli_token* file in your home directory.

(iii) Fill in the valid one-time code

Note: You should be granted access, a message should be displayed, and there should be a *.dds_cli_token* in your home directory.

(iv) Open the *.dds_cli_token* file or (in Unix systems) run

```
cat ~/.dds_cli_token
```

Are the contents/output readble?

Note: The file should be readable but the token **should look like jibberish**, inform the SciLifeLab Data Centre *immediately* if you can discern any information from the file contents.

---

1. Get session information: *info*

```
dds auth info
```

The information printed out should contain:

- Whether the token will expire
- When the token will expire

Is the information understandable?

1. End the authenticated session: `logout`

```
dds auth logout
```

Expected result

A success message should be displayed and the file `.dds-cli-token` file in your home directory should be deleted.

---

5. Continue with other commands

Continue using the DDS CLI:

- Manage users: *dds user*
- Manage projects: *dds project*
- Upload, download, list and remove data: *dds data*
- List projects and data: *dds ls*.

The command

dds auth

Group command for creating and managing authenticated sessions.

Authenticate yourself once and run multiple commands within a certain amount of time (currently 48 hours) without specifying your user credentials. If you do not authenticate yourself and start a new session, you will need to provide your DDS username when running the other commands.

```
dds auth [OPTIONS] COMMAND [ARGS]...
```

info

Display information about token.

Information displayed: - If the token is about to expire - Token age - Time of token expiration

```
dds auth info [OPTIONS]
```

login

Start or renew an authenticated session.

Creates or renews the authentication token stored in the '.dds_cli_token' file.

Run this command before running the cli in a non-interactive fashion as this enables the longest possible session time before a password needs to be entered again.

```
dds auth login [OPTIONS]
```

logout

End authenticated session.

Removes the saved authentication token by deleting the '.dds_cli_token' file.

```
dds auth logout [OPTIONS]
```

4.3 *dds user*

Section structure

This section begins with a description and step-by-step guide to how you could test this command. You can find the different commands and their options at the *bottom* of the section.

4.3.1 How to test the *dds user* command functionality

Note: When running the commands, remember to make a note of whether or not any information or error messages are understandable and if there's anything we need to improve on, including the documentation in this section.

From a *Unit Admin* or *Unit Personnel* account, you should be able to run all commands successfully. From a *Researcher* account, however, you will only be able to run the **dds user info** command, unless you're a *Project Owner* for a specific project. In this case you should only be able to manage accounts of other Project Owners and Researchers that are involved in the project you are set as Project Owner in.

Note: Although Project Owners and Researchers should not be able to successfully run most of these commands, we ask you to try these out anyway, and report back if anything unexpected happens.

Table 1: Different roles and their permissions

Command	Unit Admin	Unit Personnel	Project Owner	Researcher
info	Yes	Yes	Yes	Yes
add	Yes	Yes	Yes *Within certain projects*	No
deactivate	Yes	Yes	Yes *Other Project Owners and Researchers within certain projects*	No
activate	Yes	Yes	Yes *Other Project Owners and Researchers within certain projects*	No
delete	Yes	Yes	Yes *Other Project Owners and Researchers within certain projects*	Yes *Only own account*

Steps

Note: These sections/test steps assume you have already started a session with the *dds auth* command.

1. Help: `--help`

Run

```
dds user --help
```

Note: Please let us know whether there is any additional information that you would like to see added to the help text.

2. Get user info: `info`

```
dds user info
```

Note: The information printed out should contain your

- Username
- Role
- Name
- Emails connected to the account

3. Add users: `add`

```
dds user add
```

Note: Use quotation marks to use spaces within the value passed into an option.

E.g: `--role "Project Owner"`

3.1. Invite a new user to the DDS

> **Warning:** Please either use one of your own accounts or a colleague that is also involved in the testing of the DDS.

3.2. Invite the same user to DDS again

> **Note:** This should not work and a message notifying you that the user has an ongoing invite should be displayed.

3.3. Try to invite yourself by specifying the email your current account is registered with

> **Note:** This should not work and a message notifying you of this should be displayed.

3.4. Try to invite a user (without the *project* option) and specify the *role*

- Unit Admin

  > **Note:** Should only work for other Unit Admin accounts.

- Unit Personnel

  > **Note:** Should only work for other Unit Personnel and Unit Admin accounts.

- Project Owner

  Note: Should work for Researcher accounts assigned as Project Owners within a specific project, Unit Personnel and Unit Admin accounts.

- Researcher

  Note: Anyone should be able to invite a user with the role Researcher.

3.5. Try to invite a user (*project* option *specified*) and the *–role*:

- Unit Admin

  Note: This should work for other Unit Admins as above, but there should be a message displayed saying that all Unit Admins get access to all projects within a specific unit.

- Unit Personnel

  Note: This should work for other Unit Personnel and Unit Admin accounts, but as for the Unit Admin, all Unit Personnel accounts get access to all unit projects and there should therefore be a print out of a message informing you of this.

- Project Owner

  Note: Should work for Researcher accounts assigned as Project Owners within a specific project, Unit Personnel and Unit Admin accounts.

- Researcher

  Note: Anyone should be able to invite a user with the role Researcher.

4. Deactivate user: `deactivate`

```
dds user deactivate
```

4.1. Try to deactivate your own account

Note: This should not work and a message notifying you of this should be displayed.

4.2. Try to deactivate a fake account

Note: A fake account does not exist and should therefore not be possible to deactivate.

4.3. Try to deactivate another account, either one of your own, created in the steps above, or another colleagues.

Warning: Please make sure to notify the user you are attempting to deactivate.

Note: You can also attempt inviting yourself to multiple accounts and specifying different roles, after which (and after registration in the *web*<*web*>) you can attempt to deactivate the different accounts. Have a look at the table at the top of the section if you are uncertain about which actions should be possible.

5. Activate/Reactivate user: `activate`

```
dds user activate
```

5.1. Try to activate your own account

Note: This should not work and a message notifying you of this should be displayed.

5.2. Activate an account that is already activate

Note: Use one of the accounts which you invited in the steps above. They should be automatically activated once they have registered an account in the web, and therefore should not be possible to activate again.

5.3. Try to activate a fake account

Note: A fake account does not exist and should therefore not be possible to activate.

5.4. Reactivate the other account that you attempted (and hopefully in some cases succeeded) to deactivate in the *step above*

Note: Try this command by specifying users with different roles. Have a look at the table at the top of the section if you are uncertain about which actions should be possible.

6. Delete user: `delete`

Warning: Do not delete any accounts during this testing period. If you wish to try out this functionality, please wait until you are finished with testing the other commands. Deleted accounts are non-reversible.

The command

dds user

Group command for managing user accounts, including your own.

```
dds user [OPTIONS] COMMAND [ARGS]...
```

activate

Activate/Reactivate user accounts.

If you have sufficient admin privileges, you may activate the accounts of other users. Specify the e-mail address as argument to the main command to initiate the activation process.

```
dds user activate [OPTIONS] [EMAIL]
```

Arguments

[EMAIL]
Required argument

add

Add a user to the DDS system or hosted projects.

Specify a users email and role to associate it with projects.

If the user doesn't exist in the system yet, an invitation email will be sent automatically to that person.

```
dds user add [OPTIONS] [EMAIL]
```

Options

-p, --project <project>
Existing Project you want the user to be associated to.

-r, --role <role>
Required Type of account.

Options Super Admin | Unit Admin | Unit Personnel | Project Owner | Researcher

--unit <unit>
Can only be used by Super Admin. To specify which unit the user should belong to.

--no-mail
Do not send e-mail notifications regarding project updates.

Arguments

[EMAIL]
Required argument

deactivate

Deactivate user accounts in the Data Delivery System.

If you have sufficient admin privileges, you may deactivate the accounts of other users. Specify the e-mail address as argument to the main command to initiate the deactivation process.

```
dds user deactivate [OPTIONS] [EMAIL]
```

Arguments

[EMAIL]
Required argument

delete

Delete user accounts from the Data Delivery System.

Use this command with caution. Deletion of accounts cannot be undone.

To request the removal of your own account, use the *–self* flag without any arguments. An e-mail will be sent to you asking to confirm the deletion.

If you have sufficient admin privileges, you may also delete the accounts of other users. Specify the e-mail address as argument to the main command to initiate the removal process.

```
dds user delete [OPTIONS] [EMAIL]
```

Options

--self
Request deletion of own account.

Arguments

[EMAIL]
Optional argument

info

Display information connected to your own DDS account.

```
dds user info [OPTIONS]
```

4.4 *dds project*

Section structure

This section begins with a description and step-by-step guide to how you could test this command. You can find the different commands and their options at the *bottom* of the section.

4.4.1 How to test the *dds project* command functionality

Note: When running the commands, remember to make a note of whether or not any information or error messages are understandable and if there's anything we need to improve on, including the documentation in this section.

From a *Unit Admin* or *Unit Personnel* account, you should be able to run all commands successfully. From a *Researcher* account, however, you should only be able to run the **dds project ls** and **dds project status display** commands, unless you're a *Project Owner* for a specific project. In this case you should also be able to grant (**dds project access grant**) and revoke (**dds project access revoke**) project access to other Project Owners and Researchers that are involved in the project you are set as Project Owner in.

Note: Commands that should only work for Unit Adminds and Unit Personnel are noted in the step it applies to with three asterisks (***). Asterisks applied to a main item (e.g. 3.) also applies to the subitems (e.g. 3.1., 3.2. etc). If there is additional information about the different permissions, this is displayed in a parenthesis beside the asterisk.

Although Project Owners and Researchers should not be able to successfully run most of these commands, we ask you to try these out anyway, and report back if anything unexpected happens.

Steps

Note: These sections/test steps assume you have already started a session with the *dds auth* command.

1. Help: `--help`

Run

```
dds project --help
```

Note: Please let us know whether there is any additional information that you would like to see added to the help text.

2. List projects: `ls`

```
dds project ls
```

Note: This command performs the same actions as `dds ls` (with out any specified project). You can find the documentation for that *here*. Please test this and compare the output, it should be identical to what you see here.

2.1. Without any options

Expected result

A table containing the projects you have access to should be displayed.

Warning: Notify us immediately if you are a **Unit Admin** or **Unit Personnel** and have created an project, but this project is not displayed in the printed table.

2.2. Sort the projects using the `--sort` option.

Expected result

The projects should be sorted according to *Last updated* by default.

Try sorting the projects according to different columns.

2.3. Display the usage using the `--usage` flag ***

Expected result

This should add two columns to the table: *Usage* and *Cost*.

2.4. Display the projects in json format using the `--json` flag.

Expected result

This should print out the projects in json format instead of a table. The output should look something like this:

```
[
   {
      "Last updated": "Sun, 20 Feb 2022 11:16:18 CET",
      "PI": "PI Name",
      "Project ID": "project_1",
      "Size": 0,
      "Status": "In Progress",
      "Title": "First Project"
   },
   {
      "Last updated": "Sun, 20 Feb 2022 11:16:18 CET",
```

(continues on next page)

(continued from previous page)

```
      "PI": "PI Name",
      "Project ID": "project_2",
      "Size": 0,
      "Status": "In Progress",
      "Title": "Second Project"
   }
]
```

3. Create a project `create` ***

```
dds project create
```

Note: All projects are set as **sensitive** by default. This means that all data will be encrypted before upload, and decrypted after download. Depending on the size of the data, this may take some time. There is a `--non-sensitive` flag available, however at this time this is not functional and all projects are handled in the same way. Therefore, if you use the `--non-sensitive` flag, the project will be saved in the database as non-sensitive, but the data will still be handled as if it were sensitive. This will be changed as soon as possible.

3.1. Without any options

Expected result

To create a project you need to specify a title, a description and the principal investigator (PI) of that project. Without this information, creating a project should not be possible.

3.2. With all required options: `--title`, `--description`, `--principal-investigator` but without adding any users

Note: Use quotation marks to use spaces within the value passed into an option.

E.g: `--title "This is a longer title"`

Expected result

A project should be created and you should see a message displayed stating the new Project ID. This Project ID should be passed in as the `--project` option when running project-specific commands. If you forget the Project ID, use the `dds ls` command to list all projects.

3.3. Create a project and specify a Researcher (`--researcher`) that should have access to the project.

You can either specify a researcher that you know has a DDS account, or you can specify a user which you wish to invite to the DDS.

Expected result

A project should be created, a message should be displayed stating the new Project ID, and an additional message should be displayed, stating that the specified Researcher has either been sent an invitation, or granted access to the project, depending on whether or not the specified email has an existing account.

3.4. Create a project and specify an Project Owner (`--owner`)

As in 3.3. above, the owner can either be a new user or and existing one.

Expected result

A project should be created, a message should be displayed stating the new Project ID, and an additional message should be displayed, stating that the specified owner has either been sent an invitation, or granted access to the project, depending on whether or not the specified email has an existing account. The message should also inform you that the user has been granted access as a Project Owner.

3.5. Specify both a Researchuser and an owner.

Perform the same steps as in 3.3. and 3.4. but specify both a `--researcher` and an `--owner`.

Expected result

This should result in a similar output as in the previous steps.

3.6. With multiple users.

Perform the same steps as in 3.3. and 3.4. but try specifying multiple researchers and / or owners.

Expected result

This should result in a similar output as in the previous steps.

4. View and manage the project statuses: `status`

```
dds project status
```

4.1. Display the status of a project (`status display`)

- Specify a non-existent project

Expected result

A message saying that the project does not exist should be displayed.

- Specify an existing project

Expected result

The output should look something like this:

```
Current status of someunit00002: In Progress
```

- Also show the status history with the `--show-history` flag

Expected result

The output should look something like this:

```
Current status of someunit00002: In Progress
INFO     Status history
In Progress, Sun, 20 Feb 2022 11:51:13 CET
```

4.2. Attempt changing the project status ***

Tip: We recommend testing this functionality in the following steps:

(i) Create a project

(ii) Display status. The status should always be **In Progress** at this point.

(iii) Attempt changing the status.

(iv) Display status.

Please attempt to change the project status in different orders.

The possible status changes are displayed visually on this board and are listed in the *documentation below*.

5. Manage project access: `access` *** (Also possible for Project Owners)

```
dds project access
```

Tip: We recommend testing this functionality in the following steps:

(i) List the users with access to a specific project: `dds ls --project <project_id> --users`. More details on the `dds ls` command can be found *here*.

(ii) Grant / Revoke / Fix access for a specific user as described in the steps below.

(iii) Do step (i)

5.1. Grant access to a project (`access grant`)

Tip: We suggest you list the users with access to the project in question before performing the following tests. Go *here* for the instructions on how to do this.

(i) Specify a non-existent user.

Expected result

The user should be invited and a message notifying you of this should be displayed. Note that you can only use `grant` for Researchers, not Unit Admins or Unit Personnel.

(ii) Specify an existing user.

- Attempt to grant access to a user with the role **Unit Admin** or **Unit Personnel**

  Expected result

  This command should produce an error message. Unit Admins / Personnel have access to *all* projects connected to a specific unit. Only researchers can be granted access with this command.

- Attempt to grant access to a user with the role **Researcher**

  Try to grant access both to a user which already has access to the specified project, and one who does not. Also try this with the `--owner` flag.

  Expected result

  If the user already has access to the project, and is already set as the Project Owner, using the `--owner` flag for this command should return a message stating that the user is already associated to the project in that capacity. The same applies to it the user is associated to the project as a Researcher and the `--owner` flag is *not used*.

5.2. Revoke project access (`access revoke`)

Tip: We suggest you list the users with access to the project in question before performing the following tests. Go *here* for the instructions on how to do this.

(i) Specify a non-existent user

Expected result

A non-existent user cannot have access to a project and it should therefore not be possible to revoke project access for that user.

(ii) Specify an existing user that does not have access to the current project.

Expected result

This should produce a message saying that the specified user does not have access to the project.

(iii) Revoke project access for the users that you granted access in step 5.1. (ii)

Expected result

A message should be displayed informing you that the users' project access has been revoked.

5.3 Fix project access (`access fix`)

Note: This command is used to reactivate a users' project access a password reset. More specifically, the user has performed the following steps:

(i) Requested a password reset

(ii) Clicked on the link in the received email

(iii) Chosen a new password

(iv) Contacted the Project Owner or a Unit Admin / Personnel connected to the unit responsible for a specific project to regain access

Tip: Unless someone contacts you about losing access, this step is slightly difficult to test. However, you can follow the *web instructions* on how to request a password reset and ask another user to reactivate your project access with this command.

You can also attempt this with users that do not have access to a specific project.

The command

dds project

Group command for creating and managing projects within the DDS.

```
dds project [OPTIONS] COMMAND [ARGS]...
```

access

Manage specific users access to a project.

```
dds project access [OPTIONS] COMMAND [ARGS]...
```

fix

Re-grant project access to user that has lost access due to password reset.

```
dds project access fix [OPTIONS] [EMAIL]
```

Options

-p, --project <project>
Project ID.

Arguments

[EMAIL]
Required argument

grant

Grant user access to a project.

```
dds project access grant [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

-e, --email <email>
Required Email of the user you would like to grant access to the project.

--owner
Grant access as project owner. If not specified, the user gets Researcher permissions within the project.

--no-mail
Do not send e-mail notifications regarding project updates.

revoke

Revoke user access to a project.

```
dds project access revoke [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

-e, --email <email>
Required Email of the user for whom project access is to be revoked.

create

Create a project.

```
dds project create [OPTIONS]
```

Options

-t, --title <title>
Required The title of the project.

-d, --description <description>
Required A description of the project.

-pi, --principal-investigator <principal_investigator>
Required The name of the Principal Investigator.

--researcher <researcher>
Email of a user to be added to the project as Researcher. Use the option multiple times to specify more than one researcher [multiple]

--owner <owner>
Email of user to be added to the project as Project Owner. Use the option multiple times to specify more than one project owner [multiple]

--non-sensitive
Indicate whether the project contains only non-sensitive data

ls

List all projects you have access to in the DDS.

Calls the *dds ls* function.

```
dds project ls [OPTIONS]
```

Options

--sort <sort>
Which column to sort the project list by.

Options id | title | pi | status | updated | size | usage | cost

--usage
Show the usage for available projects, in GBHours and cost.

Default False

--json
Output project list as json.

Default False

status

Manage project statuses.

```
dds project status [OPTIONS] COMMAND [ARGS]...
```

abort

Abort a released project.

This deletes all project data.

```
dds project status abort [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

archive

Manually archive a released project.

This deletes all project data.

```
dds project status archive [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

delete

Delete an unreleased project.

This deletes all project data.

```
dds project status delete [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

display

Display and manage project statuses.

```
dds project status display [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

--show-history
Show history of project statuses in addition to current status.

release

Make project data available for user download.

```
dds project status release [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

--deadline <deadline>
Deadline in days when releasing a project.

--no-mail
Do not send e-mail notifications regarding project updates.

retract

Set the status as *In Progress*.

This allows Unit Personnel / Admins to upload additional data to the project.

```
dds project status retract [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

4.5 *dds data*

Section structure

This section begins with a description and step-by-step guide to how you could test this command. You can find the different commands and their options at the *bottom* of the section.

4.5.1 How to test the *dds data* command functionality

Note: When running the commands, remember to make a note of whether or not any information or error messages are understandable and if there's anything we need to improve on, including the documentation in this section.

From a *Unit Admin* or *Unit Personnel* account, you should be able to run all commands successfully. From a *Researcher* account, however, you should only be able to run the **dds data get** and **dds data ls** commands.

Note: Commands which should only work for Unit Adminds and Unit Personnel are noted in the step it applies to with three asterisks (***). Asterisks applied to a main item (e.g. 3.) also applies to the subitems (e.g. 3.1., 3.2. etc). If there is additional information about the different permissions, this is displayed in a parenthesis beside the asterisk.

Although Project Owners and Researchers should not be able to successfully run most of these commands, we ask you to try these out anyway, and report back if anything unexpected happens.

Steps

Note: These sections/test steps assume you have already started a session with the *dds auth* command.

1. Help: `--help`

Run

```
dds data --help
```

Note: Please let us know whether there is any additional information that you would like to see added to the help text.

2. Upload: `put` ***

```
dds data put
```

Note: Some project statuses do not permit uploads. A visual respresentation of the project statuses can be found on this board.

Recommended testing procedure

We highly recommend that you test the upload functionality with data that is typical (both in size and number) for a project within your unit.

We also recommend that you test the upload functionality by performing the following general steps:

(i). Create a project with the `dds project create` command as described *here*.

(ii). Try uploading.

Expected result

When creating a project the status should automatically be set to **In Progress** and therefore the upload should be successful.

(iii). Change the project status to **Available** (`dds project status release`, more information *here*)

(iv). Try uploading.

Expected result

The upload should *not* work.

(v). Change the project status to **In Progress** again (`dds project status retract`, more information *here*)

(vi). Try uploading.

Expected result

The upload should once again succeed.

(vii). Change the project status to any other status and try the upload again.

Expect result

Upload should not be allowed.

Options to test

Test this command by trying different flags and options, for example:

- Without any specified options

  Expected result

  The upload requires at least a project and data to upload. The CLI should display a help message.

- Specify a file / folder with the `--source` option. Also try specifying `--source` multiple times.
- Specify files / folders within a text file and use the `--source-path-file` option.
- Specify both the `--source` and `--source-path-file` option
- Test the upload with a large number of files
- Test the upload with a few large files
- Try uploading the same file twice

  Expected result

A message should be displayed stating that there's no data to upload. To upload the same file again, overwriting the previous file, specify the `--overwrite` option.

- Try the `--silent` flag and `--num-threads` option

3. List data: `ls`

```
dds data ls
```

Note: This command performs the same actions as `dds ls --project`. You can find the documentation for this *here*. Please test this and compare the output, it should be identical to what you see here.

Make sure to compare the output to the file structure you recently uploaded. If it does not seem correct, please contact us.

3.1. Run the command without any options

Expected result

This should produce a help message. The minimum required information for this command is the Project ID, specified with the `--project` option.

3.2. List the contents of a specific folder (`--folder`)

3.3. List the project contents as json format (`--json`)

3.4. Use the `--tree` flag to list all project contents as a tree structure

3.5. List the researchers with access to the project (`--users`)

4. Download: `get`

```
dds data get
```

Note: Some project statuses do not permit downloads. For Researcher accounts, data is only available for download in projects with the status **Available**. For Unit Admin and Unit Personnel accounts, data is *also* available for download when the projects have the status **In Progress**. A visual representation of the project statuses can be found on this board.

To simplify the testing of this section, we have split it up into *4.1. Unit Admins and Unit Personnel* and *4.2. Researchers*.

Options to test

Independent of account role, test this command by trying different flags and options, for example:

- Without any specified options

Expected result

The download requires a project ID and information on which data to download. The CLI should display a help message.

- Specify a file / folder with the `--source` option. Also try specifying `--source` multiple times in the same command.
- Specify files / folders within a text file and use the `--source-path-file` option.
- Try the `--silent` flag and `--num-threads` option.
- Specify where you would like to download the data to by using the `--destination` option.

Note: The `--destination` cannot (at this time) be an existing directory. You need to specify a new directory name and the DDS CLI will create it for you.

- Use the `--verify-checksum` flag. This performs an additional check to verify that the downloaded file is identical to the file uploaded by the Unit Admin / Personnel.

Expected result

A message informing you that the checksum verification passed should be displayed.

Warning: Notify us immediately if the checksum verification fails.

4.1. *Unit Admins and Unit Personnel*

Recommended testing prodecure

We recommend that you test the functionality by performing the following general steps:

(i) Use a project which you've uploaded data to. Make sure the project status is **In Progress**. See the *dds project section* for instructions on how to do this.

(ii) List the project contents with the `ls` command described in section 3. above.

(iii) Download a file

- Try to download a file that is not listed in step (ii) above

Expected result

A message should be printed out, letting you know that the file could not be found.

- Download a file that is listed in step (ii) above.

Expected result

A folder should be created in your current working directory (or in the location you choose, see `--destination` below), the file should be downloaded, and finally decrypted. Once the file has

been downloaded and decrypted, a message should be displayed notifying you where you can find the file.

(iv) Download a folder

- Try to download a folder that is not listed in step (ii) above

Expected result

A message should be printed out, notifying you that the folder could not be found.

- Download a folder that is listed in step (ii) above

Expected result

The result of this should be similar to the download in step (iii) above.

(v) Download a folder and a file

Expected result

The result of this command should be similar to the download in steps (iii) and (iv) above.

(vi) Set the project status as **Available**. See the *dds project section* for instructions on how to do this.

(vii) Try to download a file / folder

Expected result

Since download is available for Unit Admins and Unit Personnel when the project status is **Available**, the download should be successful and the output should be similar to that of the download steps above.

(viii) Set the project as **Archived** or **Aborted**.

Expected result

Archiving or aborting a project should delete all project data.

(ix) Try to download data

Expected result

Download should not be possible and a message informing you that the project status prevents you from downloading should be displayed.

4.2. *Researchers*

Recommended testing prodecure

We recommend that you test the functionality by performing the following general steps:

(i) Display the status of a project you have access to. Use `dds ls` to list the projects, and `dds project status display` to see the status of a specific project. Choose a project which has the status **Available**.

> **Note:** When a project status is changed from **In Progress** to **Available**, you should receive an email informing you that there is data available for download. If you have access to a project which is **Available** but you have not received an email about this, first check your junk folder. If you still cannot find this email, contact us and we will look into it.

(ii) List the contents of the project. See *List data* above.

(iii) Download a file

- Try to download a file that is not listed in step (ii) above

  Expected result

  A message should be printed out, letting you know that the file could not be found.

- Download a file that is listed in step (ii) above.

  Expected result

  A folder should be created in your current working directory (or in the location you choose, see `--destination` below), the file should be downloaded, and finally decrypted. Once the file has been downloaded and decrypted, a message should be displayed notifying you where you can find the file.

(iv) Download a folder

- Try to download a folder that is not listed in step (ii) above

  Expected result

  A message should be printed out, notifying you that the folder could not be found.

- Download a folder that is listed in step (ii) above

  Expected result

  The result of this should be similar to the download in step (iii) above.

(v) Download a folder and a file

Expected result

The result of this command should be similar to the download in steps (iii) and (iv) above.

(vi) Use the `--get-all` flag to download the entire project.

Note: Make sure you have enough space.

5. Delete (remove): `rm` ***

```
dds data rm
```

Recommended testing prodecure

We recommend that you test the functionality by performing the following general steps:

(i) List the project contents with the `dds data ls` command as described in *List data* above.

(ii) Attempt to remove a file.

(iii) List the project contents again.

(iv) Attempt to remove a folder.

(v) List the project contents again.

(vi) Use the *–rm-all* flag to remove all project contents.

Note: If the CLI displays a success message, but the data is not removed, contact us and we will look into it.

Expected result (all `rm` steps above)

When attempting to remove data which does not exist, a message should be displayed in the terminal saying that the data was not found.

When attempting to remove data which does exist in the project and is listed in step (i), a success message should be displayed, informing you that the data was removed.

The command

dds data

Group command for uploading, downloading and managing project data.

```
dds data [OPTIONS] COMMAND [ARGS]...
```

get

Download files within a project.

This downloads, decrypts, and finally decompresses (if compressed by the DDS) the files.

```
dds data get [OPTIONS]
```

Options

-p, --project <project>
Required Project ID from which you're downloading data.

-nt, --num-threads <num_threads>
Number of parallel threads to perform the delivery

Default 4

-s, --source <source>
Path to file or directory. Use the option multiple times to specify more than one source [multiple]

-spf, --source-path-file <source_path_file>
File containing path to files or directories.

-d, --destination <destination>
Destination of downloaded files.

--break-on-fail
Cancel download of all files if one fails.

Default False

--silent
Turn off progress bar for each individual file. Summary bars still visible.

Default False

-a, --get-all
Download all project contents.

Default False

--verify-checksum
Perform SHA-256 checksum verification after download (slower).

Default False

ls

List project contents.

Same as dds ls [PROJECT ID].

```
dds data ls [OPTIONS]
```

Options

-p, --project <project>
Required Project ID.

-f, --folder <folder>
List contents in this project folder.

--json
Output in JSON format.

Default False

--size
Show size of project contents.

Default False

--tree
Display the entire project(s) directory tree.

Default False

--users
Display users associated with a project(Requires a project id).

Default False

put

Upload data to project.

This first compresses the files (if not already compressed), encrypts them, and finally uploads them to Safespring S3 Storage.

```
dds data put [OPTIONS]
```

Options

-p, --project <project>
Required Project ID to which you're uploading data.

-s, --source <source>
Path to file or directory (local). Use the option multiple times to specify more than one source [multiple]

-spf, --source-path-file <source_path_file>
File containing path to files or directories.

-nt, --num-threads <num_threads>
Number of parallel threads to perform the delivery

Default 4

--overwrite
Overwrite files if already uploaded.

Default False

--break-on-fail
Cancel upload of all files if one fails.

Default False

`--silent`
Turn off progress bar for each individual file. Summary bars still visible.

Default False

rm

Delete project data.

```
dds data rm [OPTIONS]
```

Options

`-p, --project <project>`
Required Project ID.

`-fl, --folder <folder>`
Path to folder to remove. Use the option multiple times to specify more than one folder [multiple]

`-f, --file <file>`
Path to file to be removed. Use the option multiple times to specify more than one file [multiple]

`-a, --rm-all`
Remove all project contents.

4.6 *dds ls*

Section structure

This section begins with a description and step-by-step guide to how you could test this command. You can find the different commands and their options at the *bottom* of the section.

4.6.1 How to test the *dds ls* command functionality

Note: When running the commands, remember to make a note of whether or not any information or error messages are understandable and if there's anything we need to improve on, including the documentation in this section.

Commands which should only work for Unit Adminds and Unit Personnel are noted in the step it applies to with three asterisks (***). Asterisks applied to a main item (e.g. 3.) also applies to the subitems (e.g. 3.1., 3.2. etc). If there is additional information about the different permissions, this is displayed in a parenthesis beside the asterisk.

Steps

Note: These sections/test steps assume you have already started a session with the *dds auth* command.

1. Help: `--help`

Run

```
dds ls --help
```

Note: Please let us know whether there is any additional information that you would like to see added to the help text.

2. List projects: `ls`

```
dds ls
```

Note: The output of this command is (should be) **identical** to the output produced by

```
dds project ls
```

See the *dds project* documentation.

2.1. Run the command without any options

Expected result

A table containing the projects you have access to should be displayed.

Warning: Notify us immediately if you are a **Unit Admin** or **Unit Personnel** and have created an project, but this project is not displayed in the printed table.

2.2. Sort the projects using the `--sort` option.

Expected result

The projects should be sorted according to *Last updated* by default.

Try sorting the projects according to different columns.

2.3. Display the usage using the `--usage` flag ***

Expected result

This should add two columns to the table: *Usage* and *Cost*.

2.4. Display the projects in json format using the `--json` flag.

Expected result

This should print out the projects in json format instead of a table. The output should look something like this:


```
[
   {
      "Last updated": "Sun, 20 Feb 2022 11:16:18 CET",
      "PI": "PI Name",
      "Project ID": "project_1",
      "Size": 0,
      "Status": "In Progress",
      "Title": "First Project"
   },
   {
      "Last updated": "Sun, 20 Feb 2022 11:16:18 CET",
      "PI": "PI Name",
      "Project ID": "project_2",
      "Size": 0,
      "Status": "In Progress",
      "Title": "Second Project"
   }
]
```


3. List project contents: `ls --project`

```
dds ls --project
```

Note: The output produced by this command is (should be) **identical** to the output produced by

```
dds data ls
```

See the *dds data* documentation.

3.1. Without any options

Expected result

This should produce a help message. The minimum required information for this command is the Project ID, specified with the `--project` option.

3.2. List the contents of a specific folder (`--folder`)

3.3. List the project contents as json format (`--json`)

3.4. Use the `--tree` flag to list all project contents as a tree structure

3.5. List the researchers with access to the project (`--users`)

The commands

dds ls

List the projects you have access to or the project contents.

To list all projects, run *dds ls* without any arguments.

Specify a Project ID to list the files within a project. You can also follow this with a subfolder path to show files within that folder.

```
dds ls [OPTIONS]
```

Options

-p, --project <project>
Project ID.

--sort <sort>
Which column to sort the project list by.

Options id | title | pi | status | updated | size | usage | cost

-f, --folder <folder>
List contents of this project folder.

--json
Output in JSON format.

Default False

--size
Show size of project contents.

Default False

--tree
Display the entire project(s) directory tree.

Default False

--usage
Show the usage for available projects, in GBHours and cost.

Default False

--users
Display users associated with a project(Requires a project id).

Default False

-lp, --projects
List all project connected to your account.

4.7 How to test the DDS web

The DDS web interface is currently very minimal. This will be improved on later. At this time the available functionalities within the web interface are

1. Register account
2. Login
3. MFA Authentication
4. Reset forgotten password
5. Change password

4.7.1 Steps

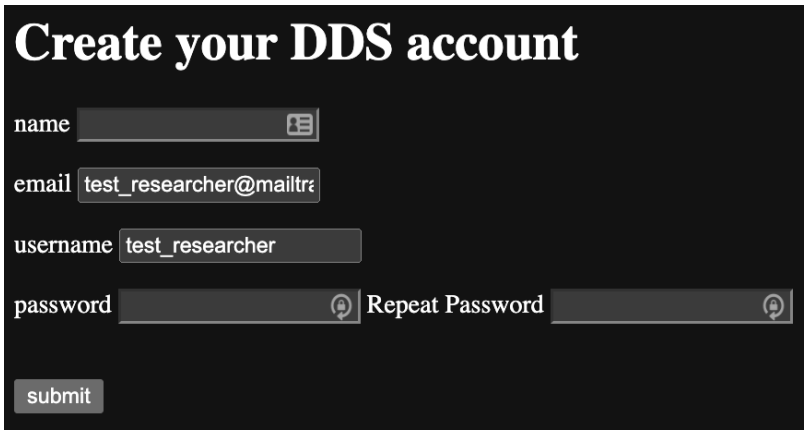
1. Register an account

When another user invites you to the DDS, you will get an email (currently from the email *dsw@scilifelab.se*). To test the registration functionality, we suggest that you perform the following steps.

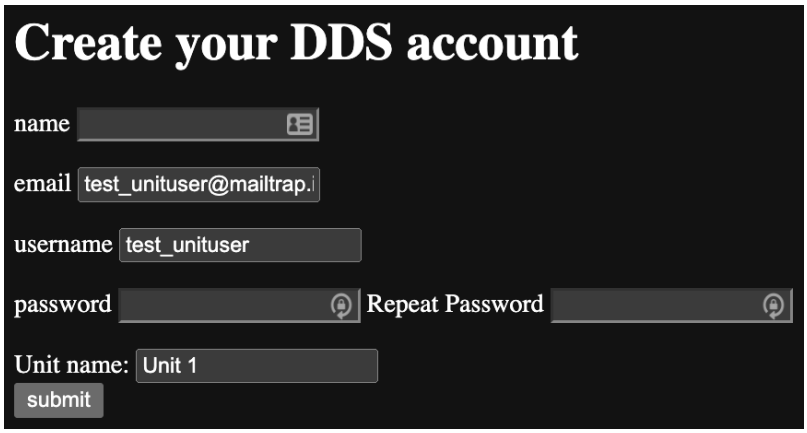
1.1. Go to your email inbox and open the email with a subject with the structure: "<Unit name> invites you to the SciLifeLab Data Delivery System". The email contents should look something like this:

1.2. Click on the **Sign Up** button. You should reach the *Registration* page.

If you have been invited to the DDS as a Researcher, the registration page should look something like this:



If you have invited to the DDS as a Unit Admin or Personnel, the registration page should look something like this:



1.2.1. Fill in your information. You cannot choose a different email than the one you got the invitation email to (but try it out), and the *Unit* field (if visible) cannot be changed.

- Fill in your full name.
- Try to choose a username shorter than 3 and one longer than 30 characters.

Expected result

An error message should be displayed, notifying you that the username is required to be between 3 and 30 characters long. You can also use underscores (`_`), hyphens (`-`) and periods (`.`).

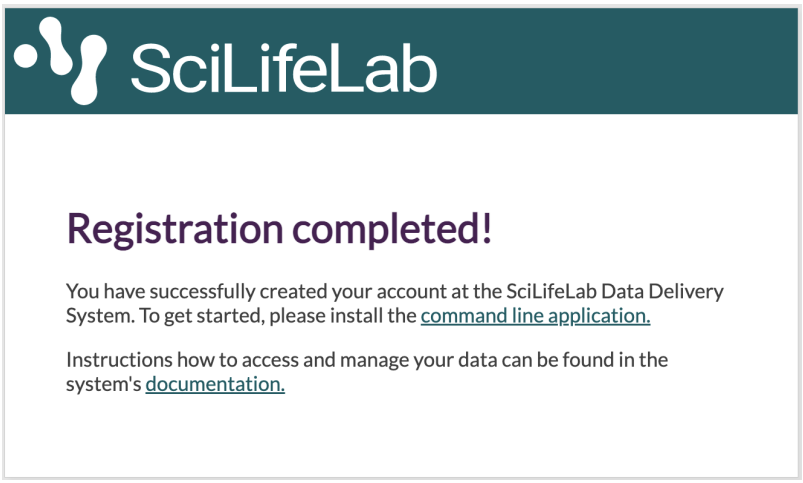
- **Try to choose an invalid password:**
  - Less than 10 characters
  - With only upper case letters
  - With only lower case letters

Expected result

A message should be displayed, notifying you that the password needs to meet the following requirements:

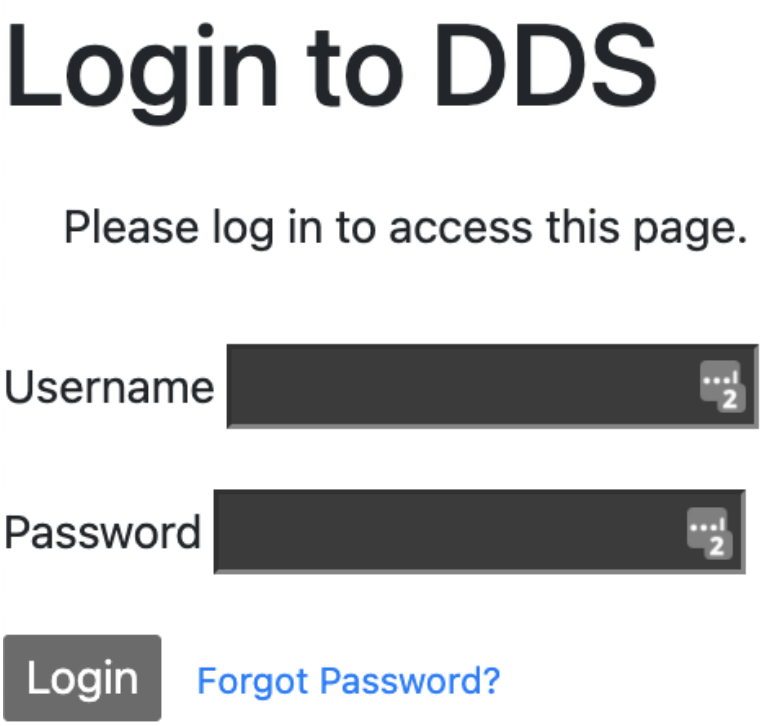
- At least 10 characters
- At most 64 characters
- Contain at least one digit OR a special character
- Contain at least one lower case letter
- Contain at least one upper case letter

1.3. Finally, fill in valid information and create an account. You should be redirected to the following page:



2. Login

2.1. Go to https://delivery.scilifelab.se/. You should see the following page.



2.2. Attempt to log in with

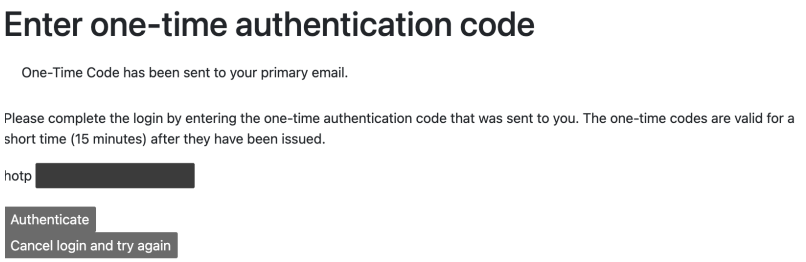
- Incorrect username
- Incorrect password
- Correct username and password

Expected result

When the username and/or password is correct, a message should be displayed notifying you of the specific error.

3. MFA Authentication

3.1. When filling in the correct user credentials and clicking *Login*, you should be met with the following page:



3.2. Go to your email inbox and open the email with the subject line "DDS One-Time Authentication Code". The email should contain a 8-digit code.

3.2.1. Go back to the DDS page and try to input

- An incorrect value for the one-time code. You can try one that is not 8 characters and one that is simply incorrect.
- The correct code that you received in the email

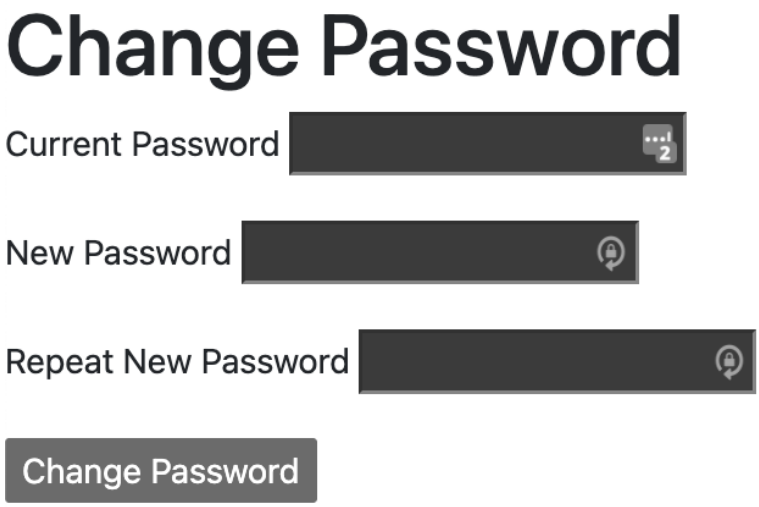
Expected result

If the code is invalid, an understandable message should be displayed.

3.2.2. When inputting the correct one-time code, you should be redirected to a very simple page with a logout button, and a link with the text "Change Password".

4. Change Password

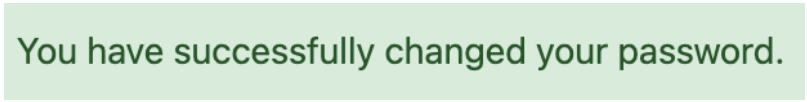
4.1. Log in to the DDS web interface and click on the "Change Password" link. You should be redirected to the following page:



4.2. Attempt to change password with

- The incorrect current password
- Invalid new password and non-matching fields
- Correct current password and valid new password

4.3. The following message should be displayed after successfully changing your password:



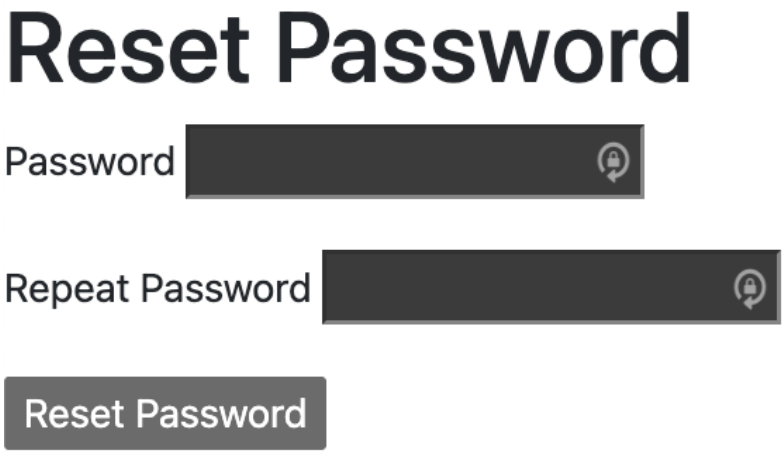
5. Reset forgotten password

5.1. Go to https://delivery.scilifelab.se/ and click on "Forgot Password?". You should be redirected to the following page:

5.2. Fill in your email address and click on the "Request Password Reset" button. Only the used when registering should work. The following message should be displayed:

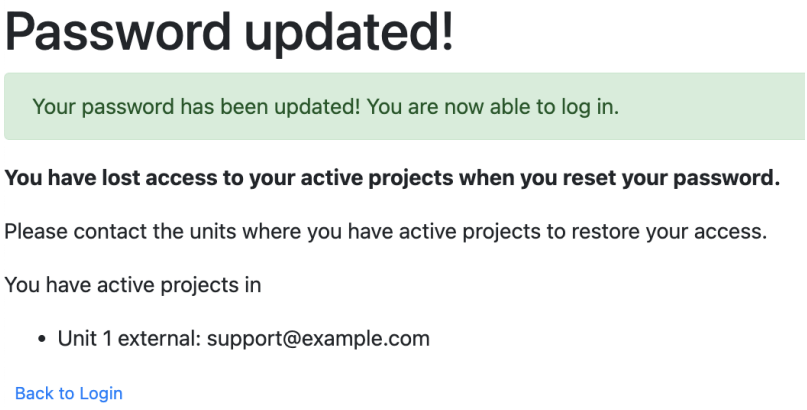
5.3. Go to your email inbox (or spam if you cannot find it in inbox) and open the email with the subject line "WARNING! Password Reset Request for SciLifeLab Data Delivery System". **Read the information in the email.**

5.4. Click the "Reset Password" button in the email. The following page should open:



5.5. Fill in a new password. Test both invalid and valid passwords, as in section 1. and 2. above.

5.6. When submitting the form, you should be redirected to the following page:



CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

INDEX

Symbols

D