CRIPT OPERATION MANUAL

Community Resource for Innovation in Polymer Technology

Add Pub number here

Intentionally Left Blank Page

Table of Contents

Introduction

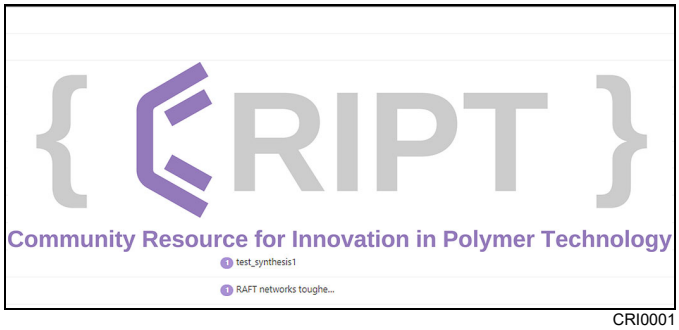
Initial Setup

Excel Uploader

Python SDK

INTRODUCTION

The Community Resource for Innovation in Polymer Technology (CRIPT) provides a platform for researchers working in polymer science and engineering to capture and share data. CRIPT is hosted by a team at the Massachusetts Institute of Technology (MIT) along with collaborators in academia, industry, and government, with support from the Convergence Accelerator Program of the National Science Foundation (NSF).



CRI0001

PREREQUISITES

Before attempting to access the CRIPT platform, users must verify their identity. Verification consists of obtaining a unique identification number from two third-party organizations. This ensures the user's identity while protecting their work.

The two organizations are: Open Researcher and Contributor ID (ORCID) and Globus ID.

ORCID

Open Researcher and Contributor ID (ORCID) is a global, not-for-profit organization. ORCID provides a persistent digital identifier (ORCID iD) that distinguishes one user from all other researchers.

To obtain an ORCID iD, visit the official ORCID website at **www.orcid.org.**

GLOBUS

Globus is a group at the University of Chicago that develops and operates a non-profit service for use by the research community.

Globus products and services are developed and operated by the University of Chicago and Argonne National Laboratory to assist researchers in sharing their work with universities, laboratories, and companies around the world.

A Globus ID allows users the ability to share research without providing private organizational or personal contact information or credentials.

To obtain a Globus ID, visit the Globus ID website at **www.globusid.org**. If the user does not have a Globus ID, they may create one as part of the CRIPT account creation process.

EDUCATIONAL/GOVERNMENTAL ORGANIZATIONS

Some unique identification credentials provided by universities and governmental agencies can be used to access the CRIPT platform. These must be obtained from the user's institution.

CONTACT TECHNICAL SUPPORT

This manual was created to instruct users on the proper usage of the CRIPT platform. Additional support may also be found using the following resources:

- The CRIPT blog can be found on the CRIPT platform home page at **blog.criptapp.org**. It contains helpful usage tips.
- "Share a Thought" links are featured on many CRIPT platform pages. Clicking on these links allow users to comment and voice their concerns regarding CRIPT platform usage.
- Email requests for help can be sent to **CRIPT@mit.edu.**

Page Intentionally Blank

Chapter 1

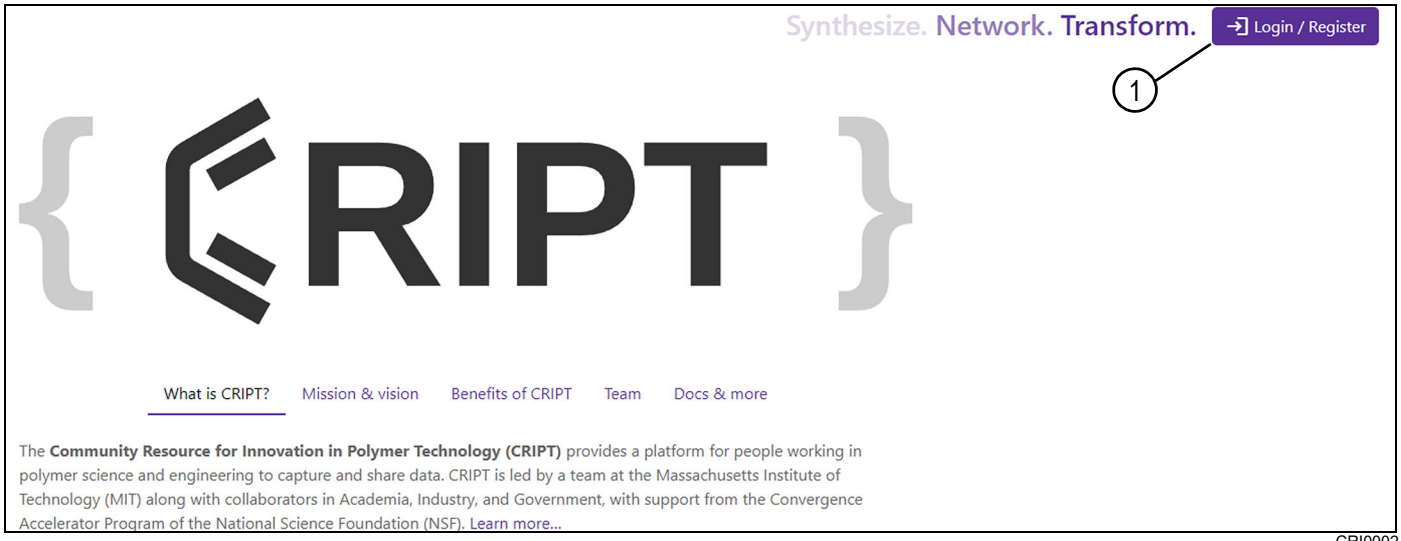
Initial Setup

INITIAL SETUP

REGISTRATION WITH CRIPT

Before registering with CRIPT, the user will require an ORCID identification number (ID). For more information, see “Prerequisites” on page i-i in the Introduction.

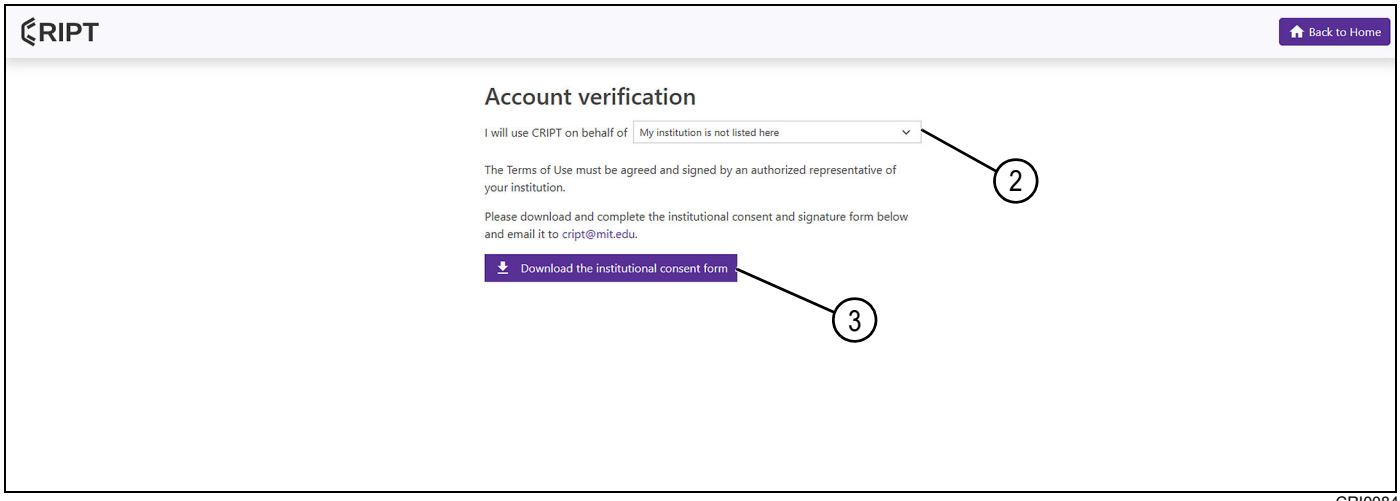
1. Navigate to the CRIPT website, **https://criptapp.org**.
2. Select Login/Register (1) on top right corner of page.



CRI0002

Figure 1-1

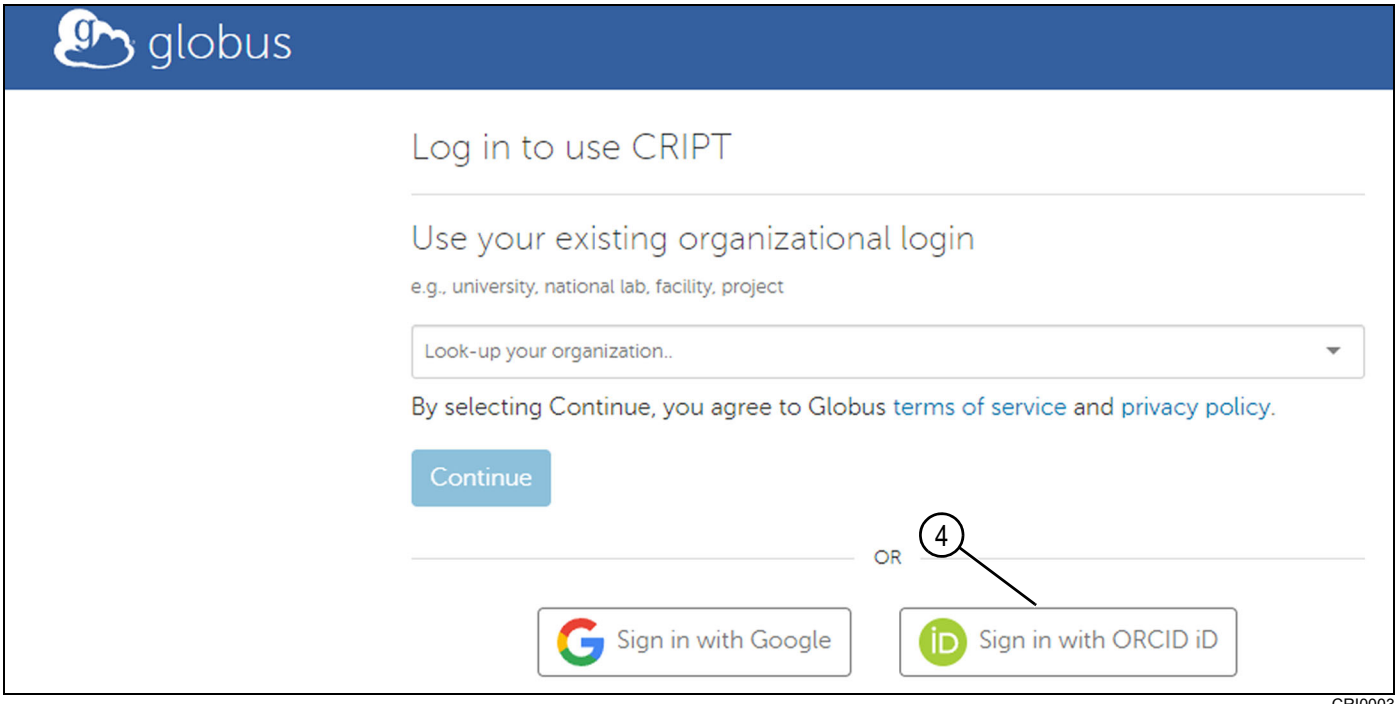
3. Use drop-down (2) to select institution. If institution is not listed, the consent form can be downloaded from the link (3). The link to the form can be downloaded from the hyperlink, **Institutional consent form.**
4. Complete the form and email back to **cript@mit.edu.**



CRI0084

Figure 1-2

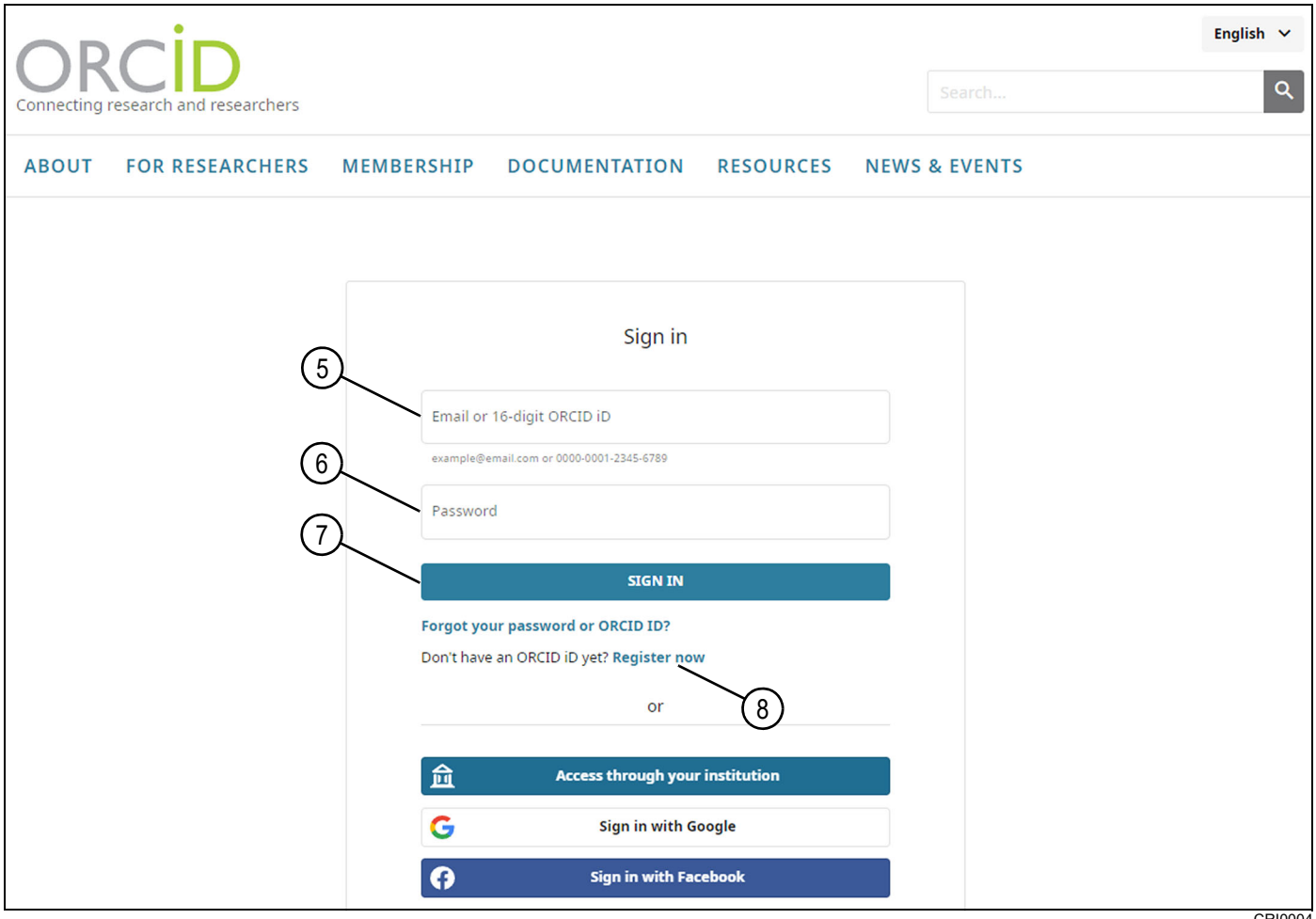
5. CRIPT will automatically launch Globus, which will offer several ways to enter the user's unique ID.
6. Select Sign in with ORCID iD (4) to launch the ORCID sign-in page.



CRI0003

Figure 1-3

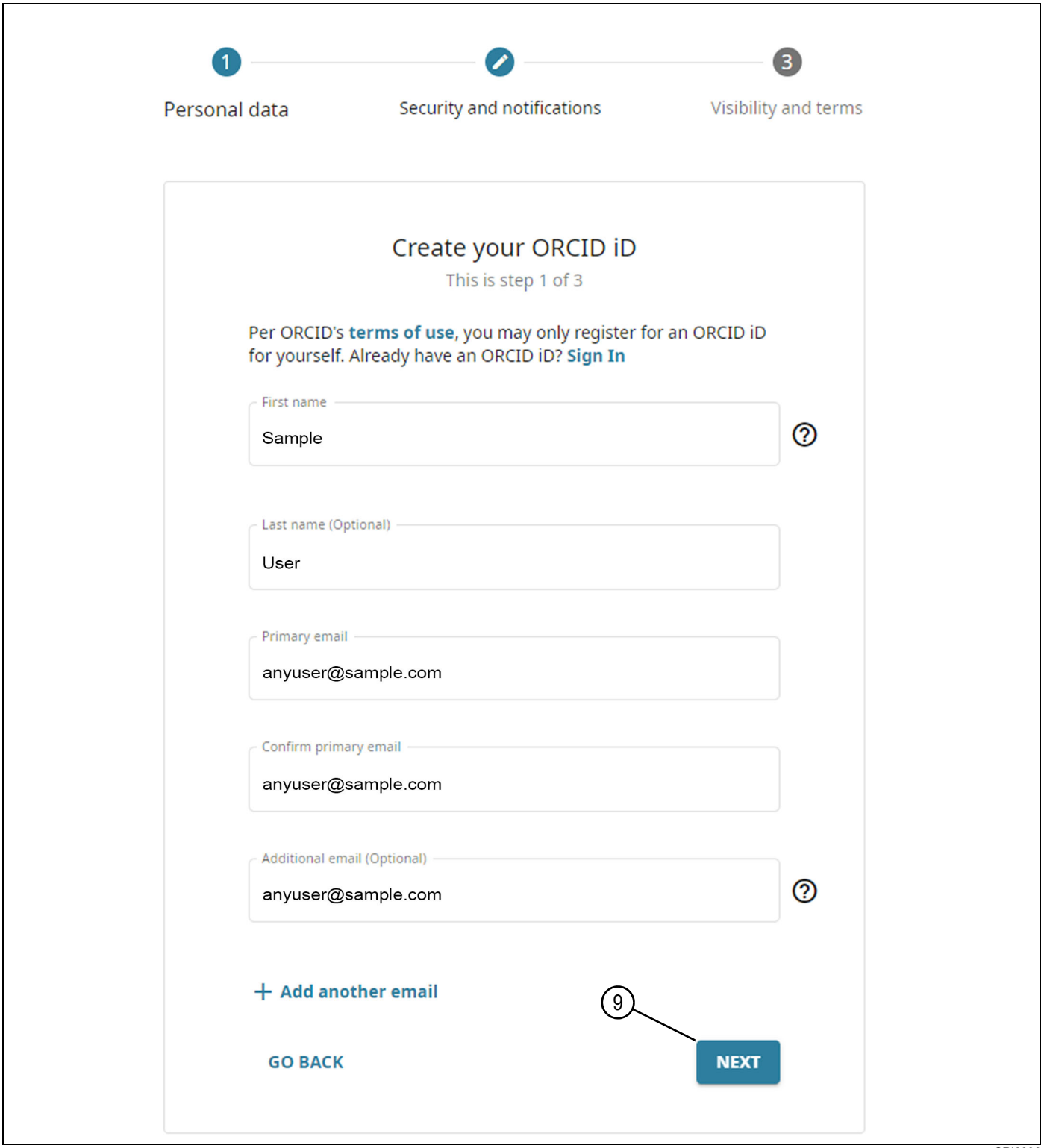
7. Enter the users Email or 16-digit ORCID iD (5) and Password (6), and select SIGN IN (7).
8. If lacking an ORCID iD, select Register now (8).



CRI0004

Figure 1-4

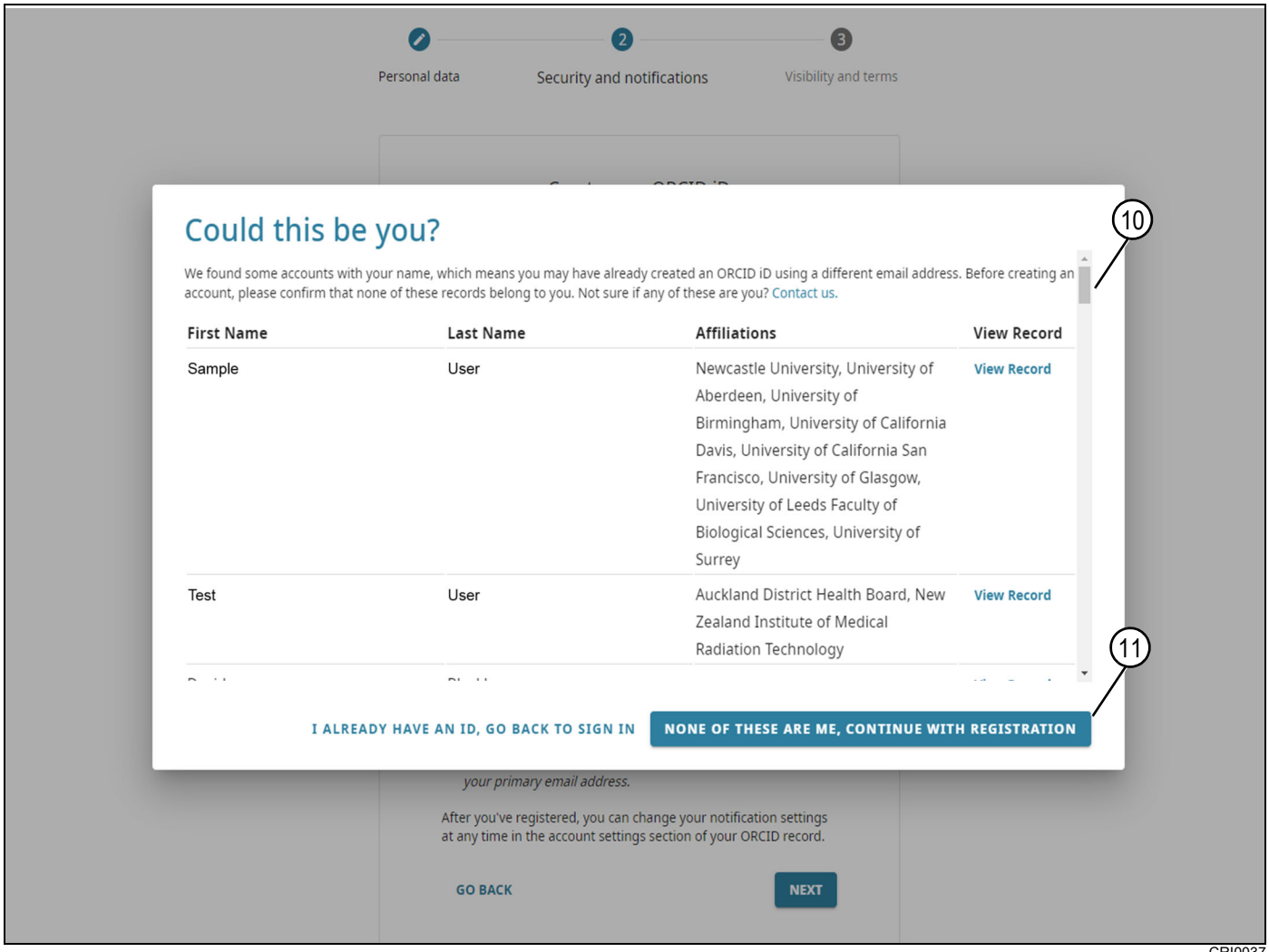
9. Enter user information and select NEXT (9).



CRI0036

Figure 1-5

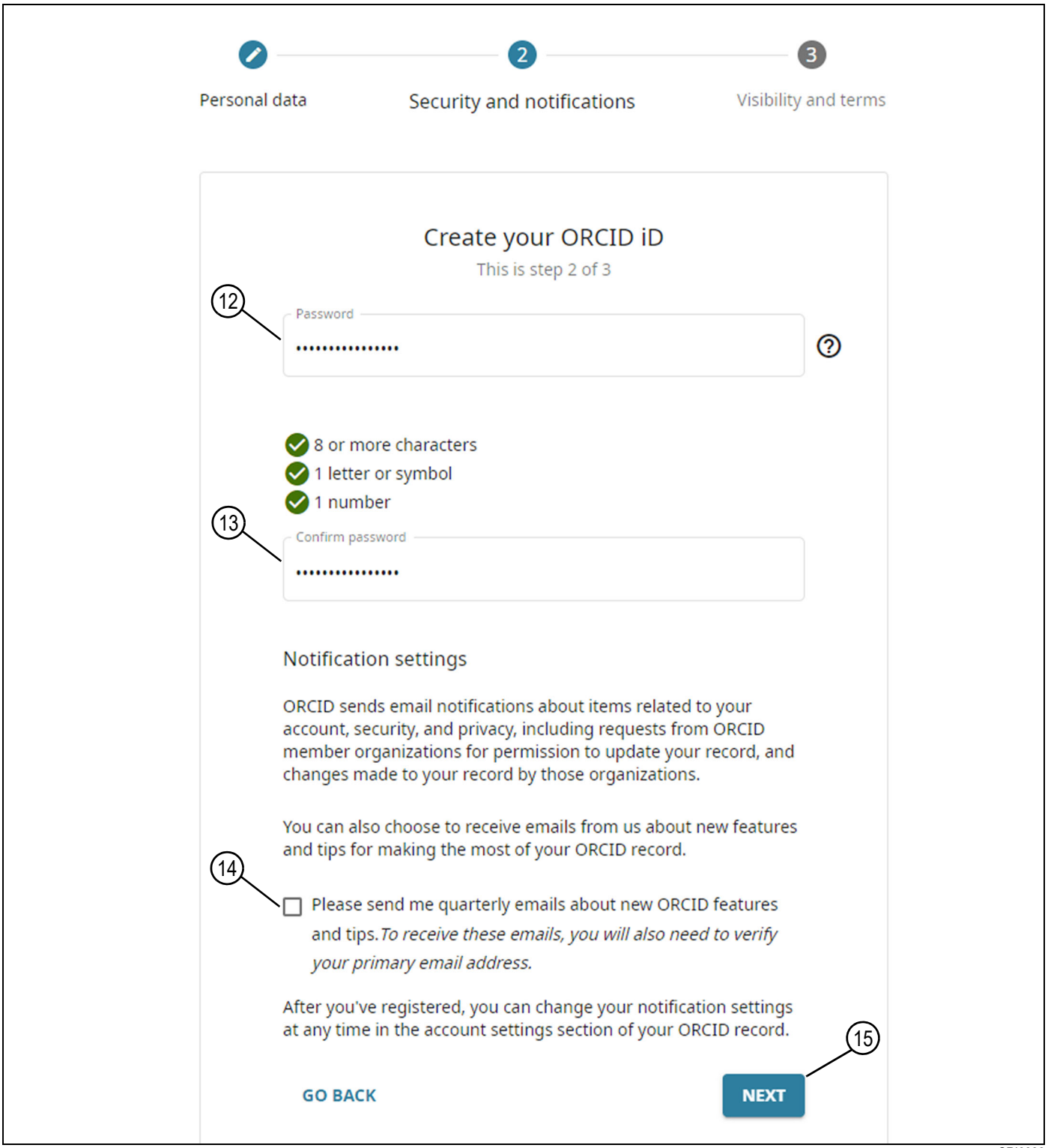
10. Verify that there is not an ORCID iD associated with user name by using the scroll bar (10) to scan through names.
11. Once confirmed, select NONE OF THESE ARE ME, CONTINUE WITH REGISTRATION (11).



CRI0037

Figure 1-6

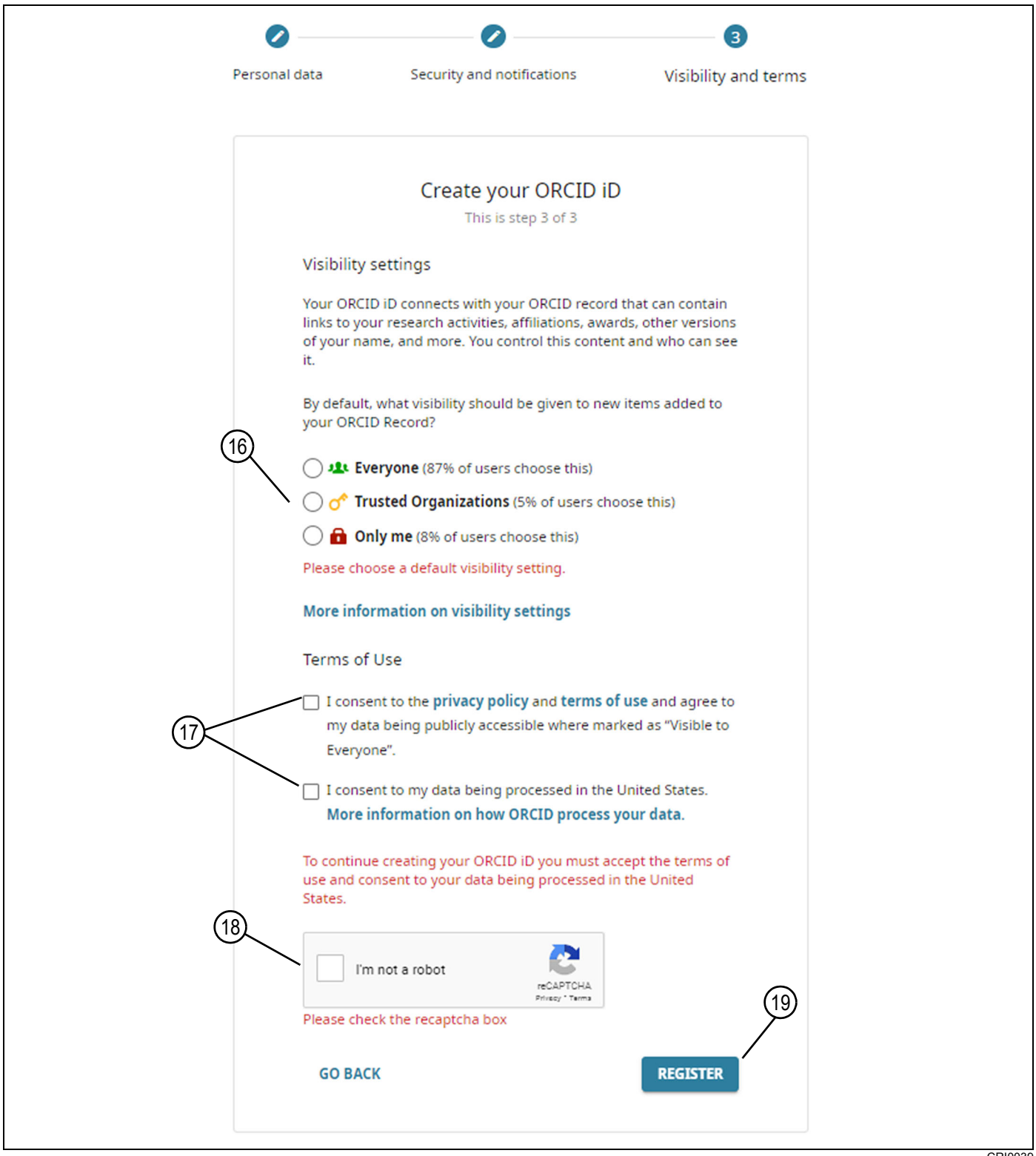
12. Create ORCID iD by entering a Password (12), Confirm password (13) by entering it again.
13. An optional check box (14) can be selected for quarterly notifications.
14. Select NEXT (15) to continue.



CRI0038

Figure 1-7

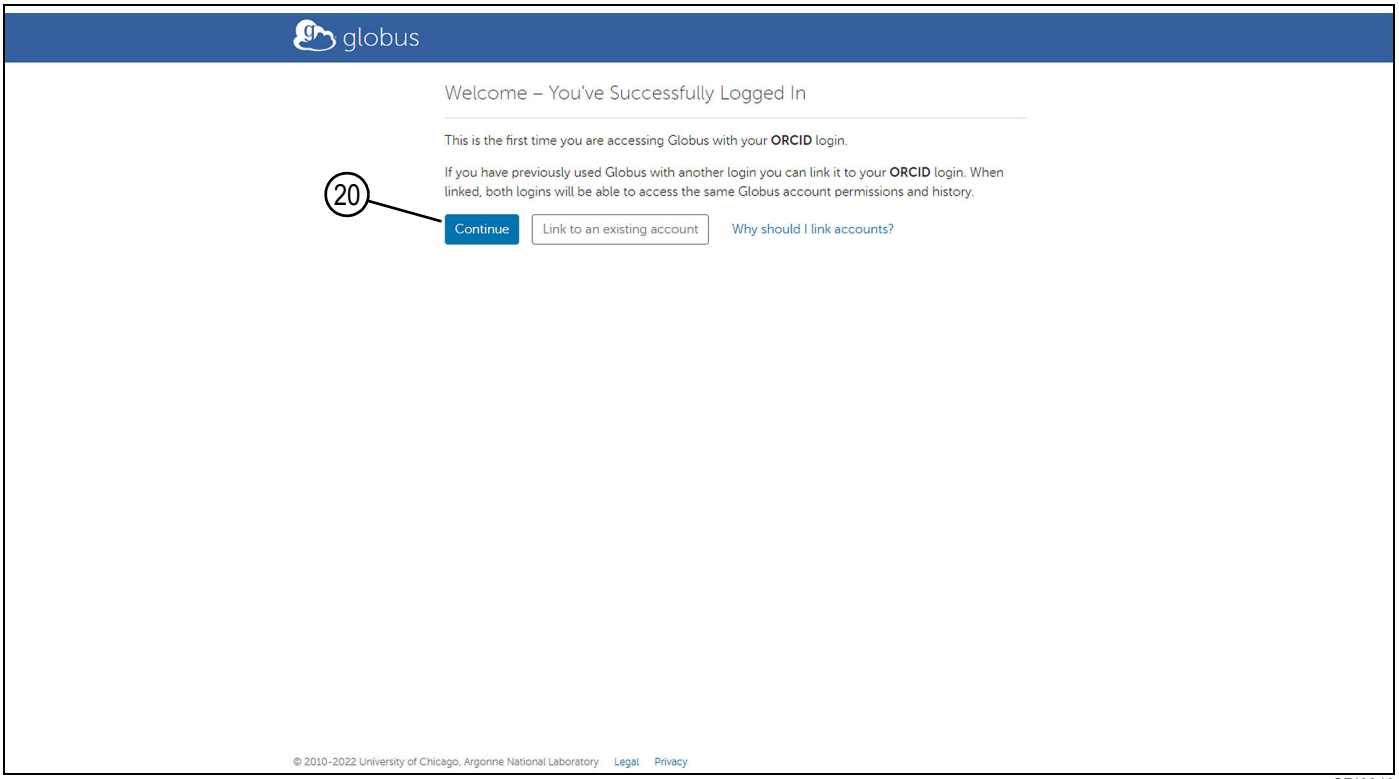
15. Set visibility settings (16) and Terms of Use (17).
16. Select the reCAPTCHA box (18) and then select REGISTER (19).



CRI0039

Figure 1-8

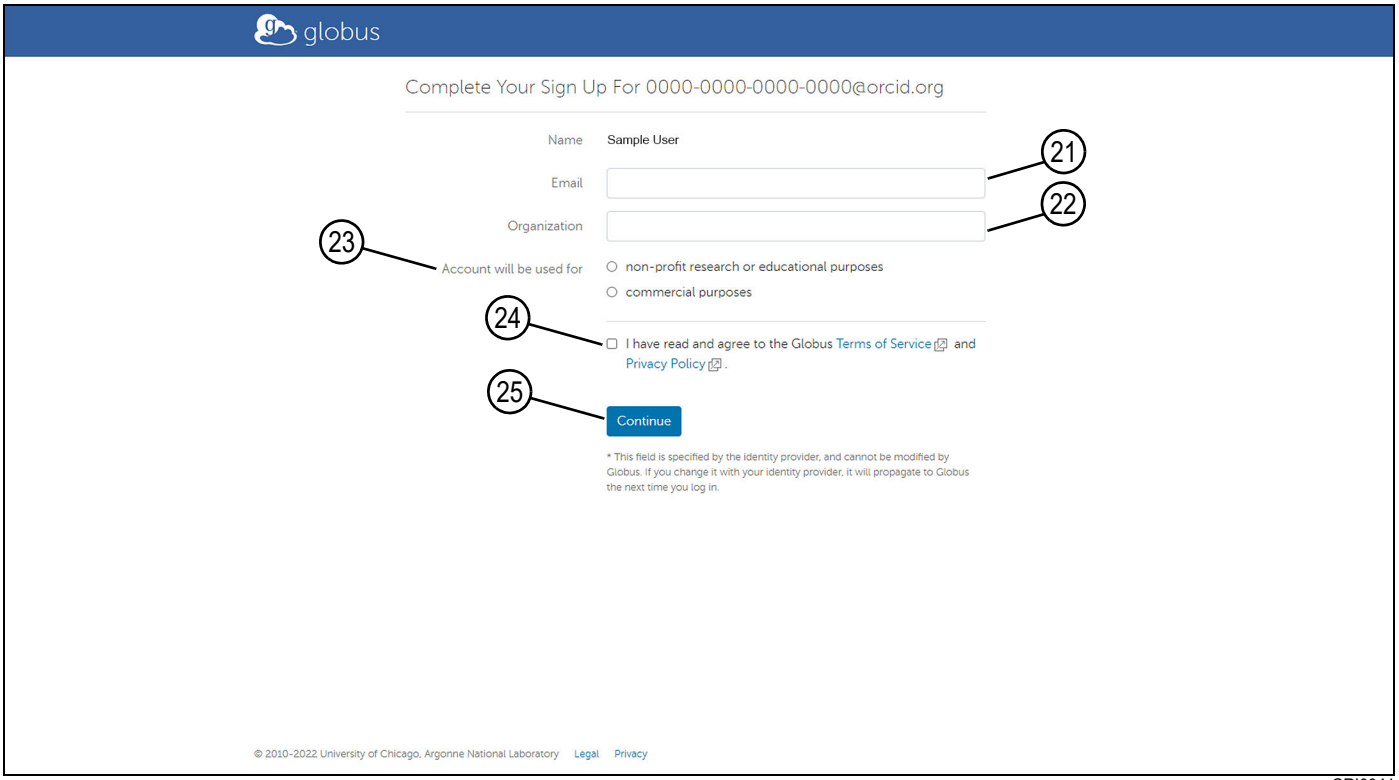
17. The globus site will launch. Select Continue (20).



CRI0040

Figure 1-9

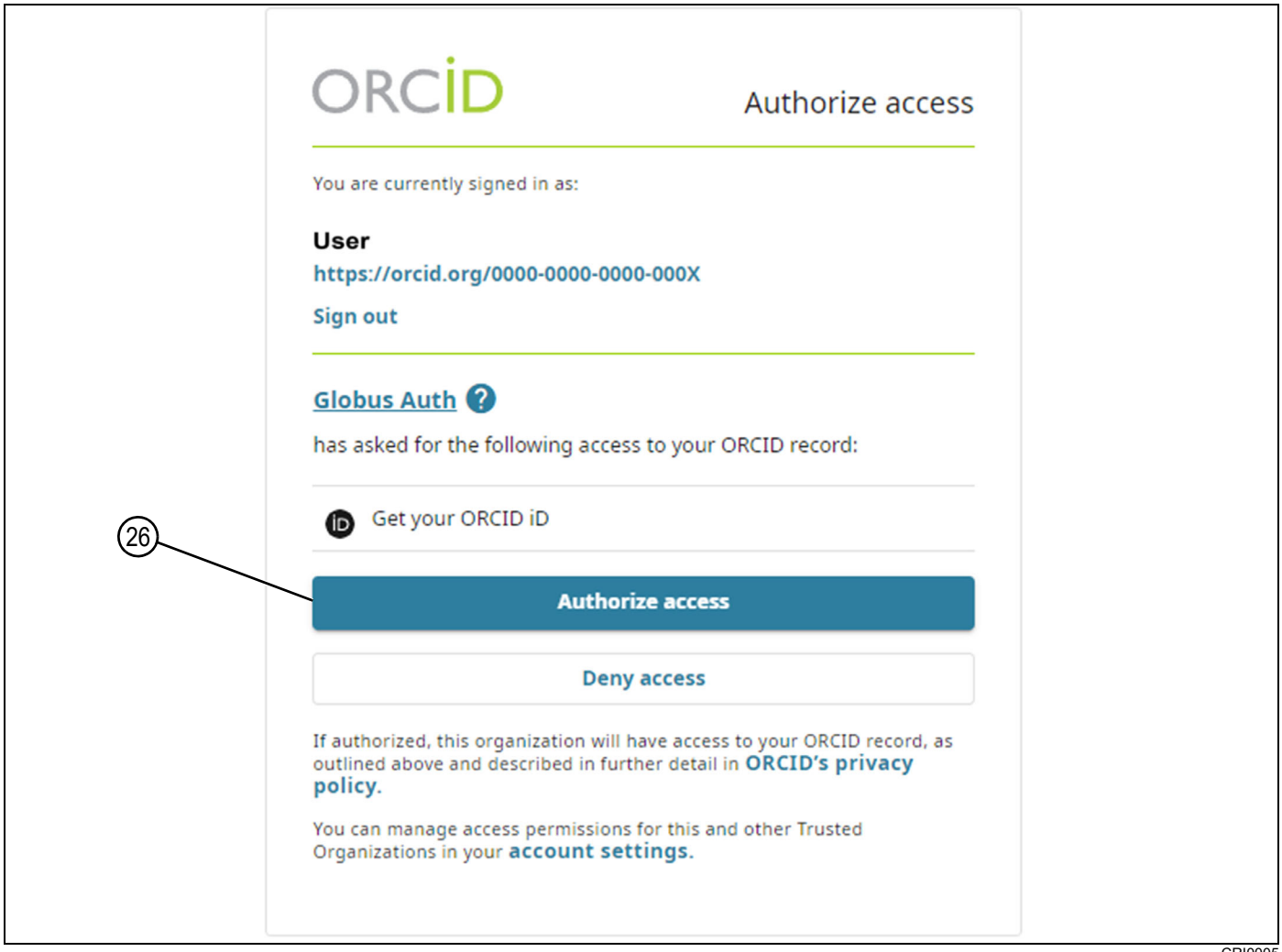
18. Enter Email address (21) and Organization (22).
19. Select which purpose the account will be used for (23).
20. Select Terms of Service (24) to accept Privacy Policy and then select Continue (25).



CRI0041

Figure 1-10

21. The ORCID Authorize access page will open, displaying the user's ORCID credentials. Selecting Authorize access (26) will launch the CRIPT home page.



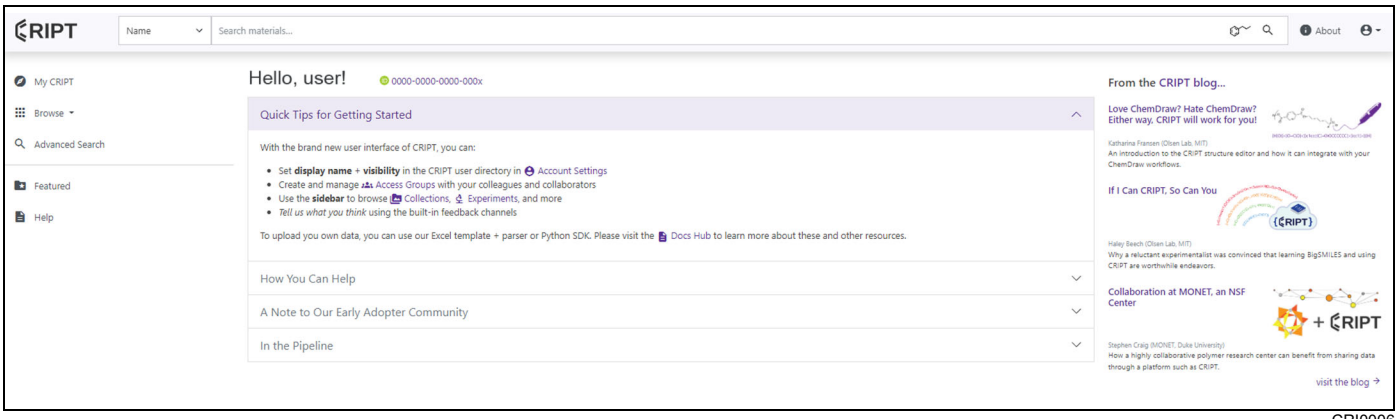
CRI0005

Figure 1-11

LOGIN TO CRIPT

The CRIPT home page allows the user to create new projects, search for existing projects, and find helpful tips on the CRIPT platform.

CRIPT HOME PAGE



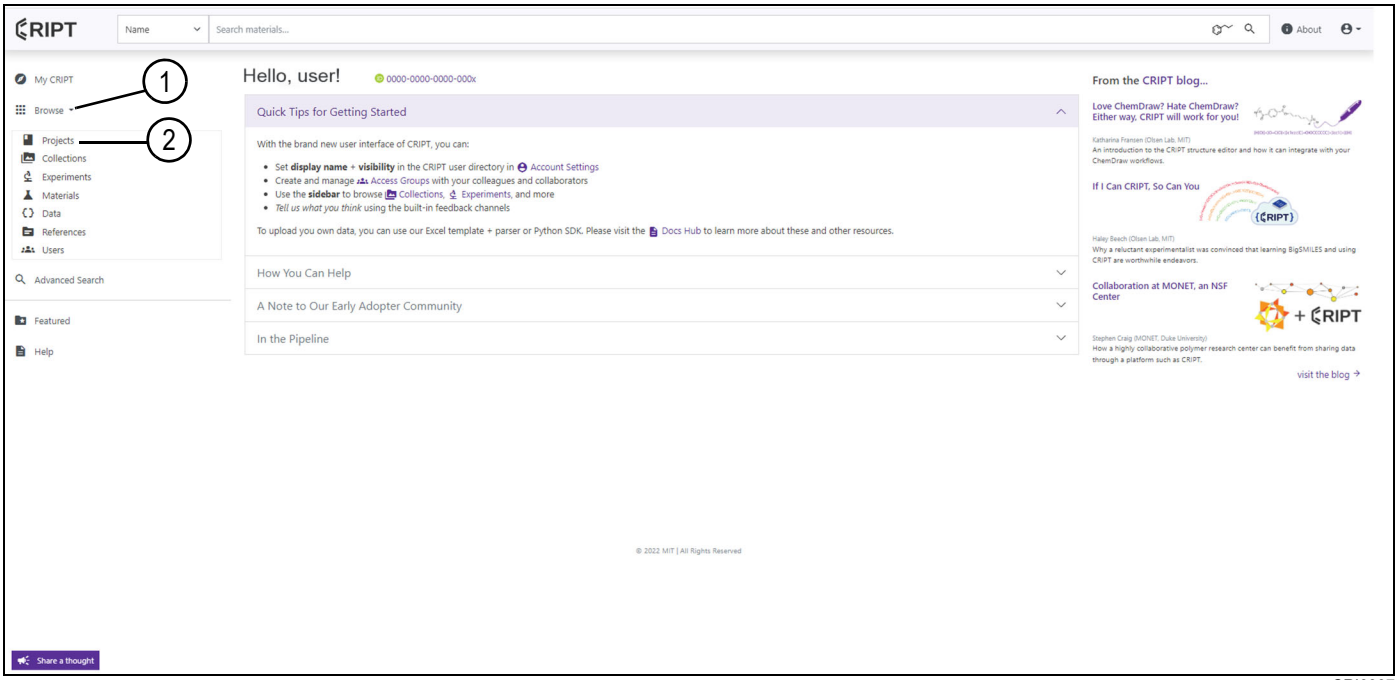
CRI0006

Figure 1-12

CREATE A PROJECT

The first step in working with user data in CRIPT is to create a project. A project is an organizational tool that will contain a collection of thematically related work that is typically a collaboration between a number of individuals and can lead to a series of scientific publications. The project will hold a set of collections, which are publishable groupings of research (i.e. papers) as well as a set of materials that are used between research performed within the project. By default, the user project will be non-public when first created. Therefore, it can only be viewed and modified by collaborators that are added to the project.

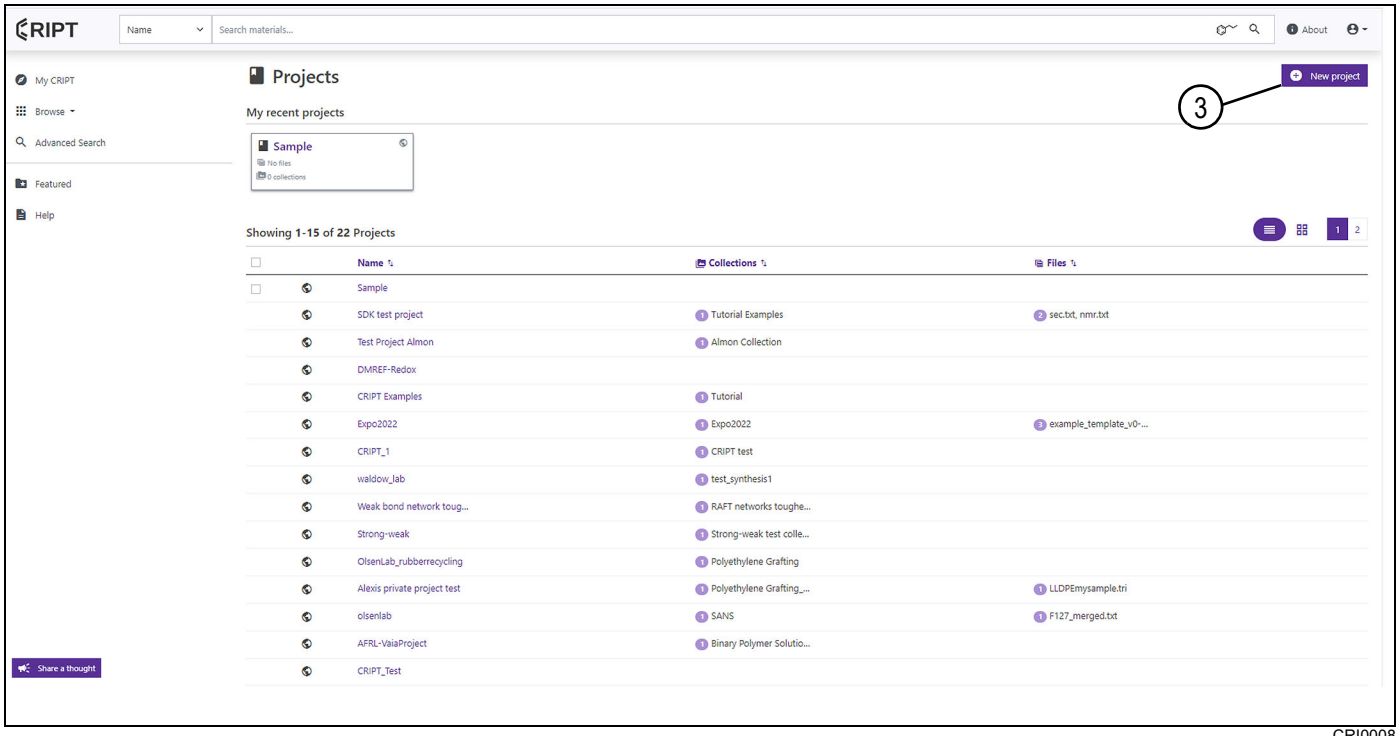
1. Select Browse drop-down (1) and select Projects (2).



CRI0007

Figure 1-13

2. Select New project (3).

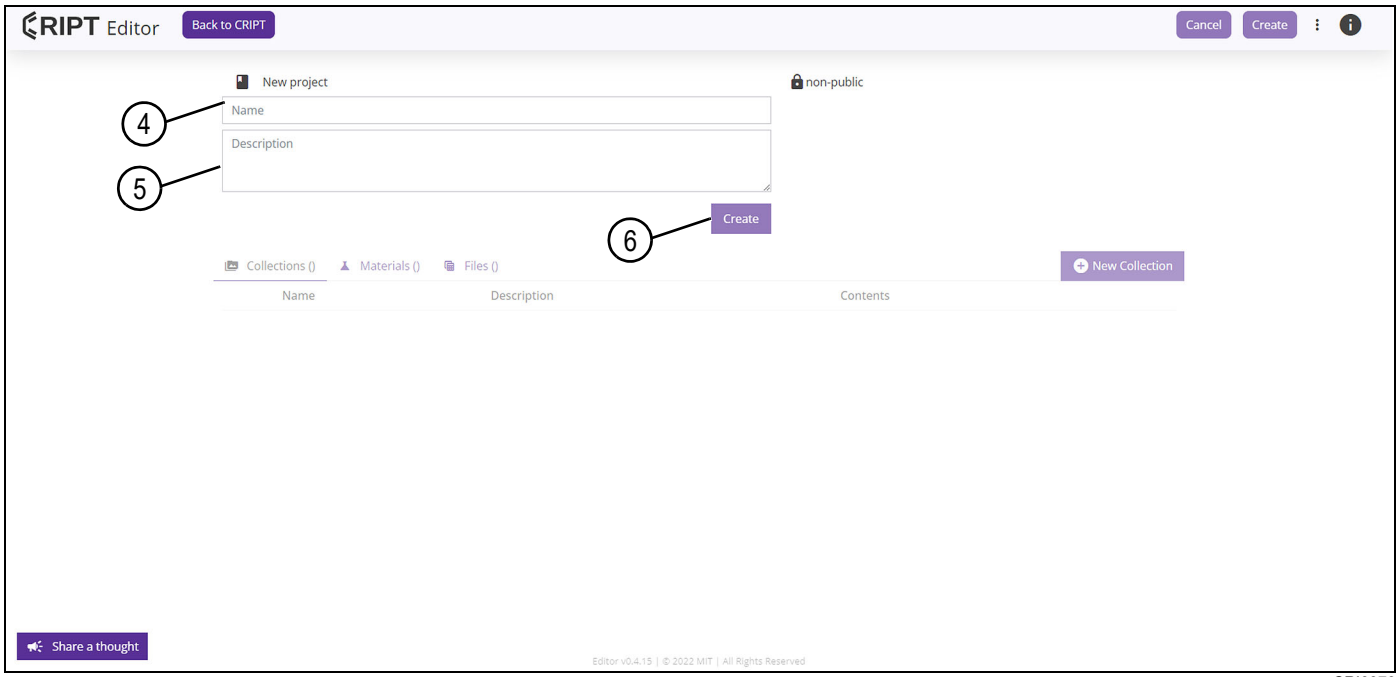


CRI0008

Figure 1-14

3. Enter a valid project Name (4).
4. Enter a Description (5) for the project.
5. Select Create (6).

NOTE: New projects are private by default and can only be viewed and modified by collaborators that are associated with the project.



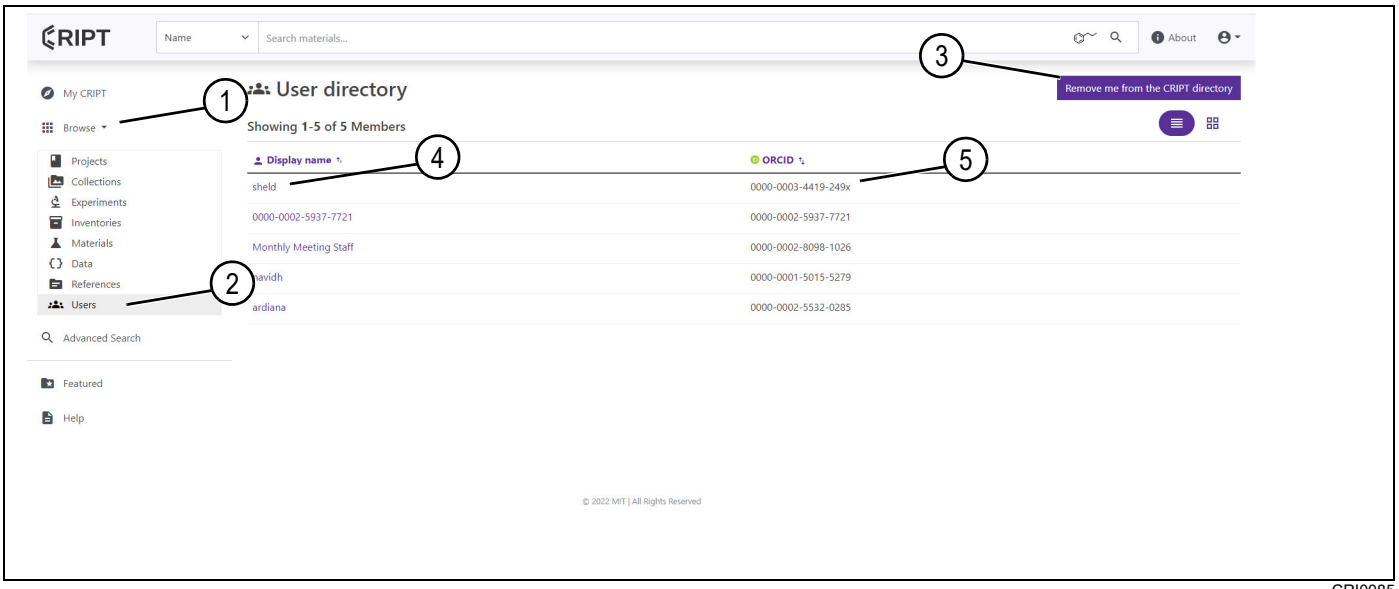
CRI0070

Figure 1-15

ADD COLLABORATORS TO A PROJECT

Once a project is created, specify which users can contribute to that project. Add all collaborators, including principal investigators. If a collaborator is not added, they will not be able to view any data in the project until it is public, and will not be able to modify any data.

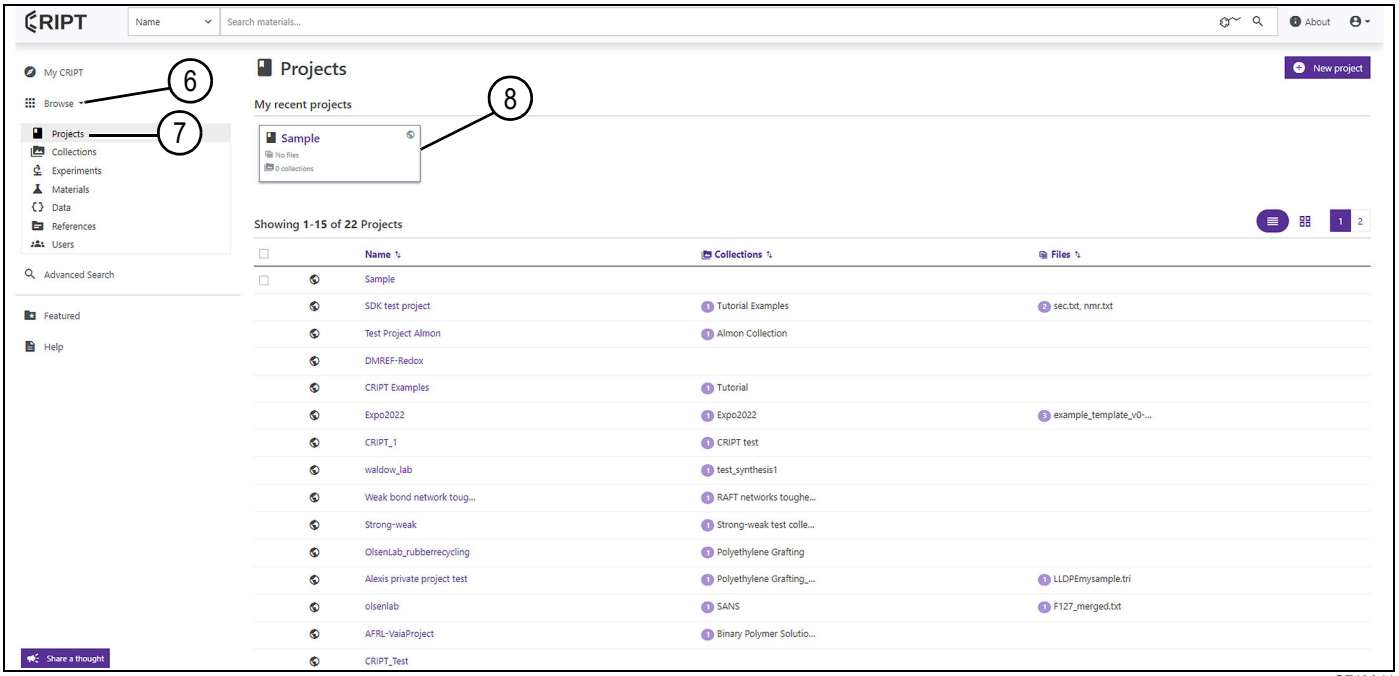
1. Select Browse drop-down (1) and select Users (2).
2. Check box (3) allows user to add or remove name from the CRIPT directory.
3. Locate the user name (4) for collaboration and copy their ORCID iD (5).



CRI0085

Figure 1-16

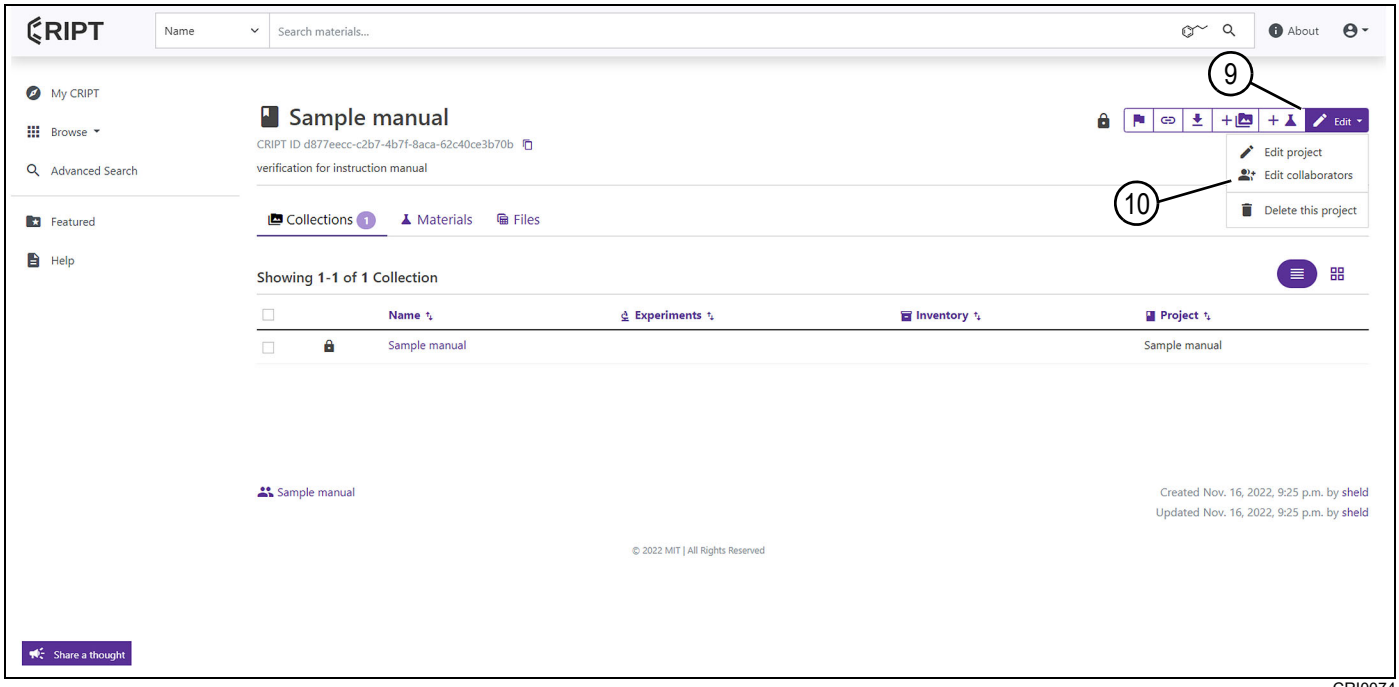
4. Select Browse drop-down (6), select Projects (7), and select project name (8).



CRI0011

Figure 1-17

5. Select Edit (9) and select Edit collaborators (10).

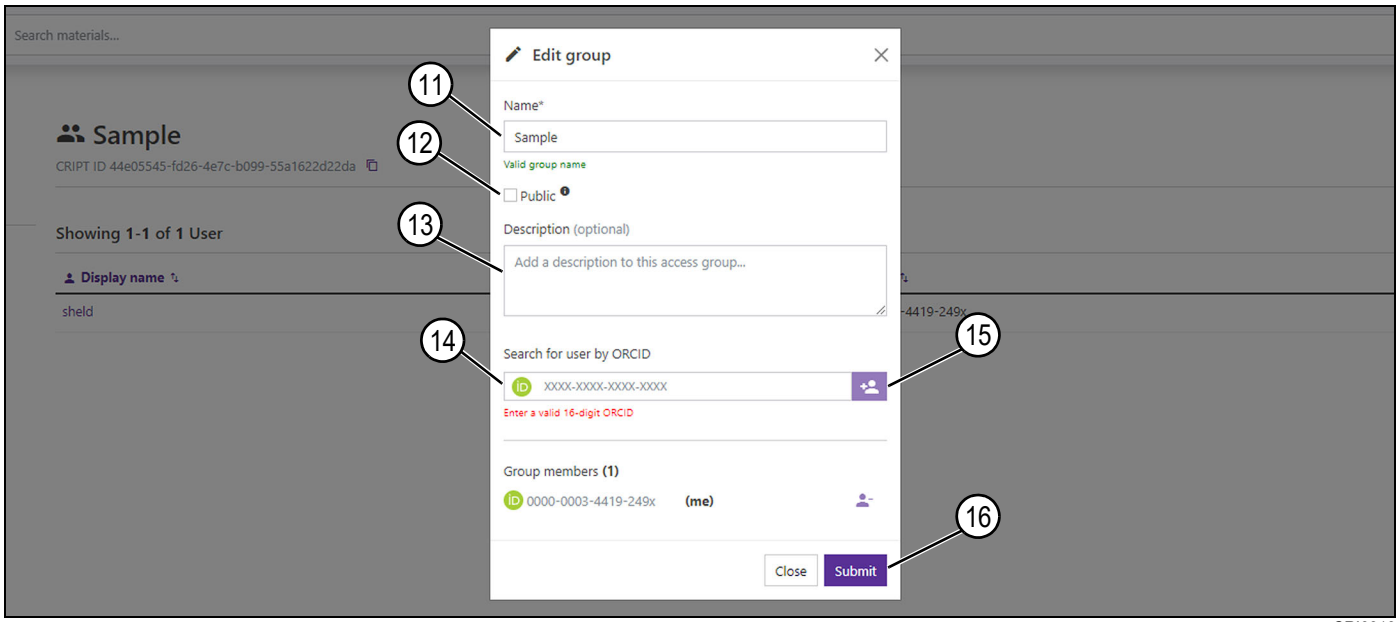


CRI0074

Figure 1-18

6. Name (11) will be autogenerated from the Project or user can enter a new name for the group.
7. Check the Public option box (12) if user wishes to make this access group visible to the community. For more information, see “Non-public change” on page 1-16.
8. Description (13) allows user to specify details about the project.
9. Enter ORCID iD (14) of the collaborator to be added, select add user icon (15), and select Submit (16).

NOTE: *If the ORCID iD contains any letters, they must be in lowercase.*

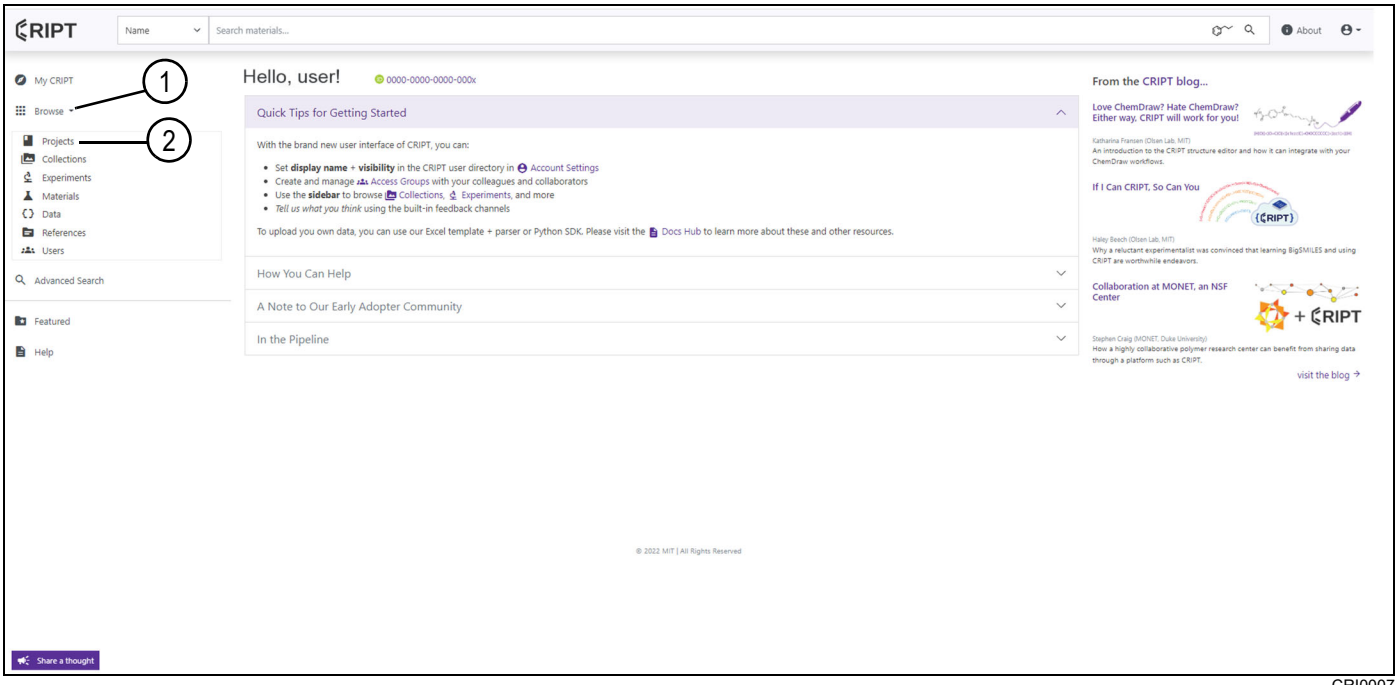


CRI0013

Figure 1-19

EDIT PROJECT

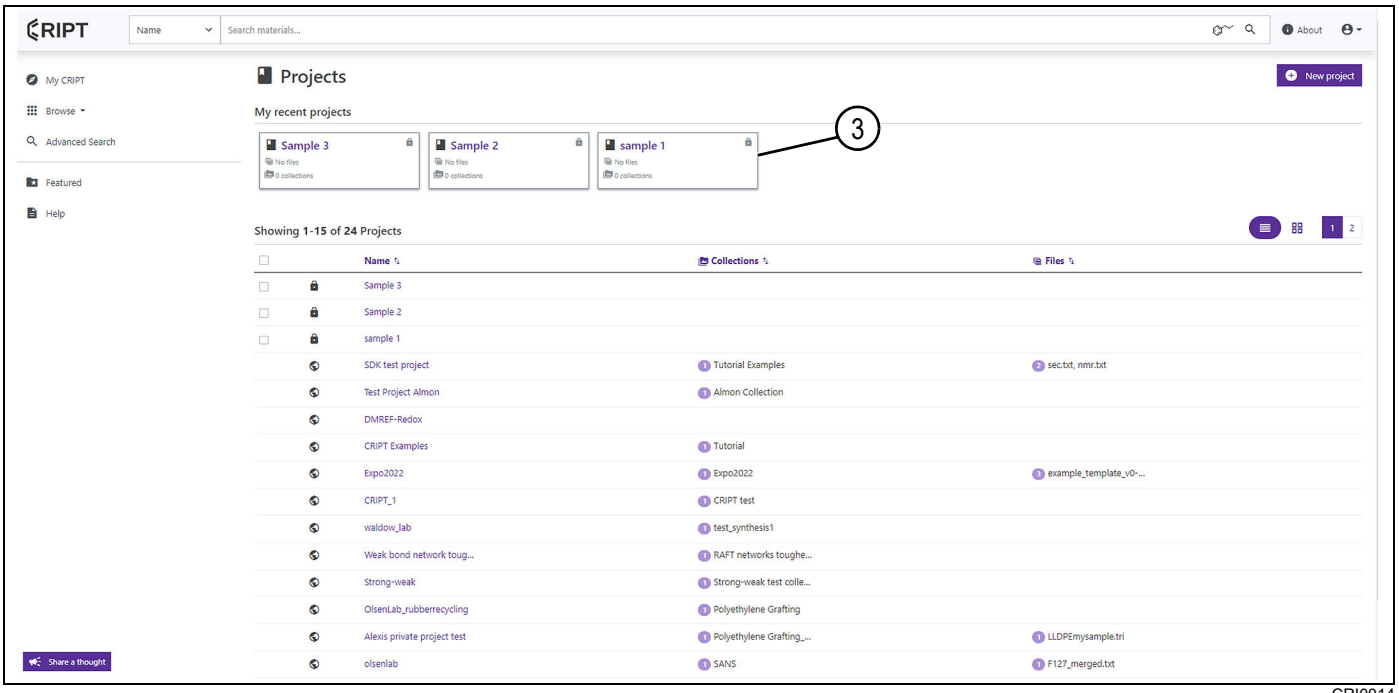
1. Select Browse drop-down (1) and select Projects (2).



CRI0007

Figure 1-20

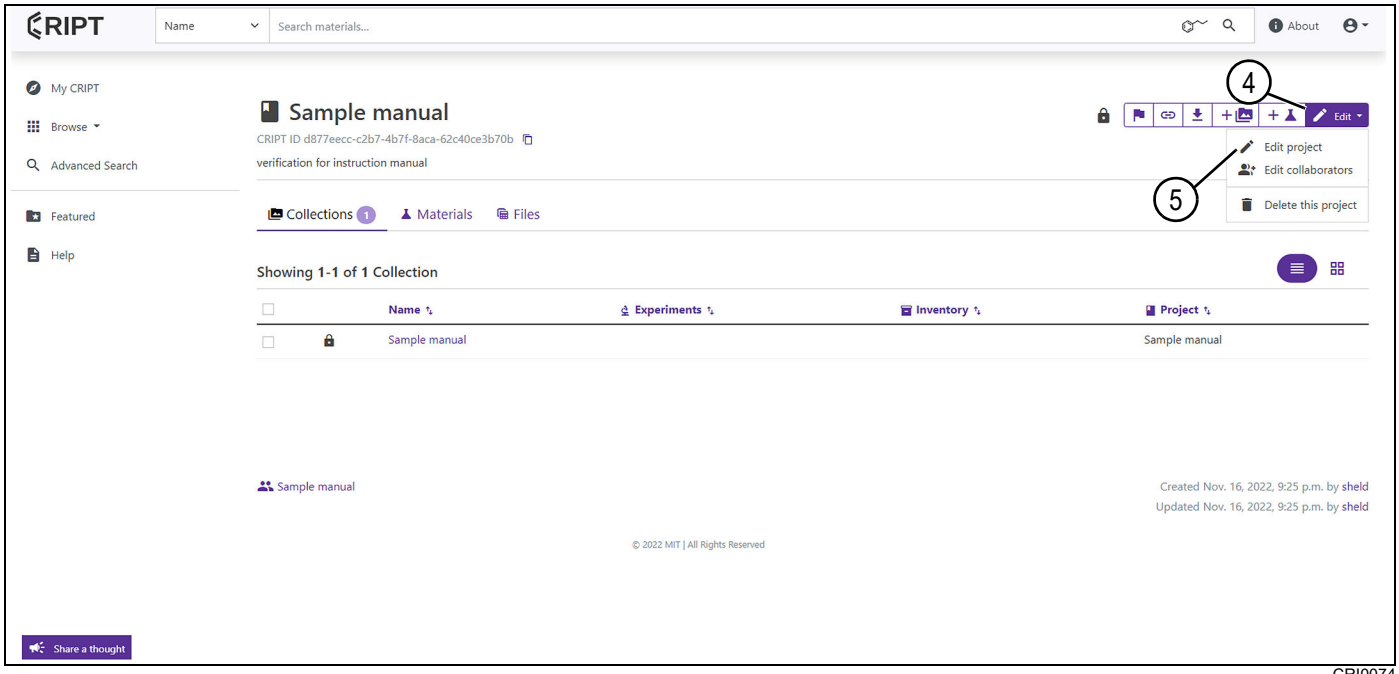
2. Select the project to be edited (3).



CRI0014

Figure 1-21

3. Select Edit (4).
4. Select Edit project (5).

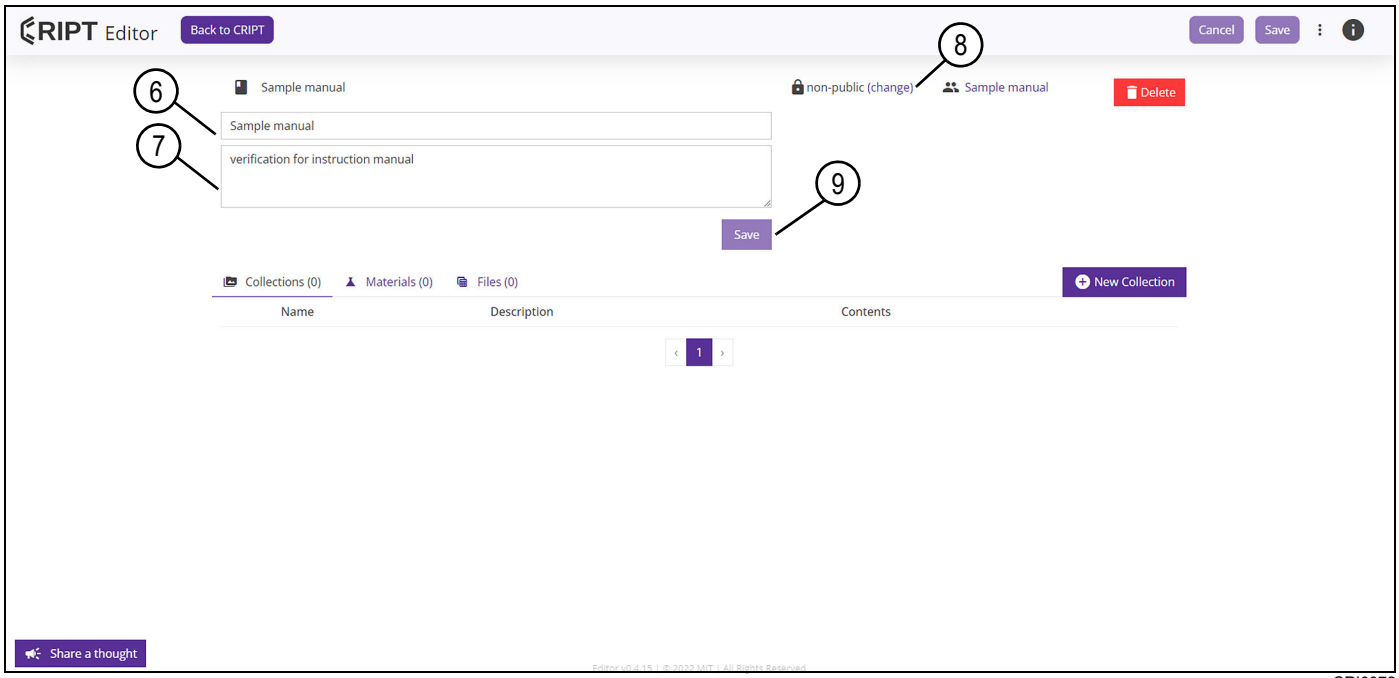


CRI0074

Figure 1-22

5. Users can edit the manual Name (6), project description (7), and non-public change (8). For more information on non-public change, see “Non-public change” on page 1-16.
6. Select Save (9) to save changes.

NOTE: New projects are private by default and can only be viewed and modified by collaborators that are associated with the project.



CRI0072

Figure 1-23

NON-PUBLIC CHANGE

1. Navigate to Edit project. For more information, see “Edit project” on page 1-14.
2. Select public (1) to make project visible to public.
3. Select Check box (2) to accept permission to continue.
4. Select Confirm (3) to save choice.

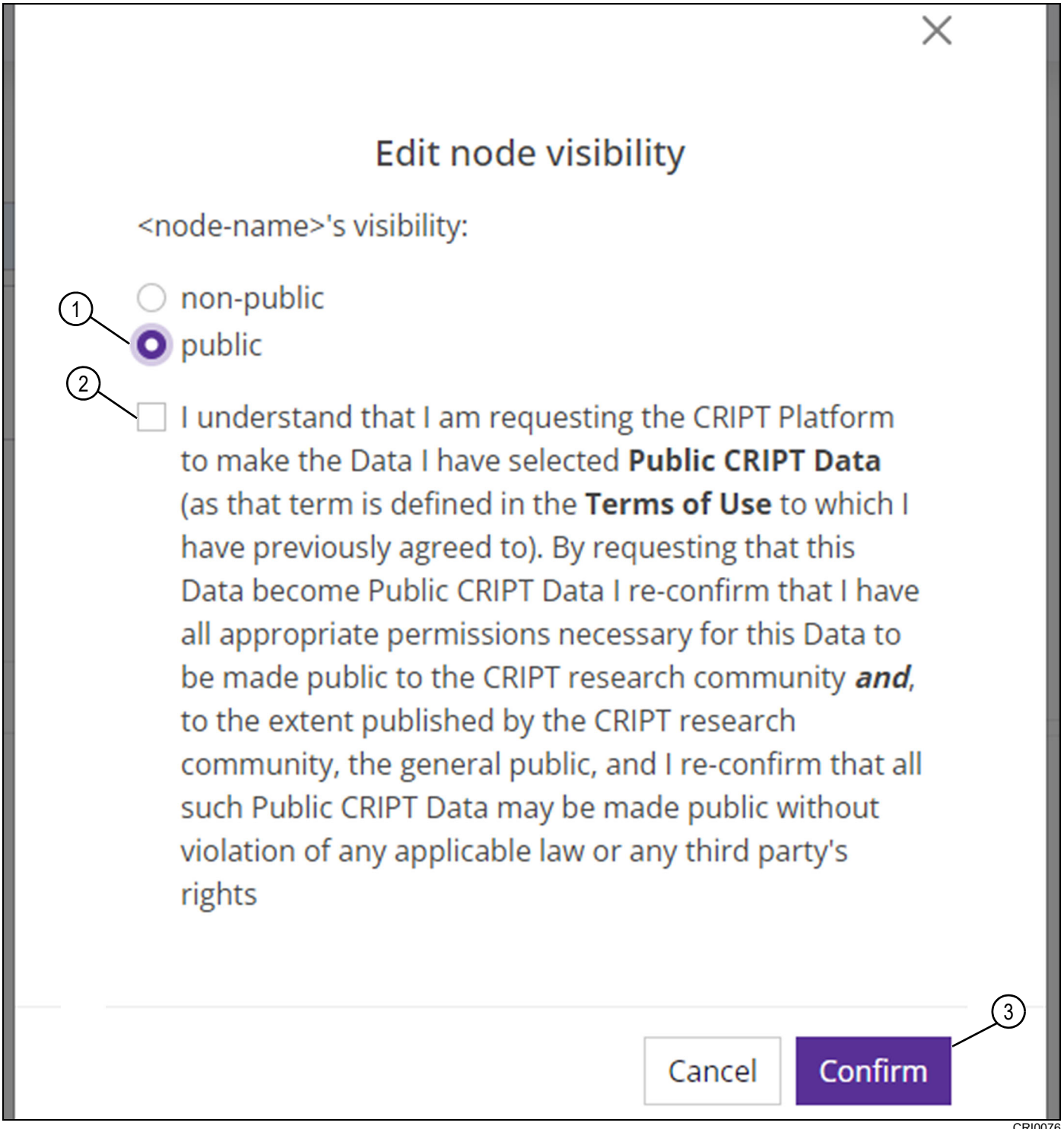
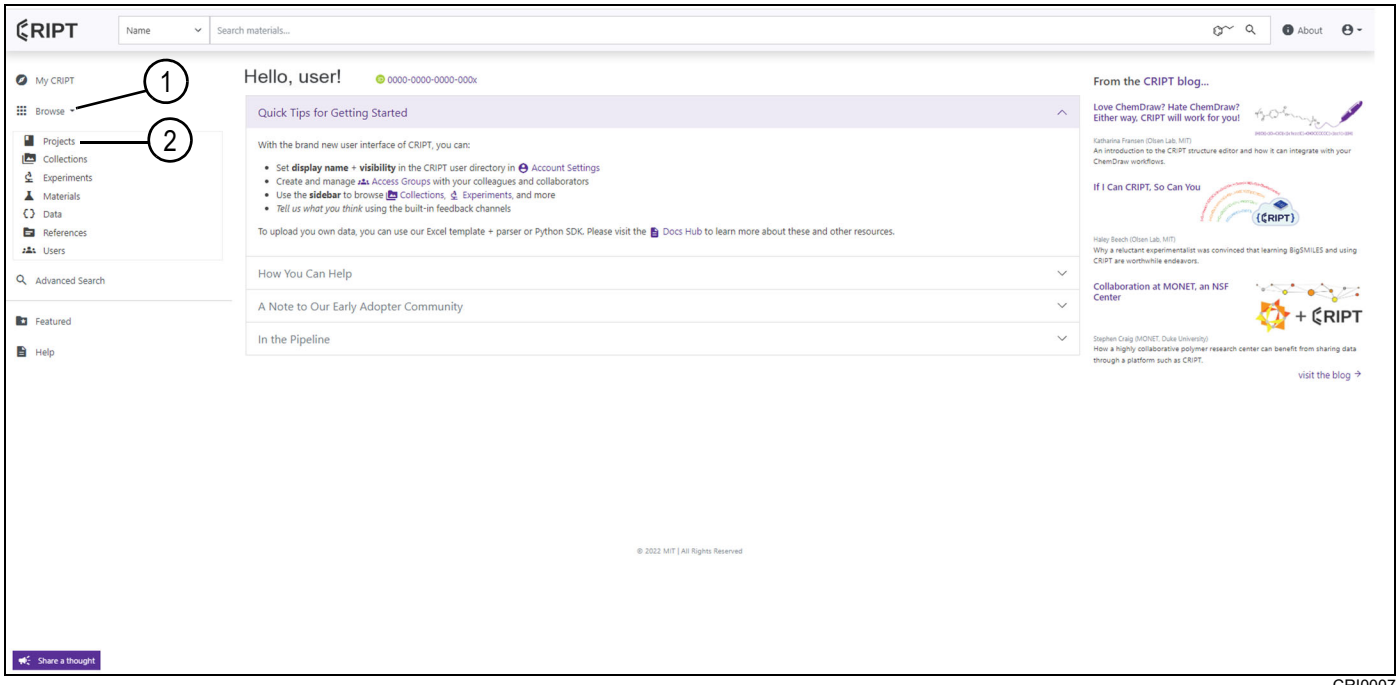


Figure 1-24

DELETE A PROJECT

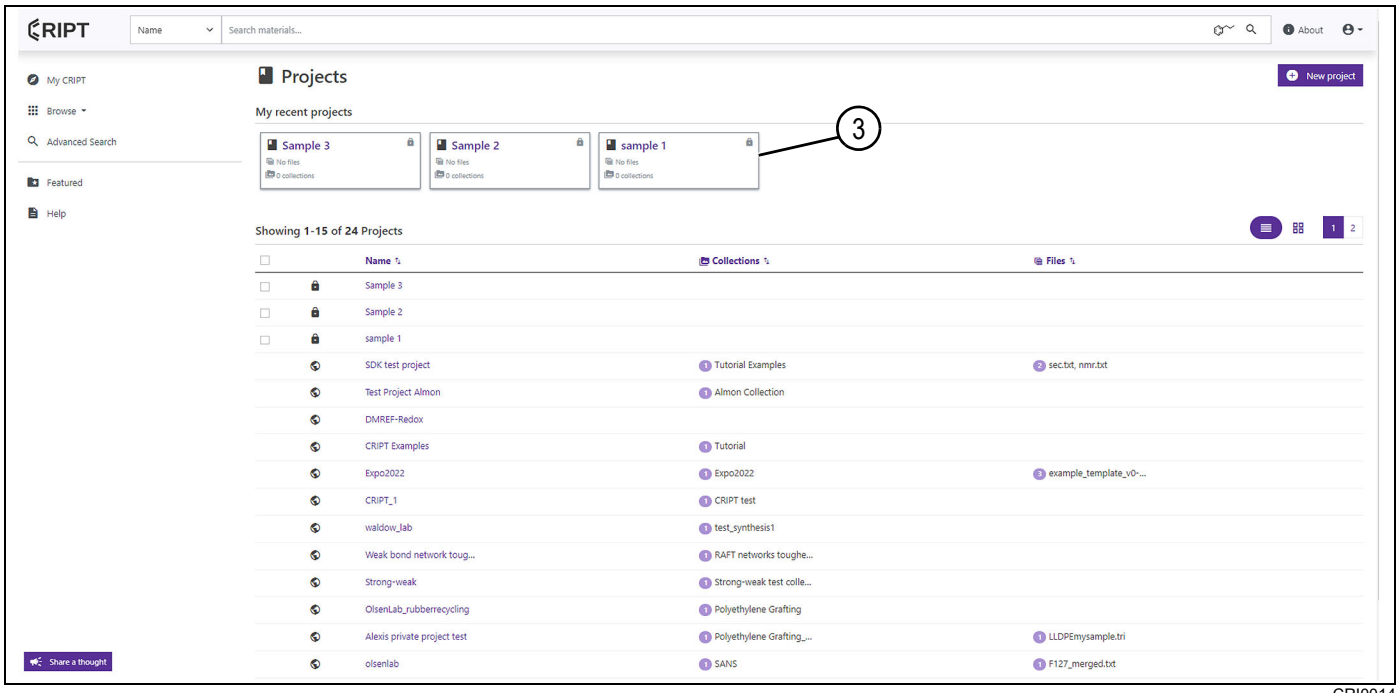
1. Select Browse drop-down (1) and select Projects (2).



CRI0007

Figure 1-25

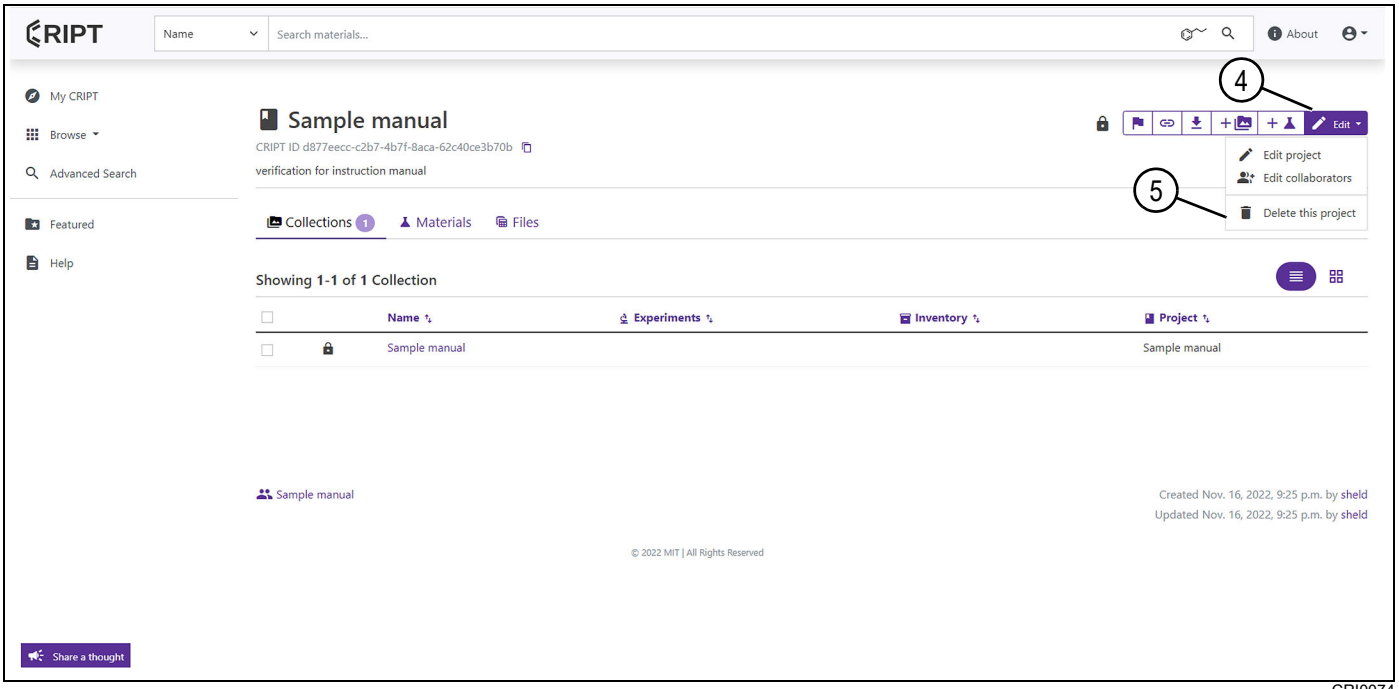
2. Select the project to be deleted (3).



CRI0014

Figure 1-26

3. Select Edit (4).
4. Select Delete this project (5).

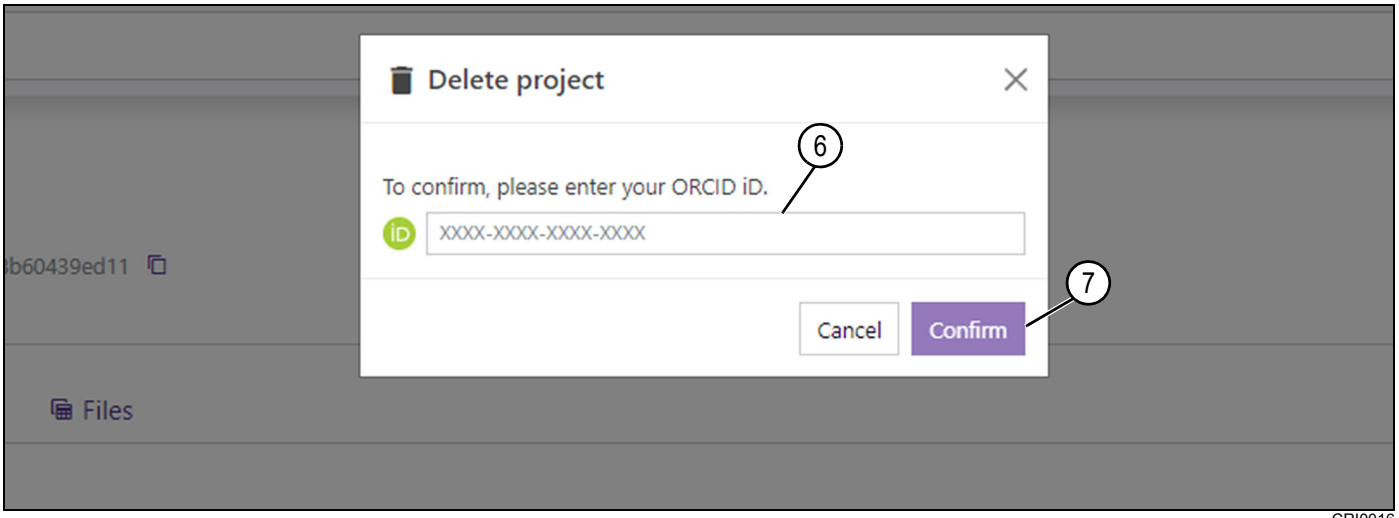


CRI0074

Figure 1-27

5. Enter ORCID iD (6) and select Confirm (7). For more information, see “Locate ORCID iD” on page 1-19.

NOTE: *If the ORCID iD contains any letters, they must be in lowercase.*

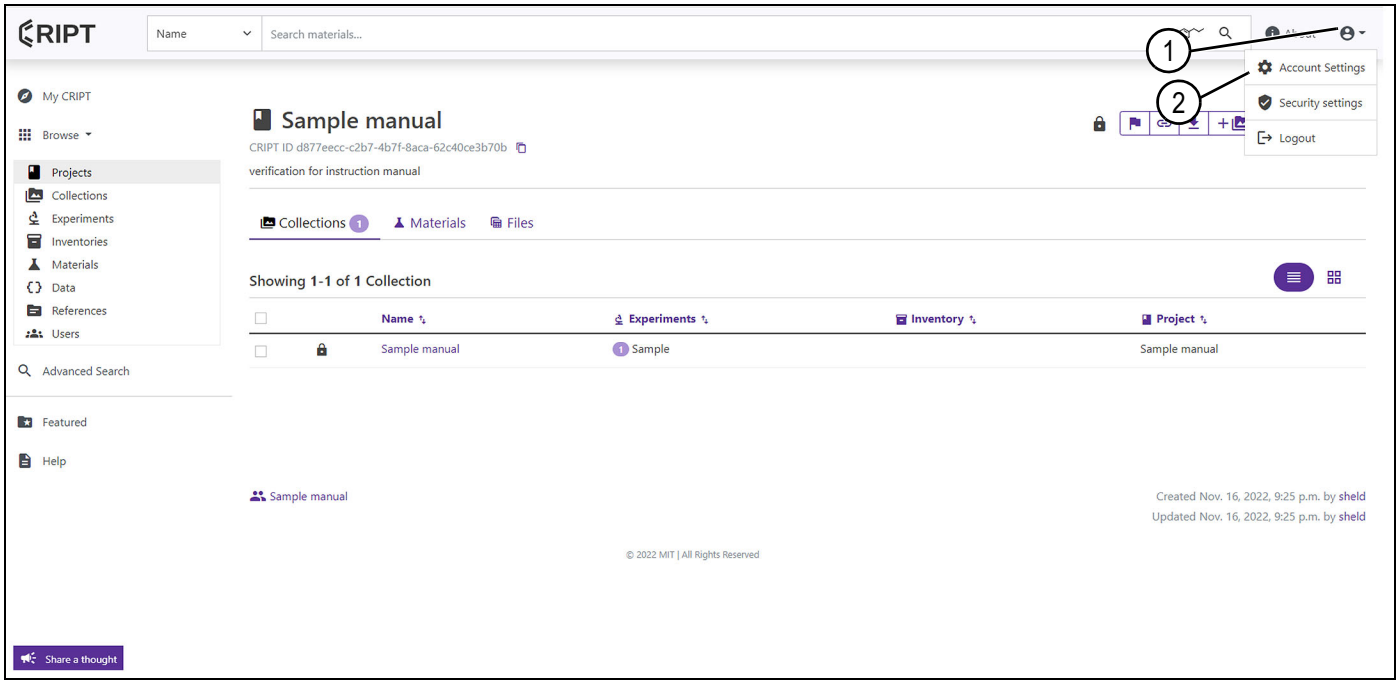


CRI0016

Figure 1-28

LOCATE ORCID ID

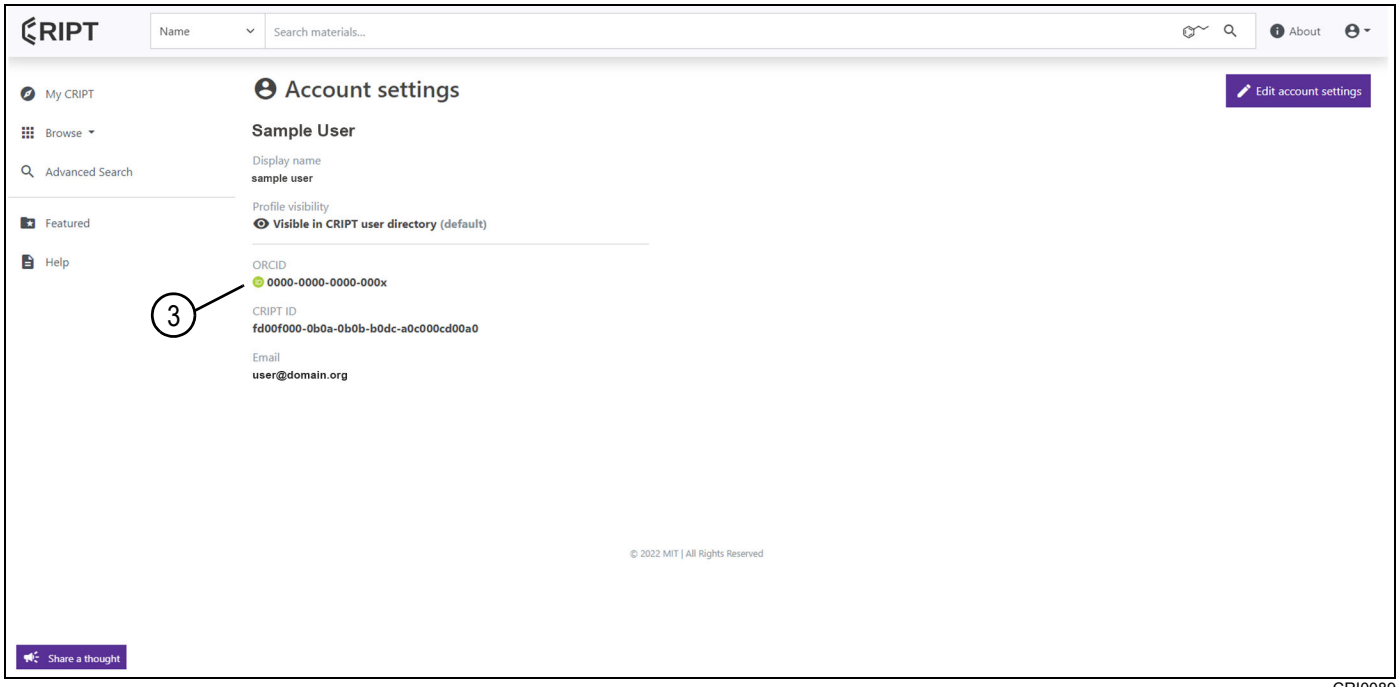
1. Select Account drop-down (1) and select Account Settings (2).



CRI0088

Figure 1-29

2. ORCID iD (3) location is shown.



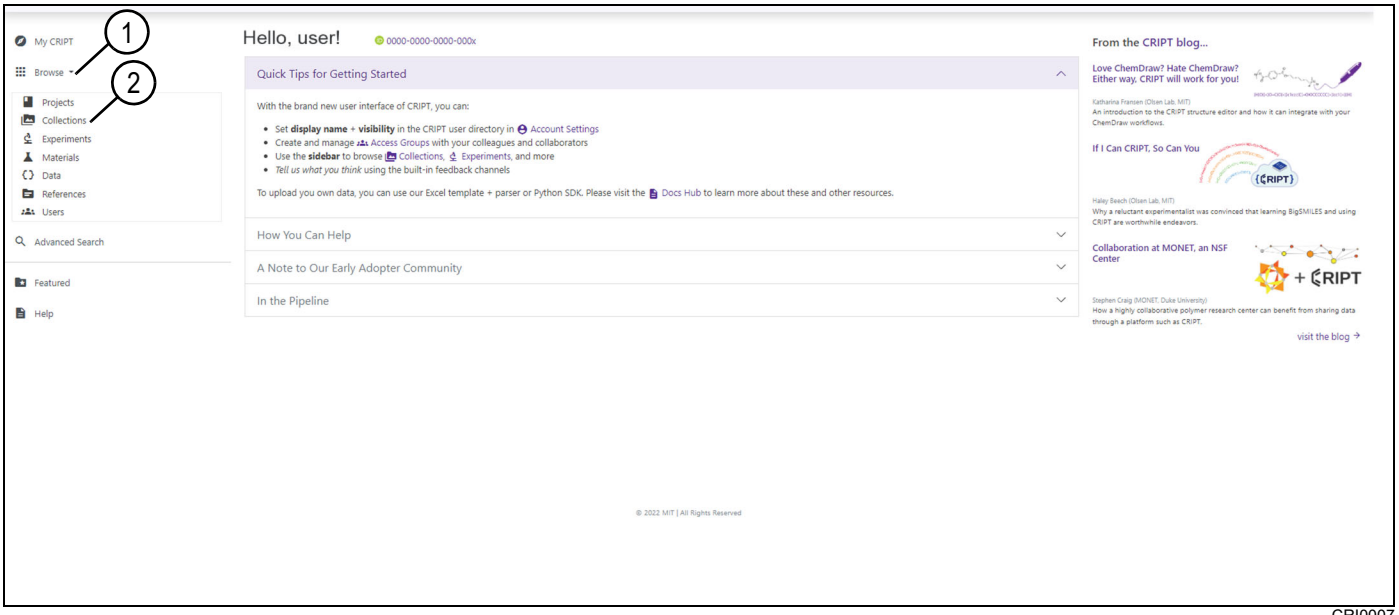
CRI0089

Figure 1-30

CREATE A COLLECTION

Each project within CRIPT holds both materials and collections. Collections are groupings of data that represent publishable units of research (i.e. one paper). Collections are the organizational level used to make data viewable to the public. All collections are by default non-public; the general public cannot view the users collection until it is made public at the project level. For more information, see “Non-public change” on page 1-16.

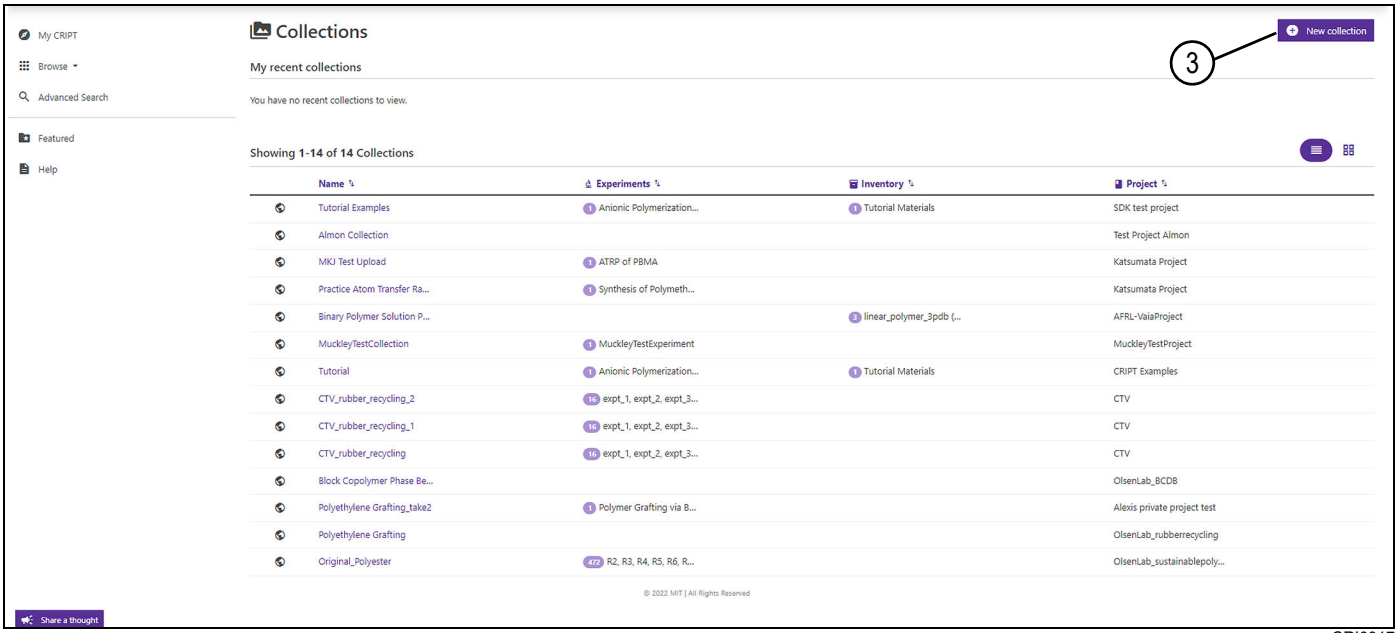
1. Select Browse drop-down (1) and select Collections (2).



CRI0007

Figure 1-31

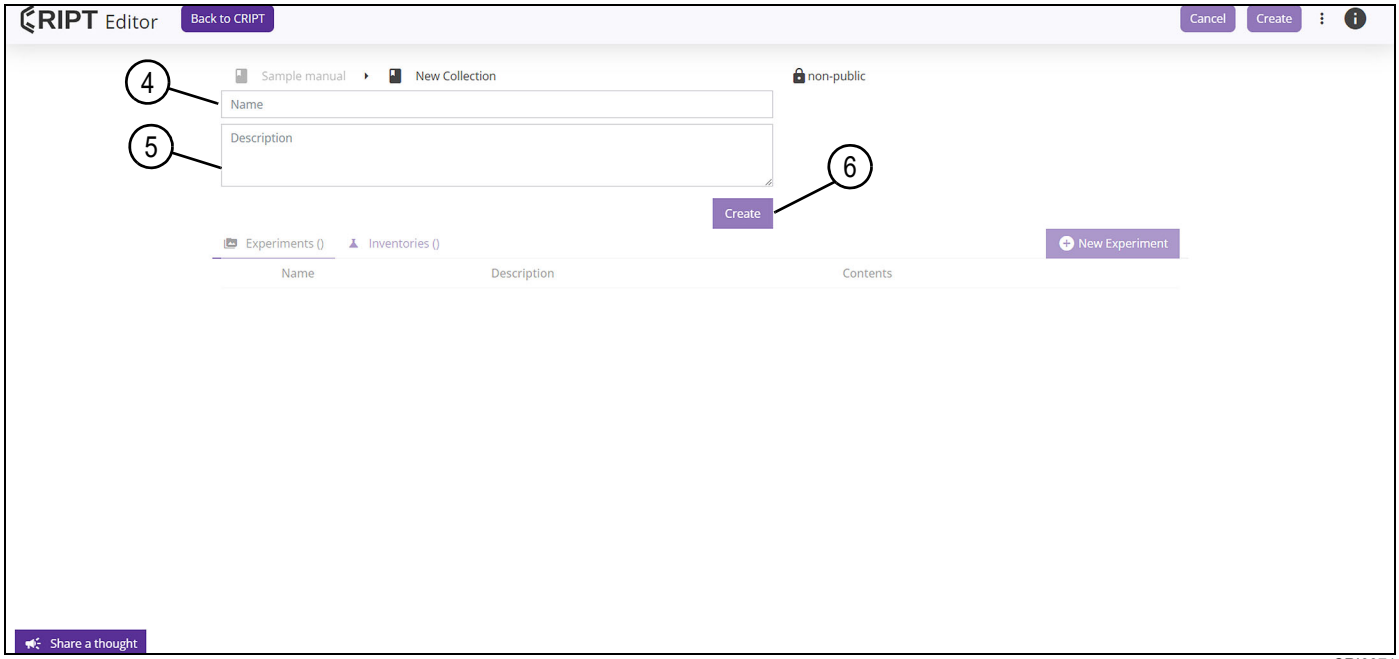
2. Select New collection (3).



CRI0017

Figure 1-32

3. Enter Name (4) for New Collection.
4. Enter Description (5) for New Collection and select Create (6) to save.

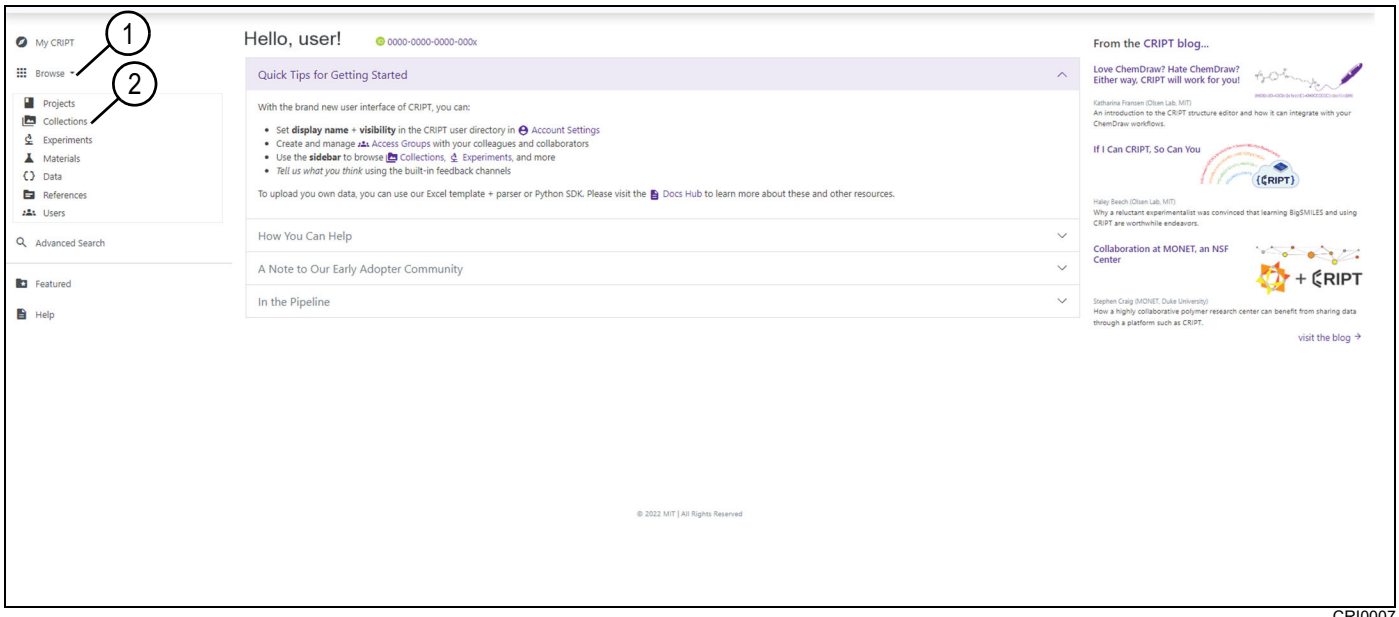


CRI0071

Figure 1-33

EDIT A COLLECTION

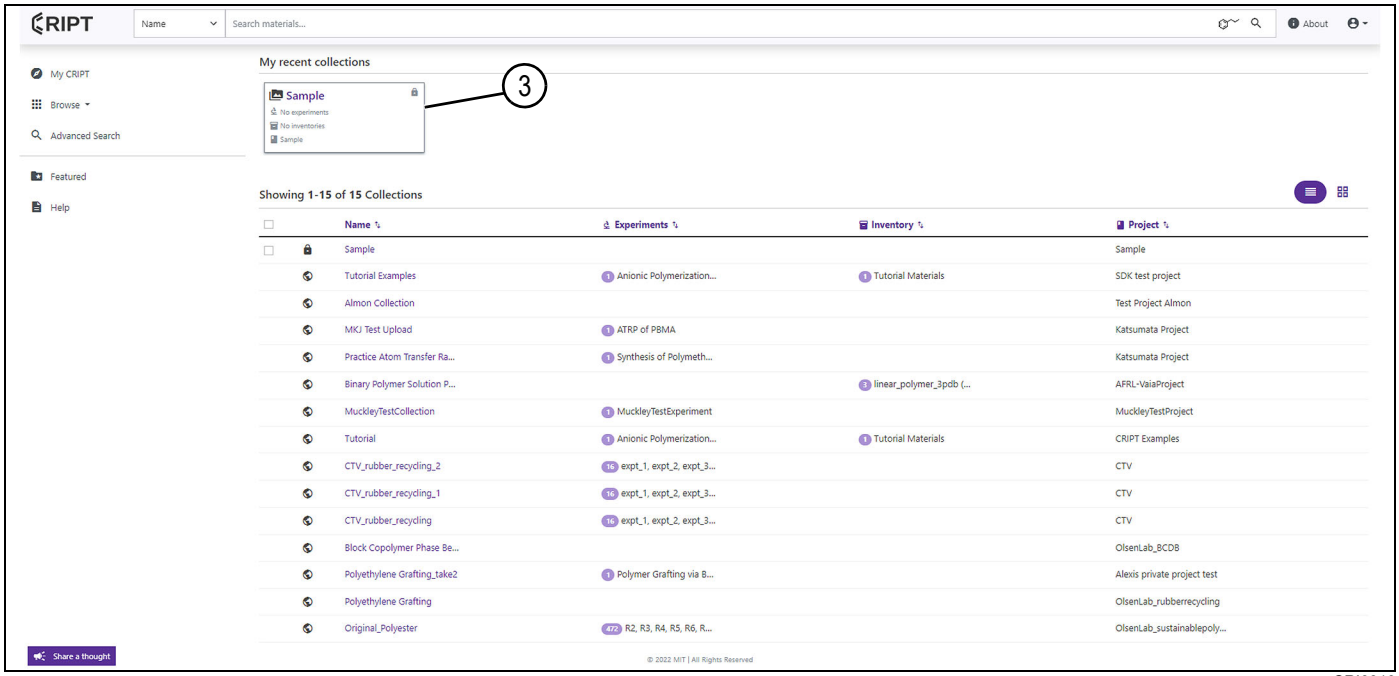
1. Select Browse drop-down (1) and select Collections (2).



CRI0007

Figure 1-34

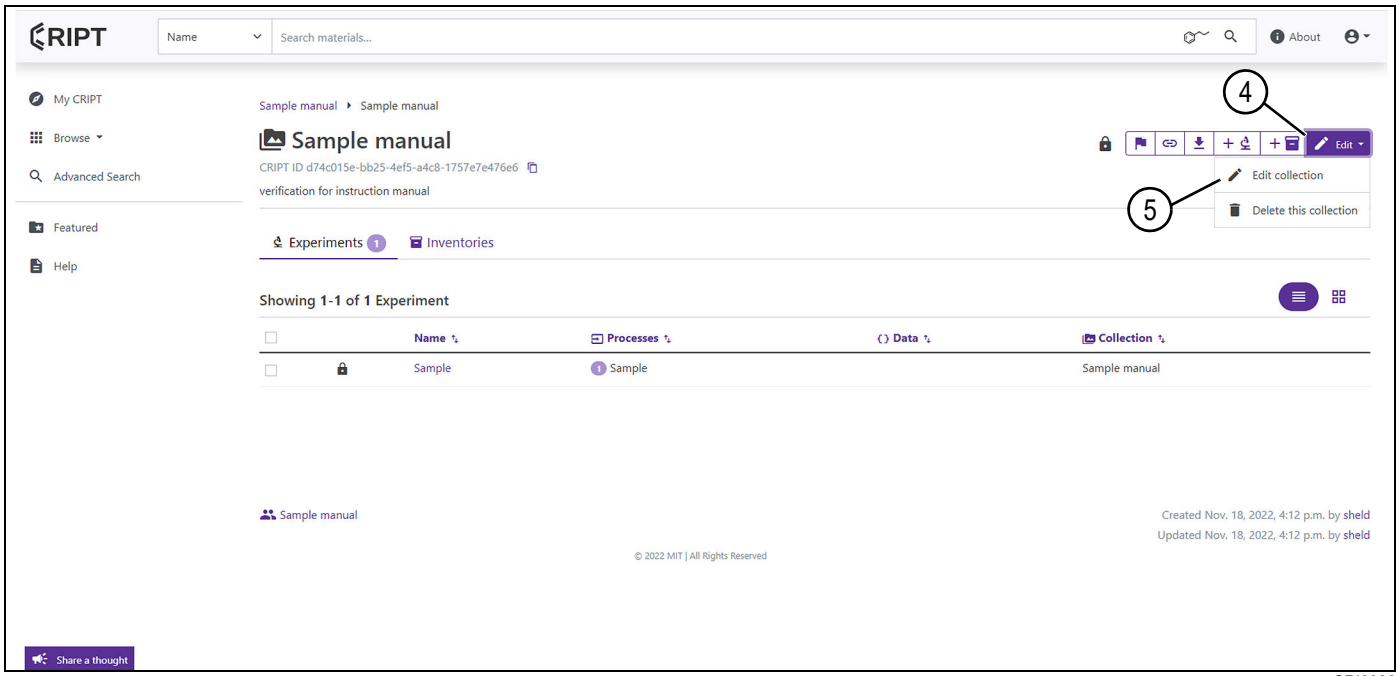
2. Select collection to be edited (3).



CRI0019

Figure 1-35

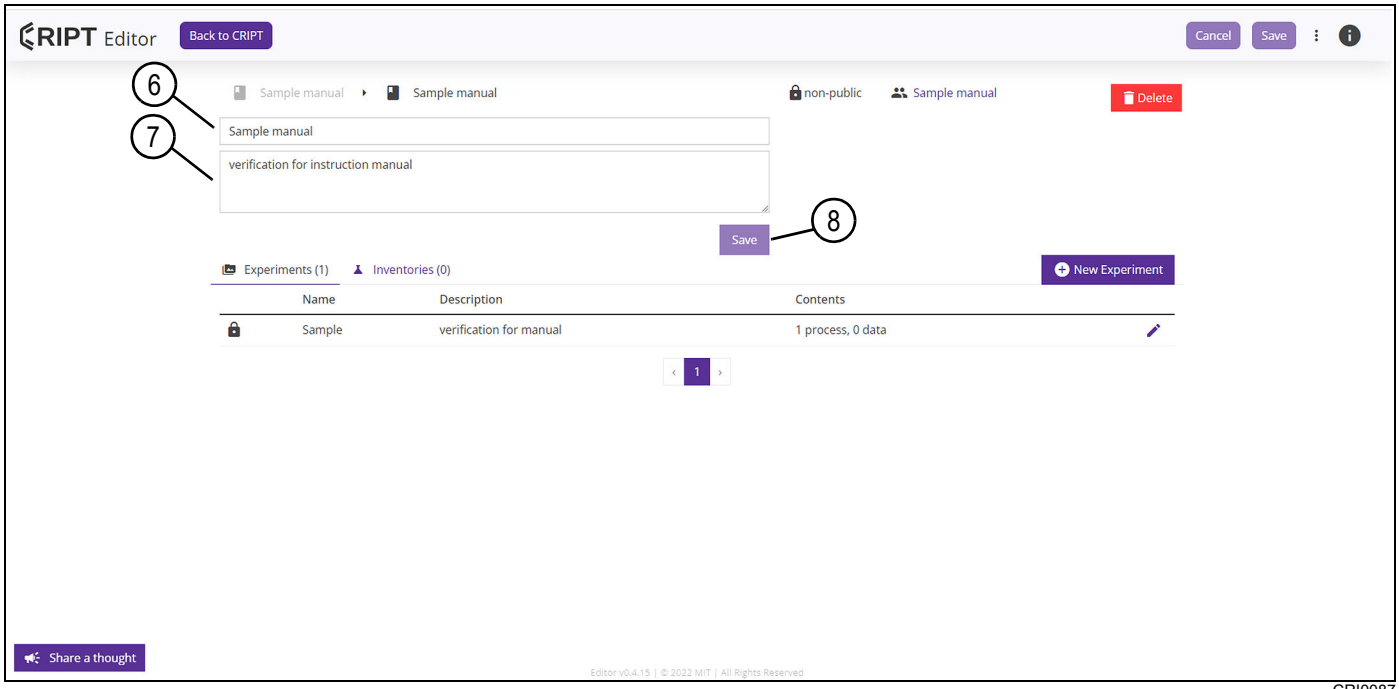
3. Select Edit (4) and select Edit collection (5).



CRI0086

Figure 1-36

4. Users can edit the collection name (6) and collection description (7).
5. Select Save (8) to save changes.

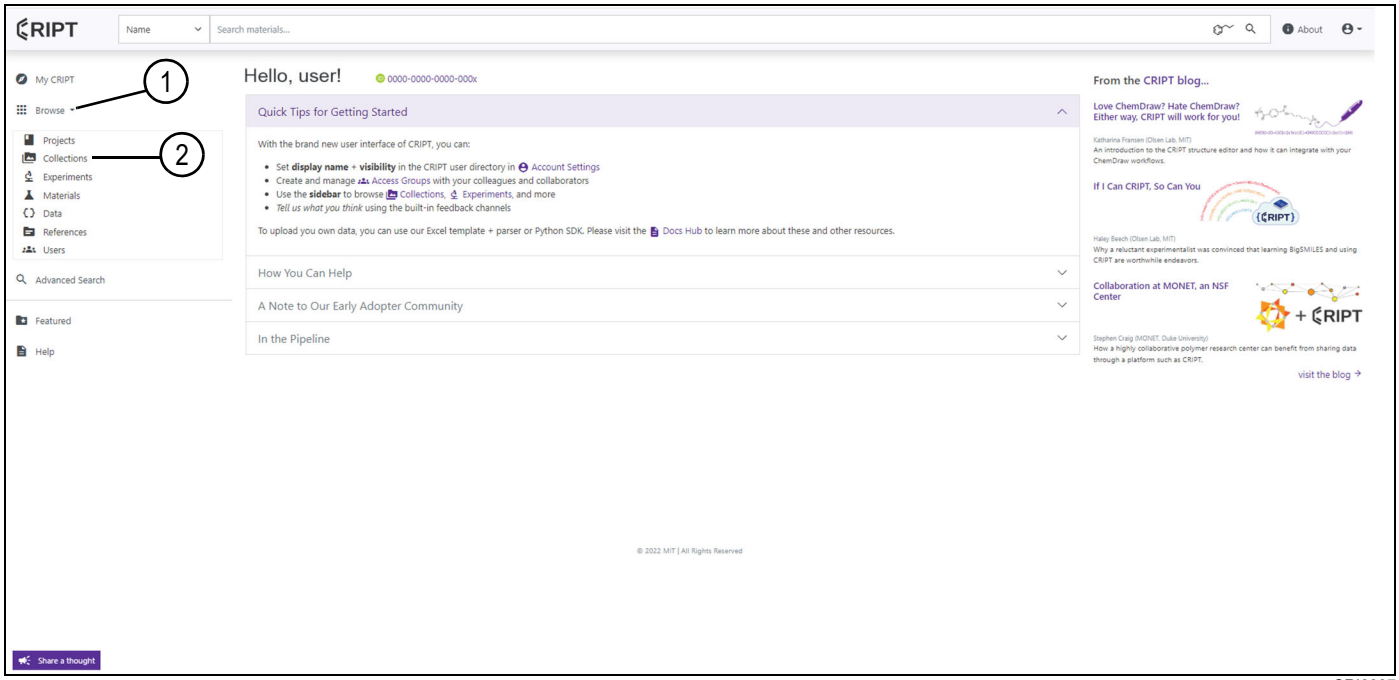


CRI0087

Figure 1-37

DELETE A COLLECTION

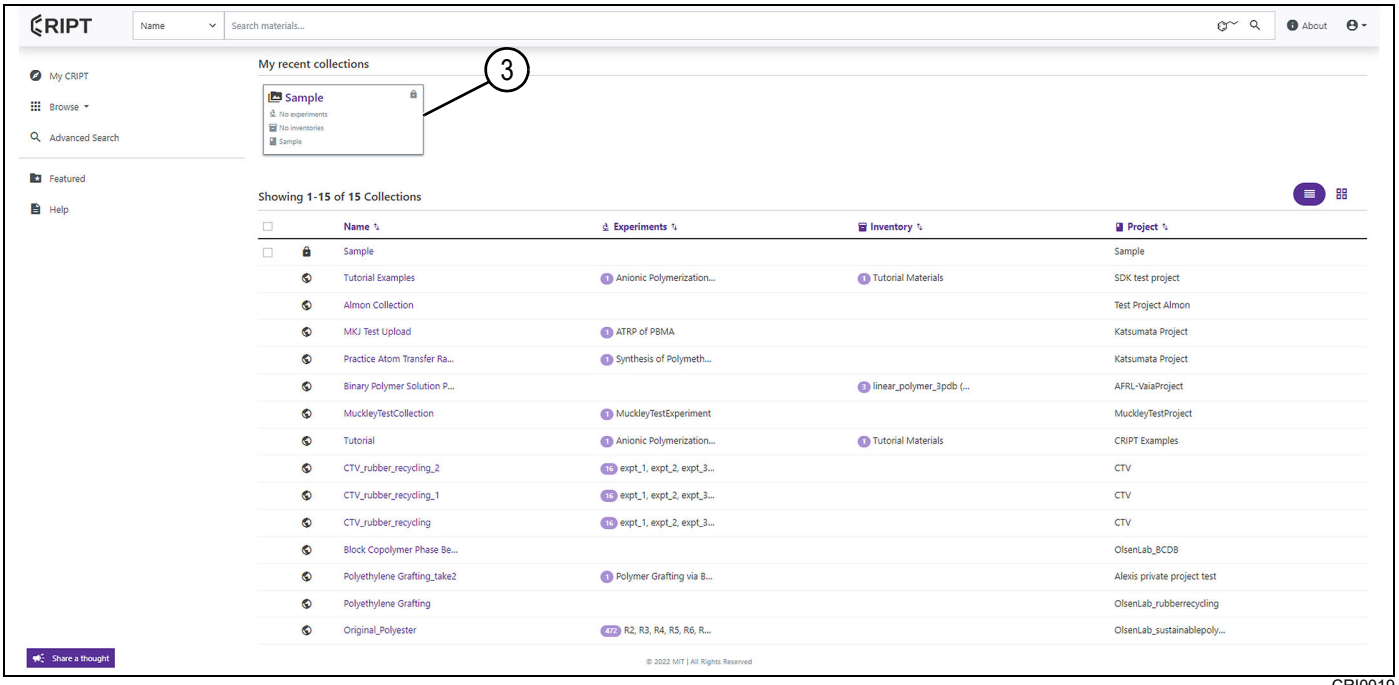
1. Select Browse drop-down and select Collections.



CRI0007

Figure 1-38

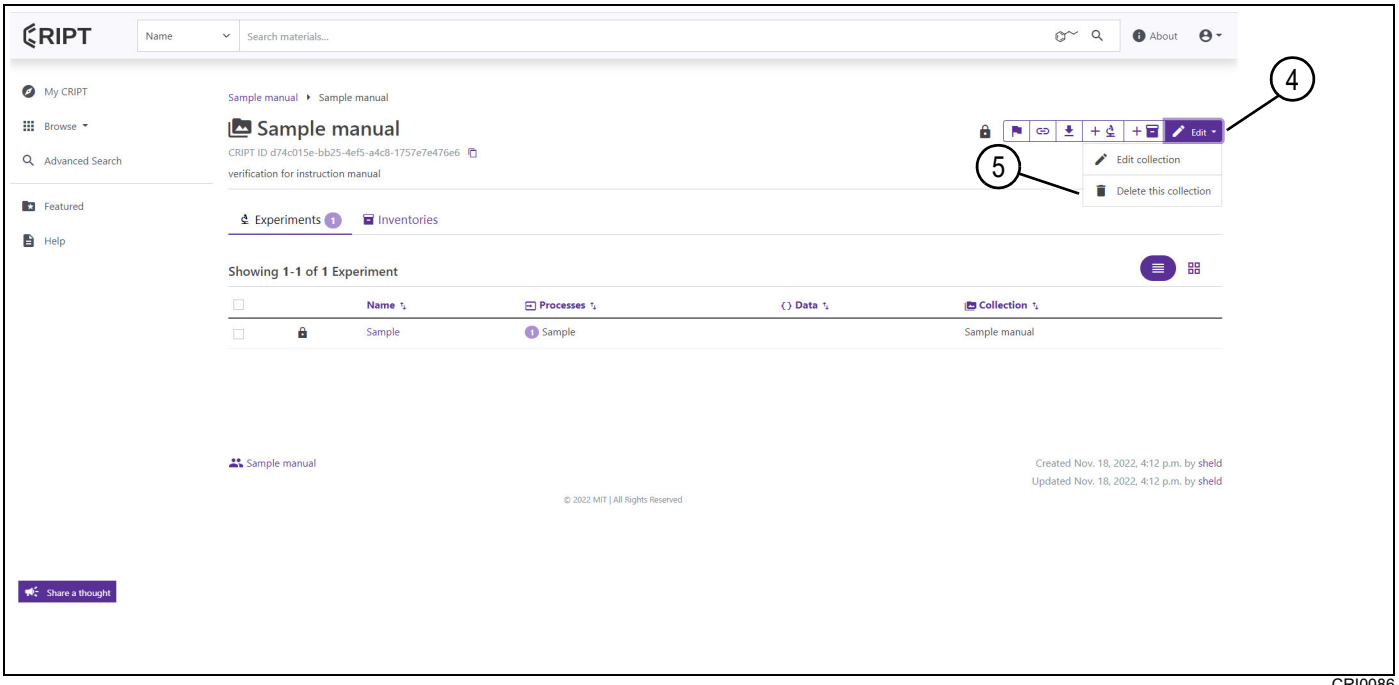
2. Select Collection to be deleted (3).



CRI0019

Figure 1-39

3. Select Edit (4).
4. Select Delete this collection (5).

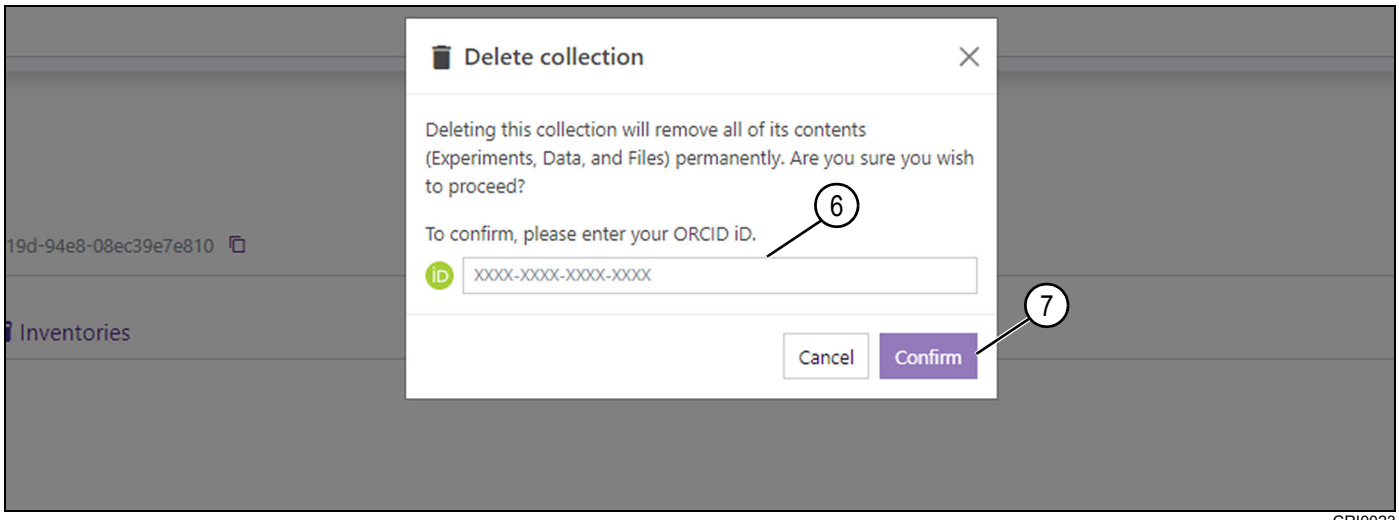


CRI0086

Figure 1-40

5. Enter ORCID iD (6) and select Confirm (7). For more information, see “Locate ORCID iD” on page 1-19.

NOTE: *If the ORCID iD contains any letters, they must be in lowercase.*

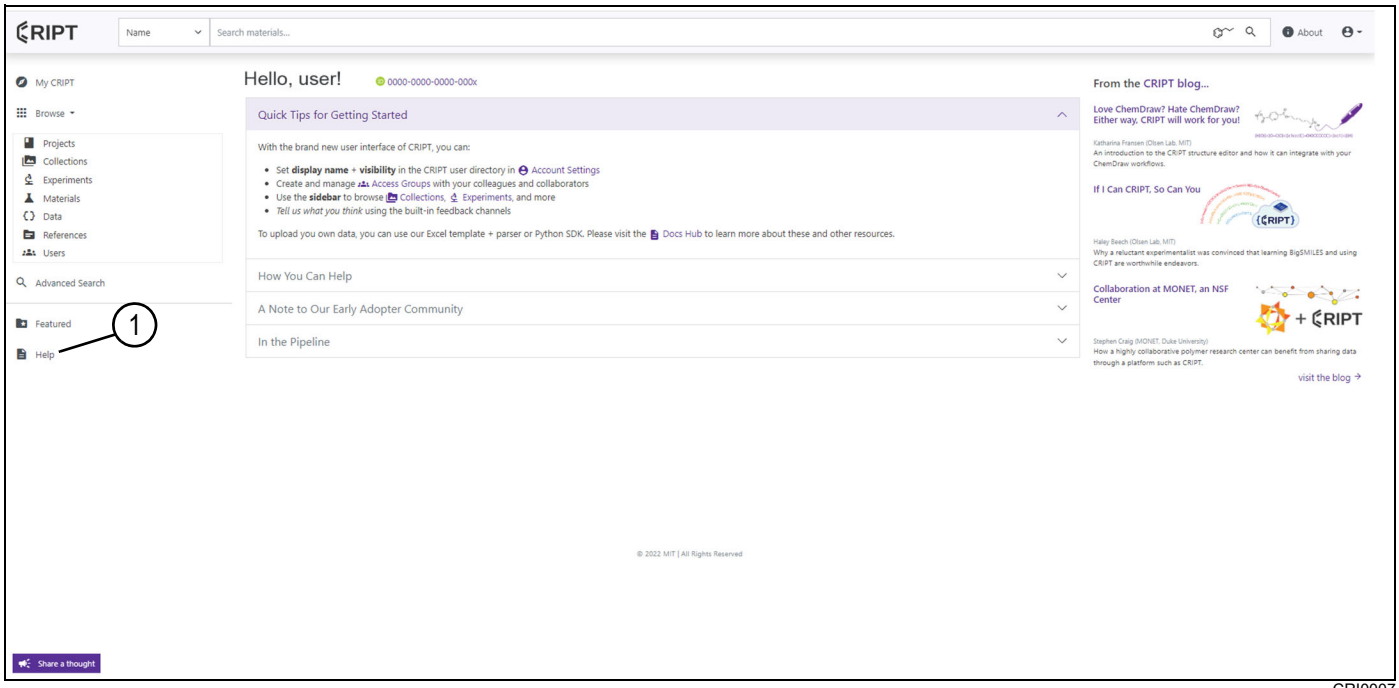


CRI0023

Figure 1-41

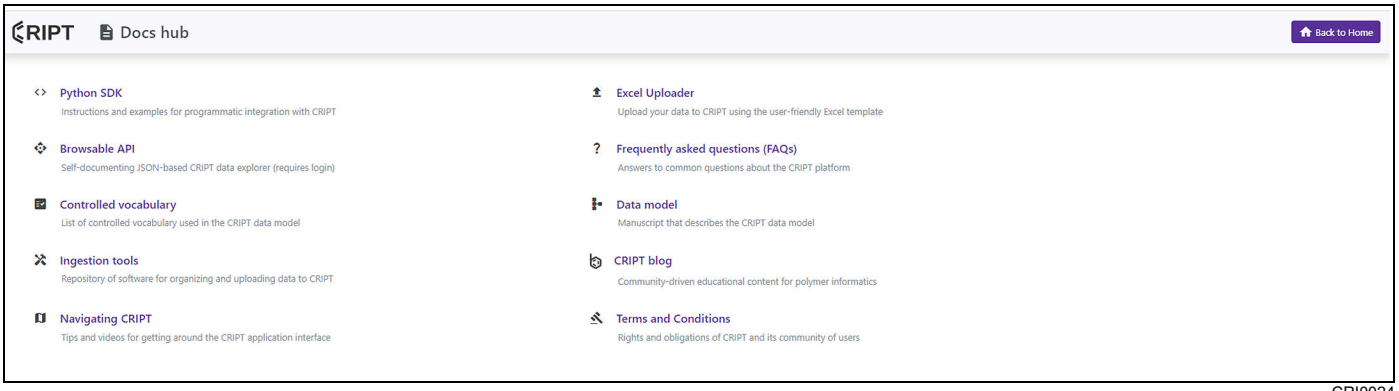
HELP MENU

1. Help menu (1) provides many different resources to assist users in getting the most out of the CRIPT platform. Most features of the help menu can be accessed from the CRIPT Home Page or the CRIPT Landing Page.



CRI0007

Figure 1-42



CRI0024

Figure 1-43

Python SDK

Instructions and examples for programmatic integration with CRIPT via the CRIPT python package cript.py. For more information, see *Python SDK*.

Browsable API

Self-documenting JSON-based CRIPT data explorer (requires login).

Controlled vocabulary

List of controlled vocabulary used in the CRIPT data model.

Ingestion tools

Repository of software for organizing and uploading data to CRIPT.

CRIPT Manual

Tips and videos for getting around the CRIPT application interface.

Excel Uploader

Upload user data to CRIPT in tabular form using the Excel template. For more information, see “Excel Uploader” on page 2-2.

Frequently asked questions (FAQs)

Answers to common questions about the CRIPT platform.

Data model

Manuscript that describes the CRIPT data model.

CRIPT blog

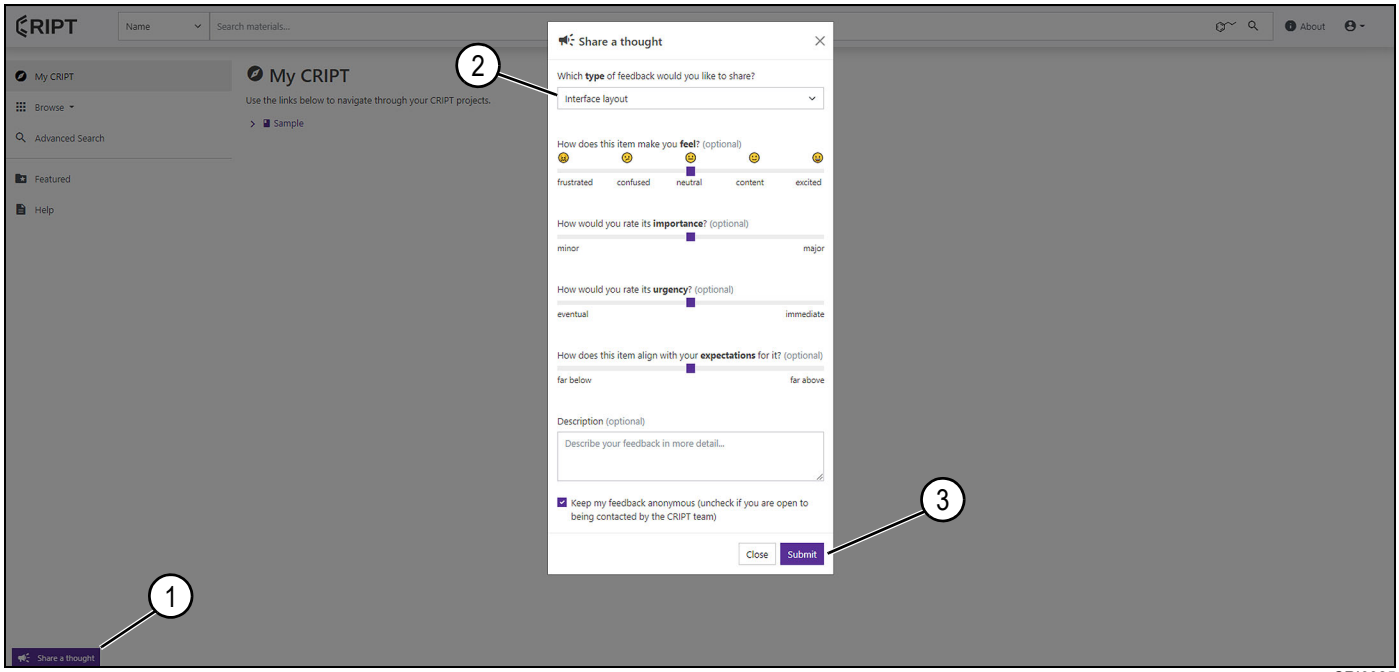
Community-driven educational content for polymer informatics.

Terms and Conditions

Rights and obligations of CRIPT and its community of users.

SHARE A THOUGHT

1. The Share a thought button (1) allows the user to provide user feedback on the CRIPT platform. The feedback collected is used to refine the CRIPT platform experience to better serve the users. Any feedback is greatly appreciated by the CRIPT platform administrators and programmers. The Share a thought button can be found in the lower left of most pages while navigating through the CRIPT web app.
2. Select the type of feedback from the drop-down (2).
3. Optional: you can select how you feel, rate the importance, and urgency.
4. Optional: you can describe your feedback in more details.
5. Optional: you can choose to submit your feedback anonymously, or uncheck it if you wish to be contacted and discuss your suggestions with the CRIPT team.
6. Select Submit (3) to send feedback.



CRI0025

Figure 1-44

Page Intentionally Blank

Chapter 2

Excel Uploader

USING EXCEL UPLOADER

REQUIRED DOWNLOADS

Uploading data to CRIPT using the Excel Uploader requires the following:

- Download the Excel Template.
- Download the CRIPT Excel Uploader executable.
- Locate and record the user specific API token.

EXCEL TEMPLATE

1. Navigate to CRIPT Home > Help > Excel Uploader > CRIPT Excel Template, or use the following hyperlink "CRIPT Excel Template".
2. Download file and save to a known working location.
3. For information on using the template, see “Excel Template” on page 2-12.

CRIPT EXCEL UPLOADER EXECUTABLE

1. Navigate to CRIPT Home > Help > Excel Uploader > CRIPT Excel Uploader executable or use the following hyperlink "CRIPT Excel Uploader executable".
2. Download file and save to a known working location.
3. For information on using the Excel Uploader, see “Excel Uploader” on page 2-2.

EXCEL UPLOADER

1. Locate the saved CRIPT Excel Uploader file and run the file.

NOTE: *All fields must be filled out before upload can begin.*

HOST

Host (1) indicates the CRIPT instance that you want to upload your data to, whether that is CRIPT or a private instance.

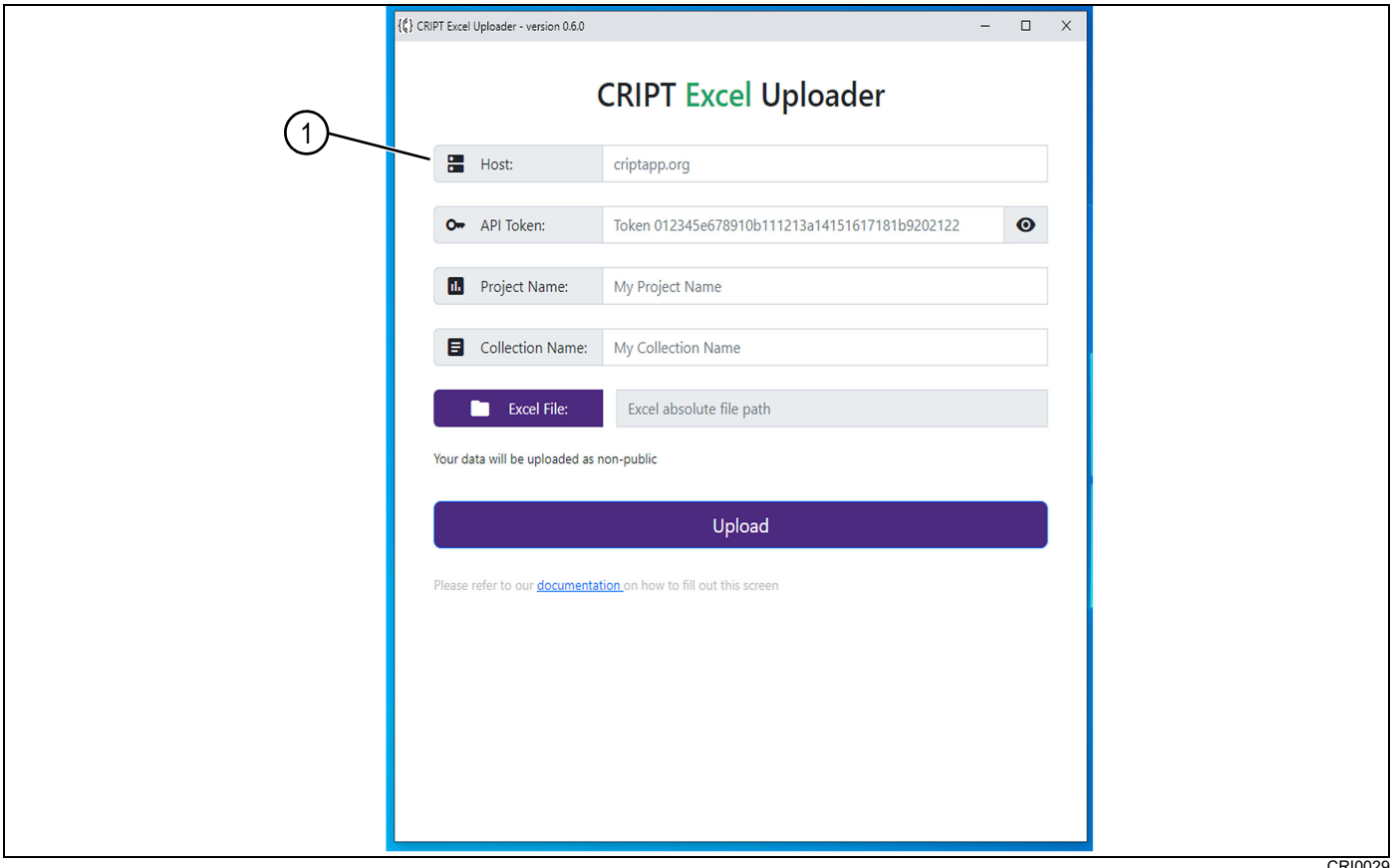
For most users, the Host will be **criptapp.org**.

NOTE: *This field will need to be entered.*

Example: Host: criptapp.org

However, if any user wants to connect to their own private instance of CRIPT, they can easily do that by changing the host to whatever URL they are using.

Example: Host: myPrivateWebsite.com



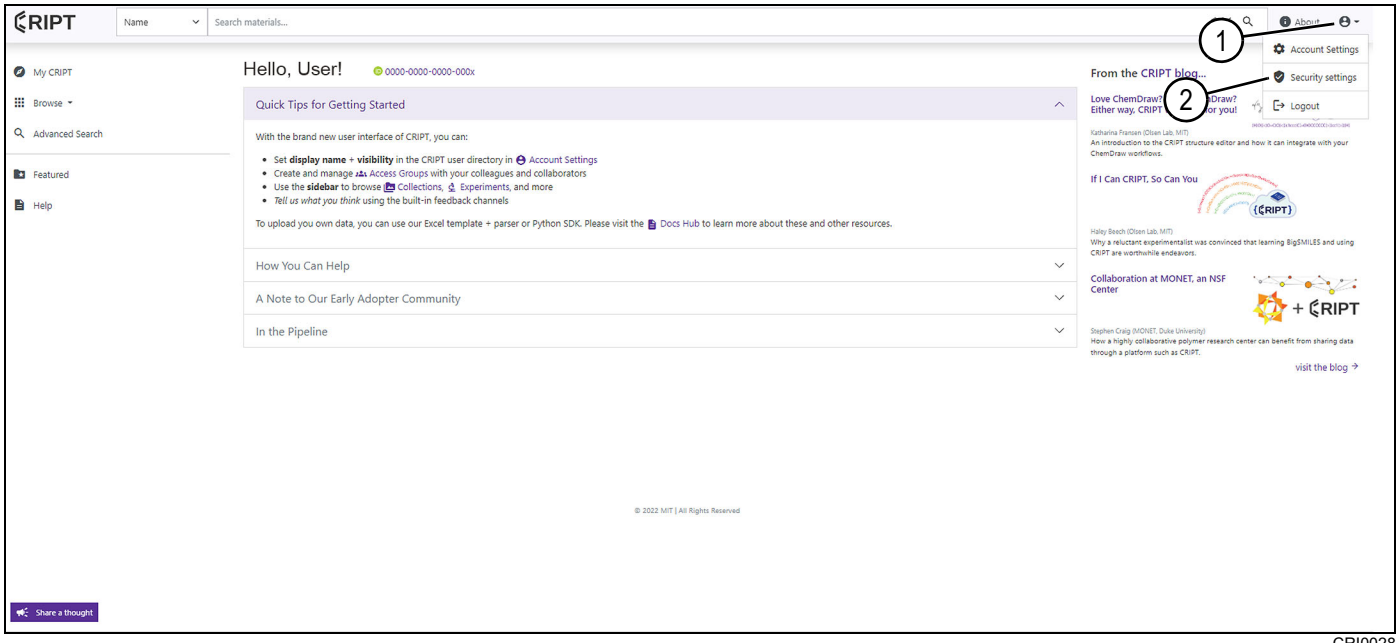
CRI0029

Figure 2-1

API TOKEN

The API Token is needed to authenticate the user before saving data.

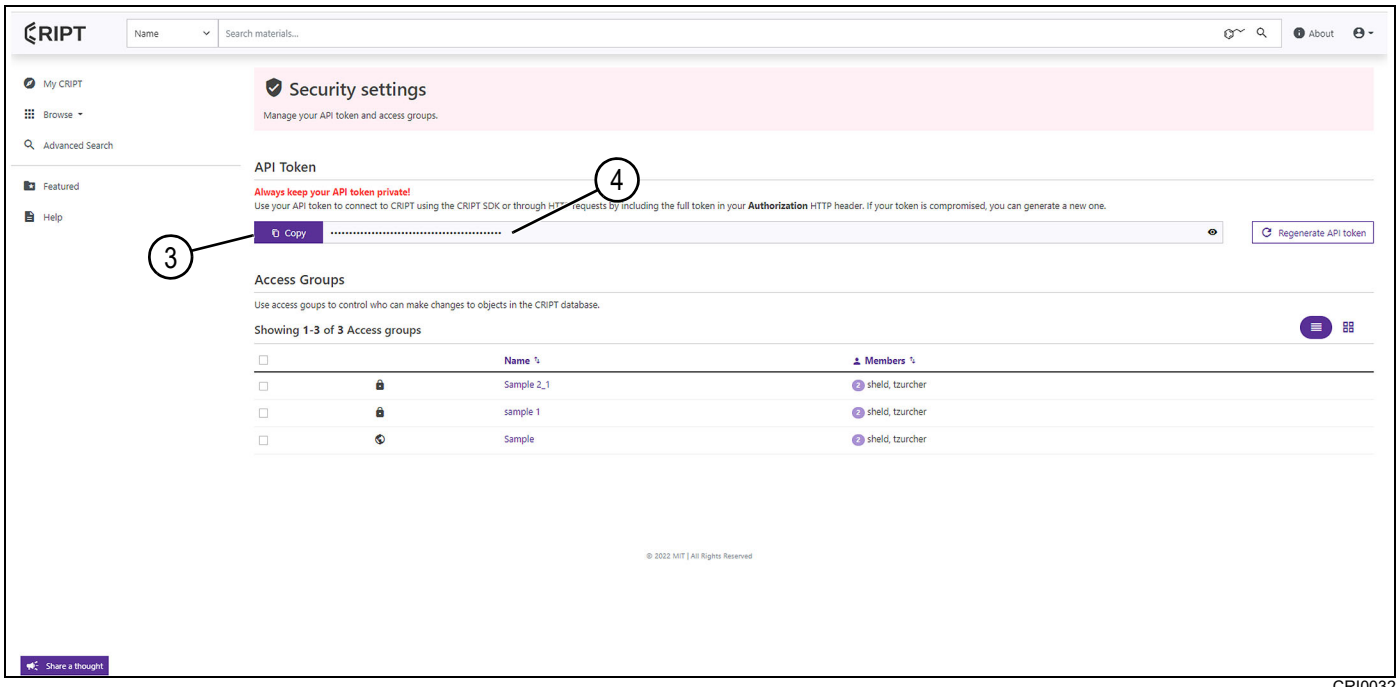
1. On the CRIPT home page, select the user icon (1) and Security settings (2).



CRI0028

Figure 2-2

2. Select the Copy button (3) next to the API Token (4) to copy it to the clipboard.



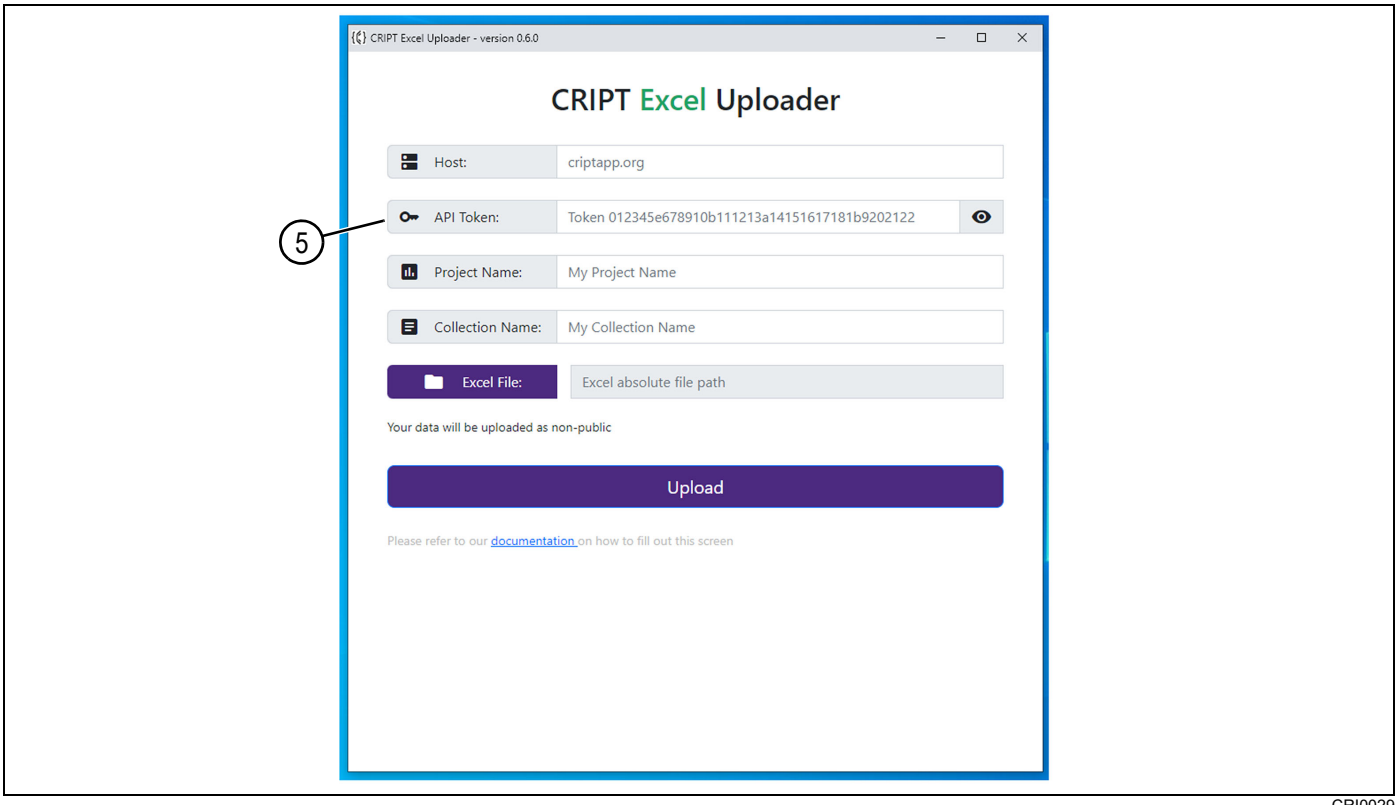
CRI0032

Figure 2-3

3. Paste API Token into the API Token field (5).

NOTE: *The word "Token" in front of the random characters is part of the token as well.*

Example: API Token: Token 4abc478b25e30766652f76103b978349c4c4b214



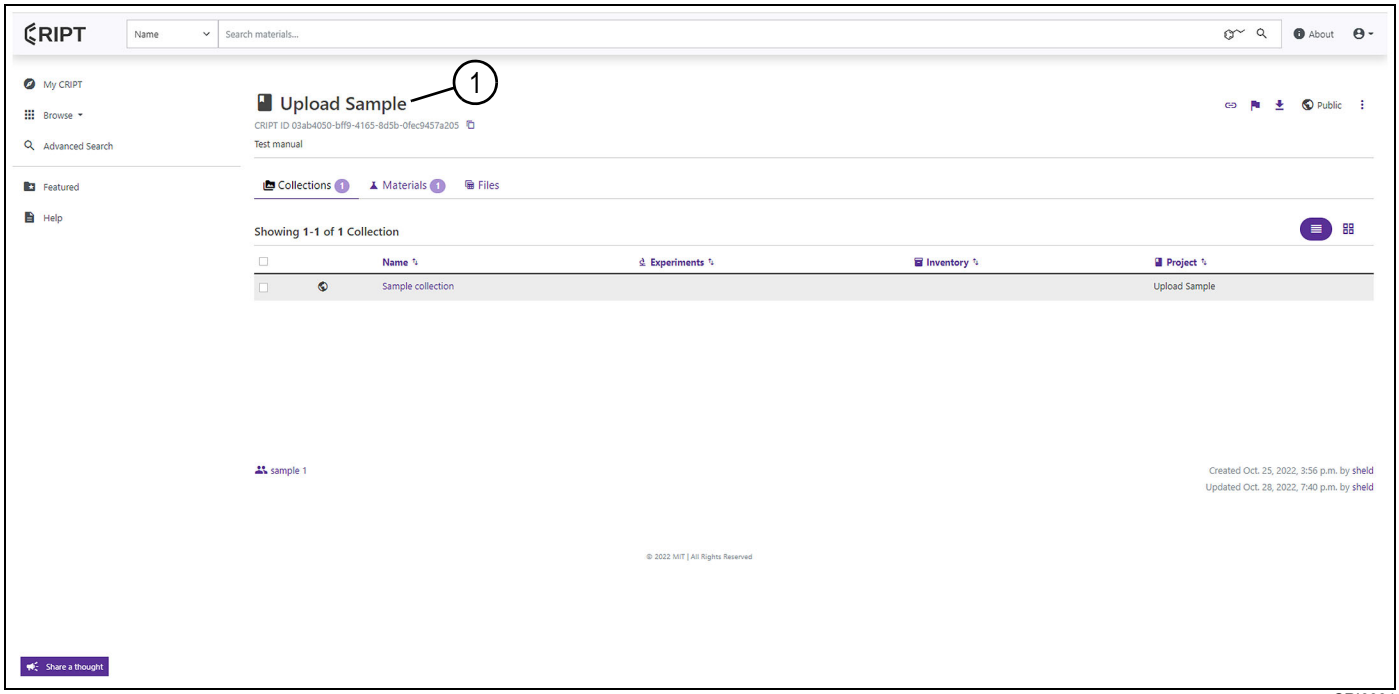
CRI0029

Figure 2-4

PROJECT

A Project can be thought of as a bunch of folders with each containing experiments that contribute to a single project. A Project is needed for the software to know which project these experiments are for, and each Collection belongs to a Project. For more information, see “Create a Project” on page 1-10.

1. Navigate to the Projects tab within CRIPT.
2. Locate and select the project associated with the data to be uploaded.
3. Copy the Project name (1).

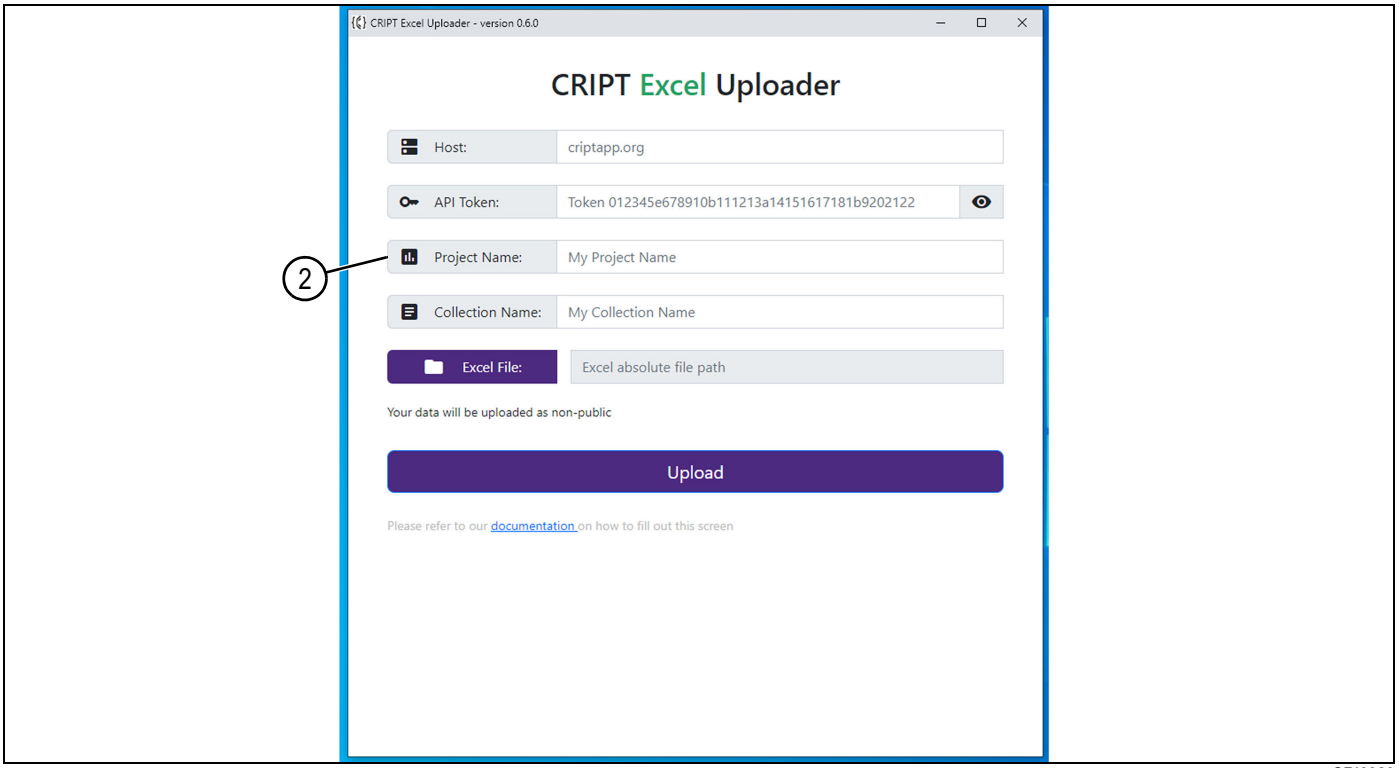


CRI0031

Figure 2-5

4. Paste Project name into the Project Name field (2).

Example: Project Name: Upload Sample



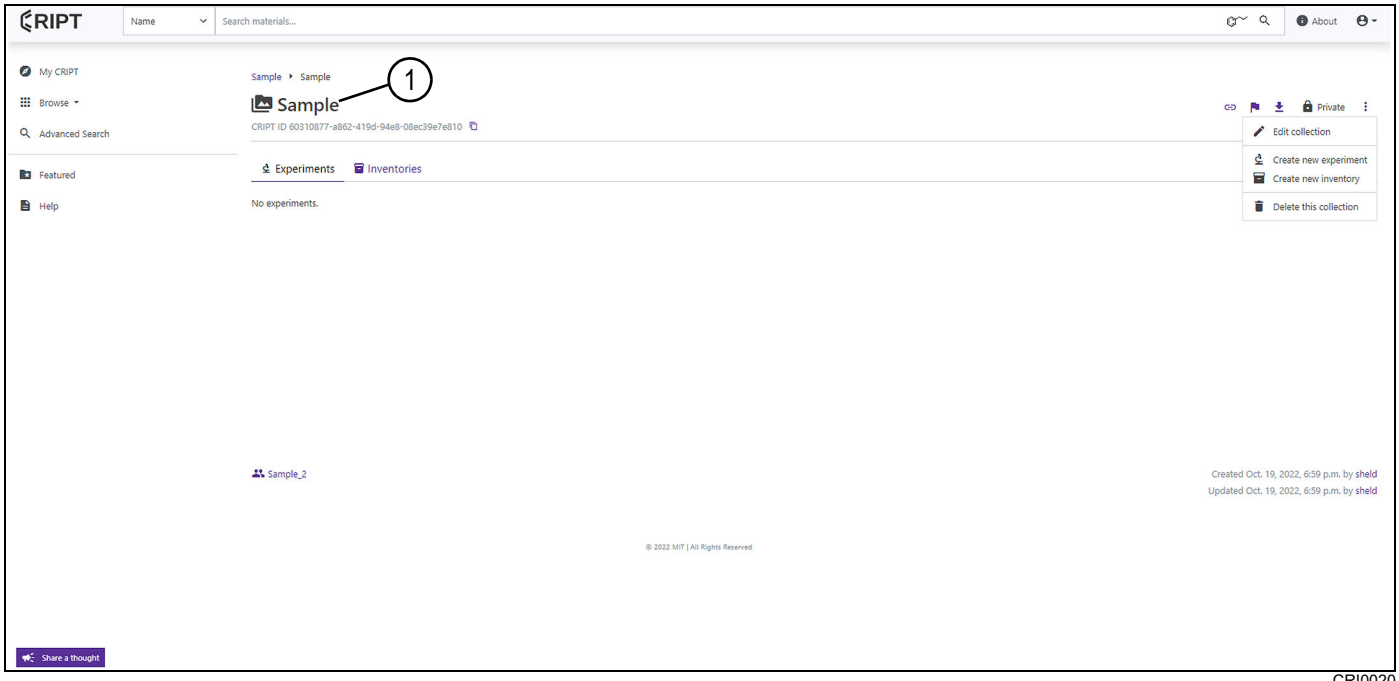
CRI0029

Figure 2-6

COLLECTION

A Collection can be thought of as a binder filled with experiments. The entire Excel file will become a collection within the CRIPT Platform. For more information, see “Create a Collection” on page 1-20.

1. Navigate to the Collections tab within CRIPT.
2. Locate and select the collection associated with the data to be uploaded.
3. Copy the Collection name (1).

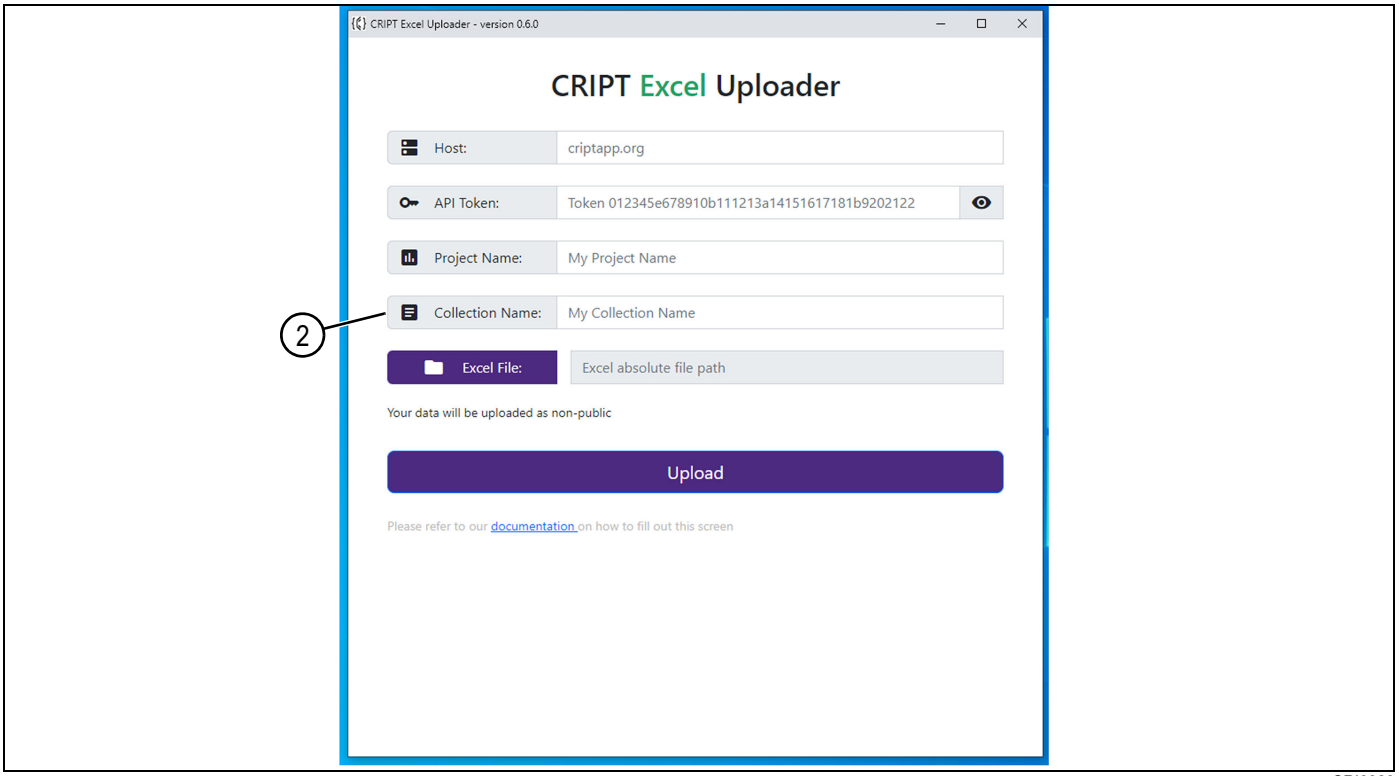


CRI0020

Figure 2-7

4. Paste Collection name into the Collection Name field (2).

Example: Collection Name: Sample



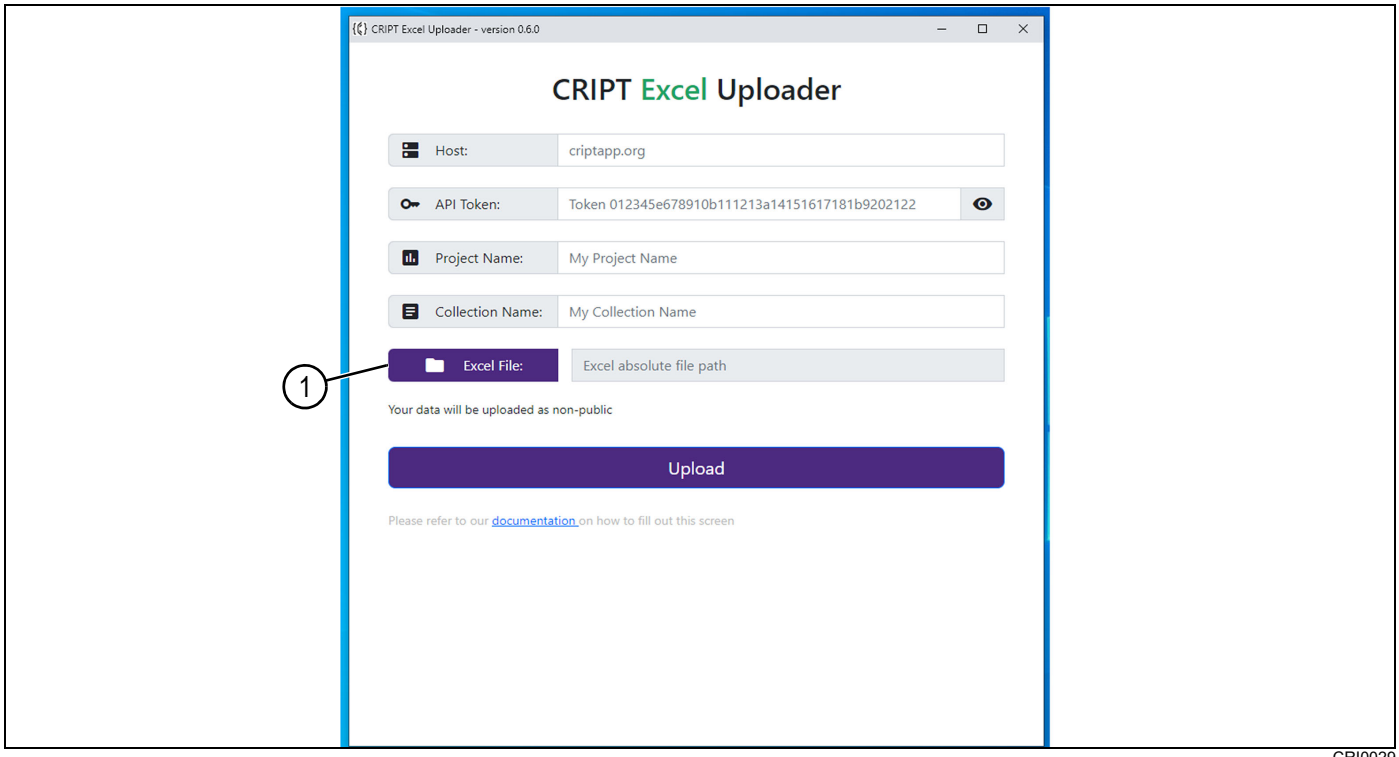
CRI0029

Figure 2-8

EXCEL FILE

This field allows the user to search for the file to upload.

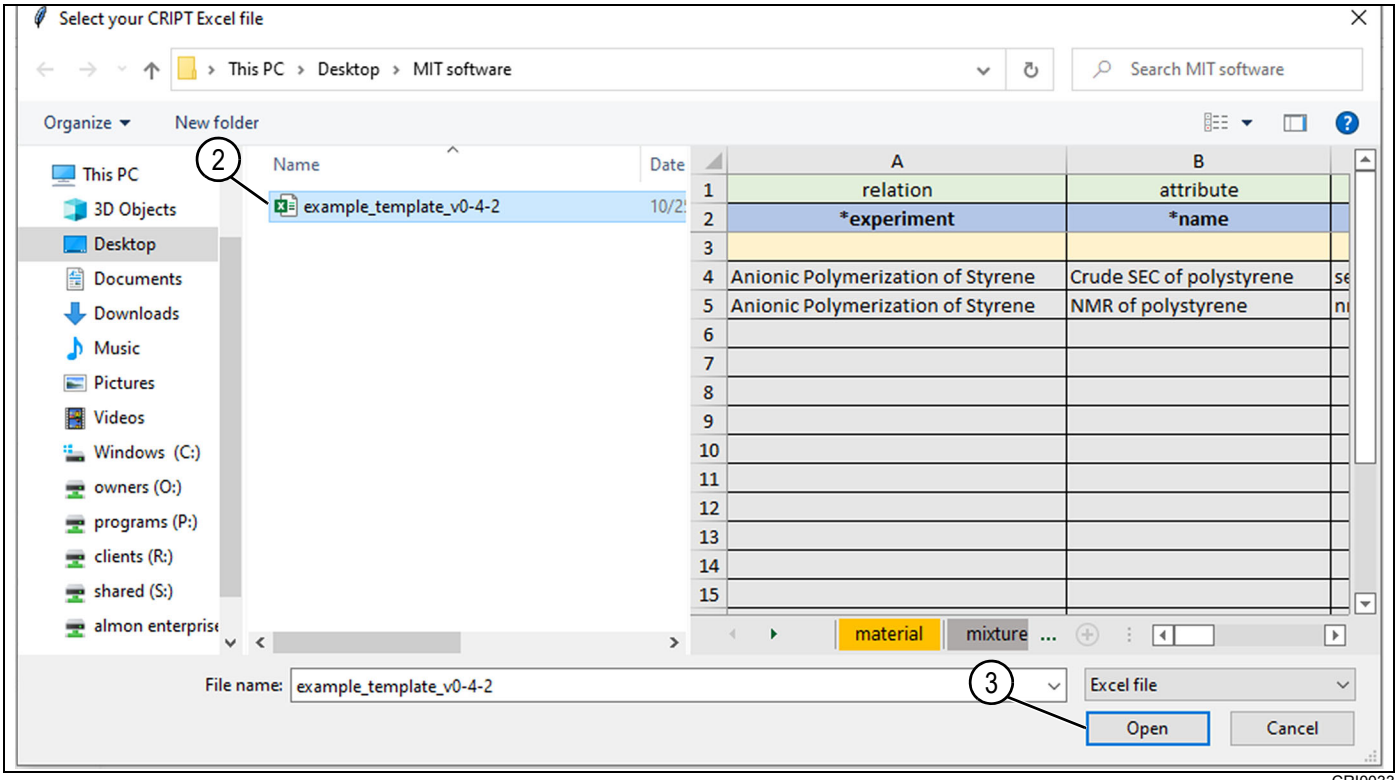
1. Select Excel File (1).



CRI0029

Figure 2-9

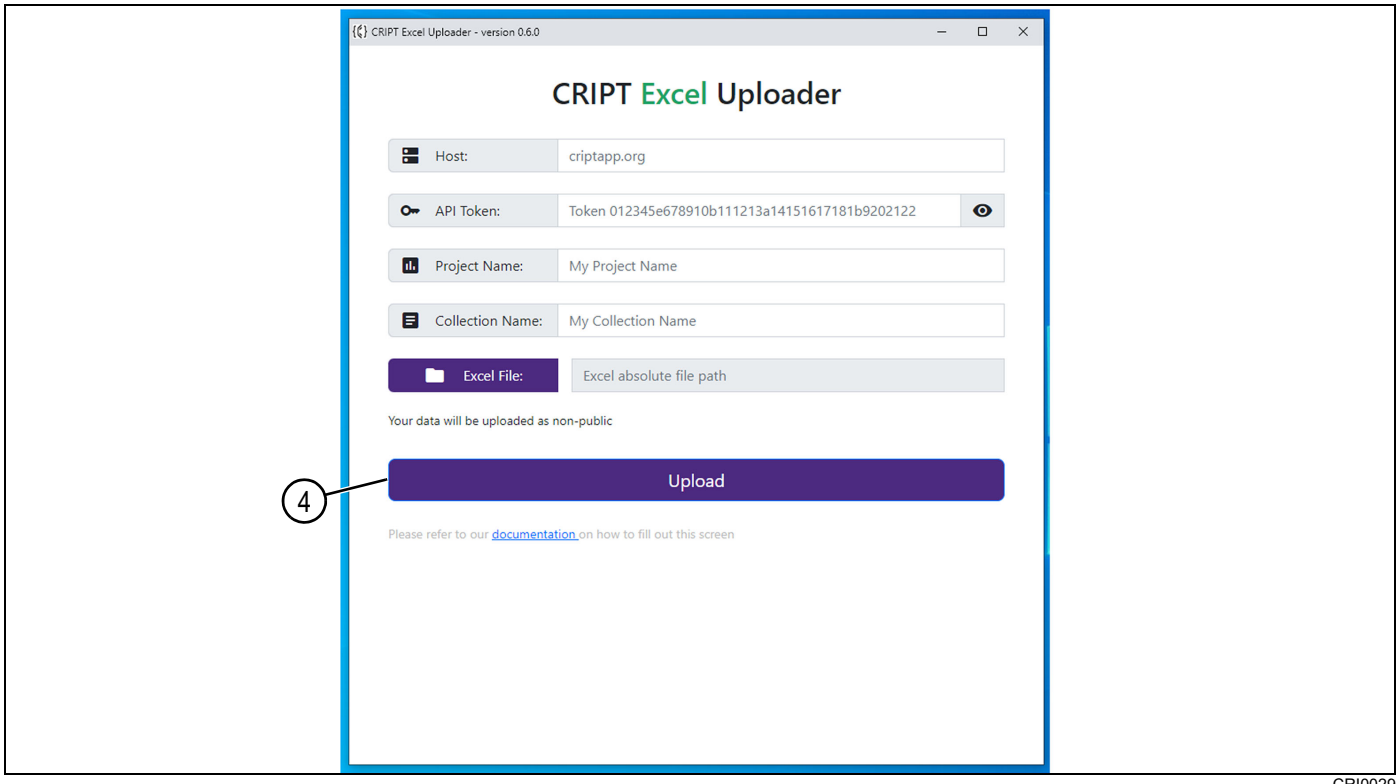
2. Locate file (2) to be uploaded and select.
3. Select Open (3) and the Excel File location will auto populate the field.



CRI0033

Figure 2-10

4. Select the Upload button (4).



CRI0029

Figure 2-11

EXCEL TEMPLATE

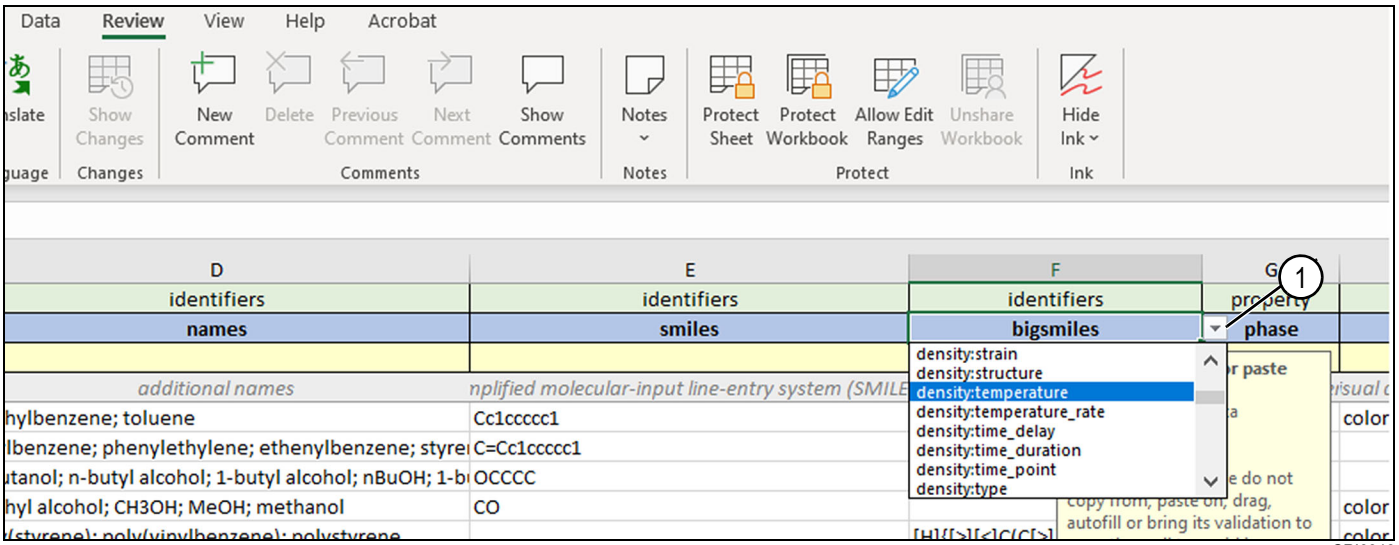
COLUMN SETUP

There are two different methods for creating the column structure for data entry.

AUTOFILL

1. Open Excel Template.
2. The Excel Template will have some of the columns started for reference points.
3. Create new columns by using the drop-downs (1).

NOTE: *Creating columns requires an understanding of the structure of the individual sheet, see* ***Chapter 2, "Structure of Excel Sheets"****.*



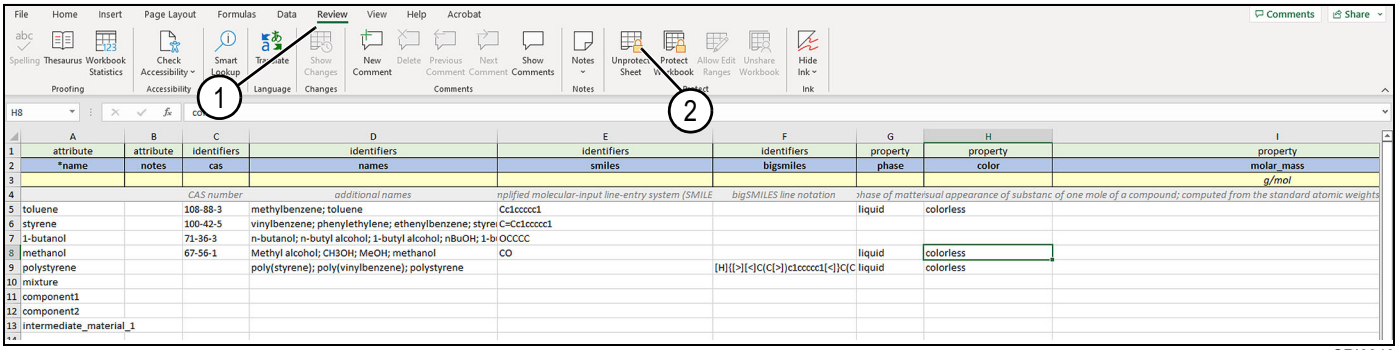
CRI0046

Figure 2-12

MANUAL ENTRY

1. Open Excel Template.
2. The Excel Template will have some of the columns started for reference points.
3. Unlock the Excel Template by selecting Review (1), and then Unprotect Sheet (2).

NOTE: *The sheets do not require a password and are locked to prevent accidentally deleting formulas within the individual cells.*

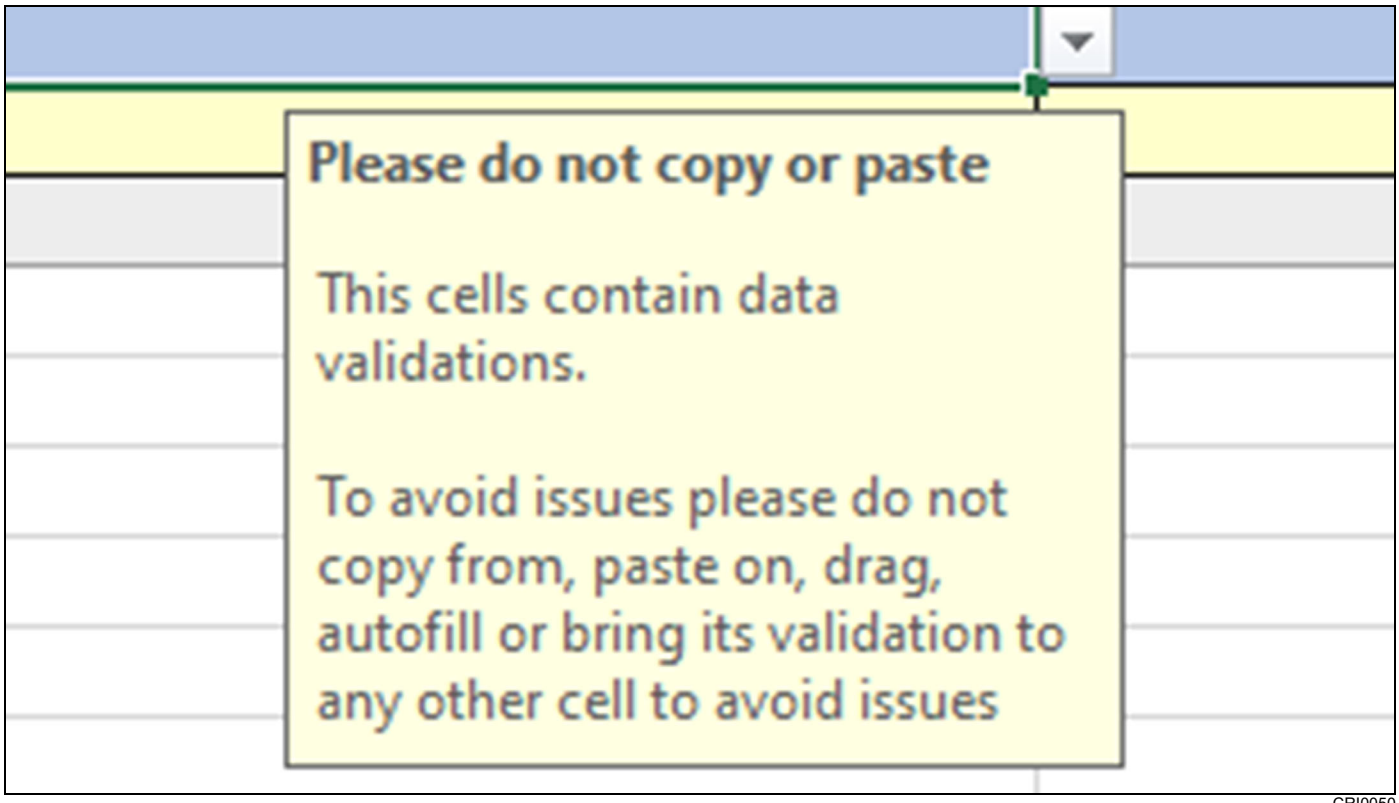


CRI0049

Figure 2-13

- To prevent Data validation problems, do not copy, cut, paste, or drag cells with drop-down data validations into any other cell.

- There are input messages on cells that contain drop-downs to alert the user not to accidentally change.

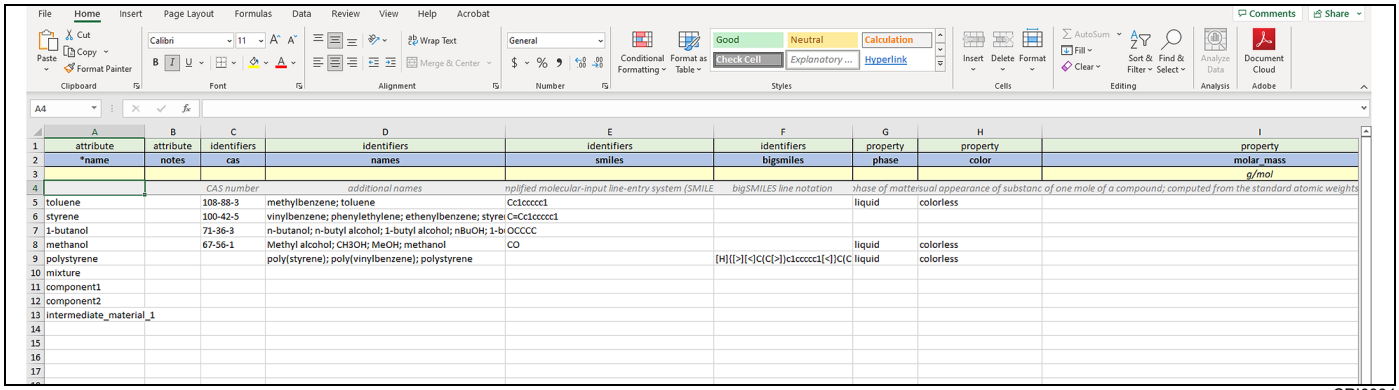


CRI0050

Figure 2-14

STRUCTURE OF EXCEL SHEETS

The first (3) rows define the type of data that will input into a column. Row 4 is used for user instructions and is not used by the program. Row 5 and beyond is used for data input.



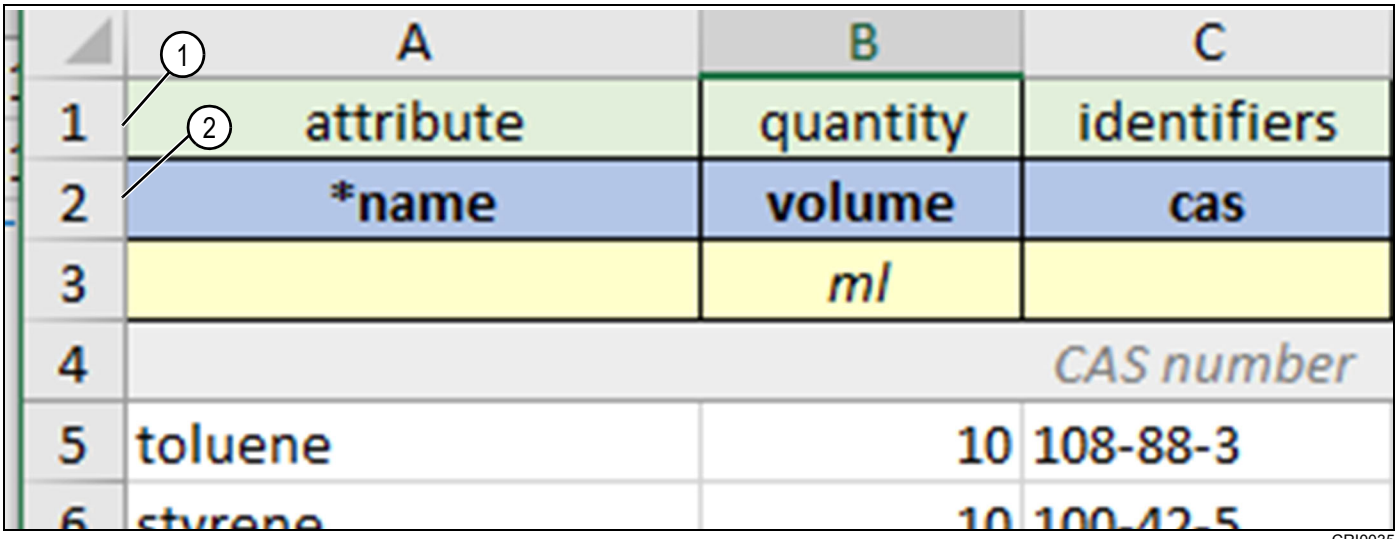
	A	B	C	D	E	F	G	H	I
1	attribute	attribute	identifiers	identifiers	identifiers	identifiers	property	property	property
2	*name	notes	cas	names	smiles	bigsmiles	phase	color	molar_mass
3									g/mol
4			CAS number	additional names	nplified molecular-input line-entry system (SMILE	bigSMILES line notation	hase of matterisual appearance of substanc	of one mole of a compound; computed from the standard atomic weights	
5	toluene		108-88-3	methylbenzene; toluene	Cc1ccccc1		liquid	colorless	
6	styrene		100-42-5	vinylbenzene; phenylethylene; ethenylbenzene; styre	C=Cc1ccccc1				
7	1-butanol		71-36-3	n-butanol; n-butyl alcohol; 1-butyl alcohol; nBuOH; 1-b	OCCCC				
8	methanol		67-56-1	Methyl alcohol; CH3OH; MeOH; methanol	CO		liquid	colorless	
9	polystyrene			poly(styrene); poly(vinylbenzene); polystyrene		{H]([>][<]C(C[>])c1ccccc1[<])C(C	liquid	colorless	
10	mixture								
11	component1								
12	component2								
13	intermediate_material_1								
14									
15									
16									
17									

CRI0034

Figure 2-15

ROW 1: CATEGORY

Row 1 (1) defines the characteristic of row 2 (2).



CRI0035

Figure 2-16

All possible options are:

- **attribute**
  - Defines the nomenclature of the column.
- **condition**
  - The condition under which the property was found.
  - Some examples include: *temperature, mixing_rate, stirring, and time_duration.*
  - A full list of conditions can be found at the CRIPT Docs hub. Navigate to CRIPT Home > Help > Controlled vocabulary > Condition key, or use the following hyperlink "**Condition controlled vocabulary**".
- **identifier**
  - Provides the naming method for material in the column.
  - Some examples include: *smiles, bigSmiles, cas, and name.*
  - A full list of material identifiers can found at the CRIPT Docs hub. Navigate to CRIPT Home > Help > Controlled vocabulary > Material identifier key, or use the following hyperlink "**Material identifier controlled vocabulary**".
- **property**
  - Describes the characteristic or unit combinations for the property in the column.
  - Some examples include: *phase, color, density, and molar_mass.*

- **relation**
  - Allows column row to reference another row in a different sheet.
  - An example would be to relate attribute (3) *name (4) to relation (5) *mixture (6) on another sheet. For more information on sheets, “See “Individual Excel Sheets” on page 2-17”.

	A	B	C	D	E	F
1	attribute	attribute	identifiers	identifiers	identifiers	identifiers
2	*name	notes	cas	names	smiles	bigsmiles
3						
4			CAS number	additional names	nplified molecular-input line-entry system (SMILE	bigSMILES line notation
5	toluene		108-88-3	methylbenzene; toluene	Cc1ccccc1	
6	styrene		100-42-5	vinylbenzene; phenylethylene; ethenylbenzene; styre	C=Cc1ccccc1	
7	1-butanol		71-36-3	n-butanol; n-butyl alcohol; 1-butyl alcohol; nBuOH; 1-b	OCCCC	
8	methanol		67-56-1	Methyl alcohol; CH3OH; MeOH; methanol	CO	
9	polystyrene			poly(styrene); poly(vinylbenzene); polystyrene		[H]{[>][<]C(C[>])c1ccccc1[<]}C(C
10	mixture					
11	component1					
12	component2					
13	intermediate_material_1					
14						
15						

CRI0044

Figure 2-17

	A	B
1	relation	relation
2	*mixture	*material
3		
4	Pick a value from *name column of material sheet that repersents the mixture	Pick a value from *name column of material sheet
5	mixture	component1
6	mixture	component2
7		
8		
9		
10		
11		
12		

CRI0045

Figure 2-18

- **quantity**
  - Column specifies the amount or combinations of quantities.
  - The numerical value can be combined with a unit of measure.

ROW 2: COLUMN NAME

Defines the name of the Column and can be edited by the user.

- Columns beginning with “ * “ are required (eg. *name).
- Columns beginning with “ # “ will be ignored (eg. #storage).
  - The # columns are a good idea to use if you want to have some notes but don't want them necessarily read or uploaded to CRIPT.
- Some sheets have drop-downs’ for row 2. After selecting an option, row 1, row 3, and row 4 are automatically populated with the correct information.

NOTE: *Some sheets have drop-downs for row 2. After selecting an option, row 1, row 3, and row 4 are automatically populated with the correct information.*

NOTE: *As the controlled vocabulary is updated, it is very possible that the drop-down options and autofill can become outdated.*

- Each Excel sheet has rows 1, 3, and 4 locked for protection.
  - This is because rows 1, 3, and 4 contain formulas that if accidentally deleted or overwritten would break the autofill feature.
  - The sheets are protected without a password and can be easily unprotected.

Nesting Headers in Row 2

Nesting can be defined as having a combination of names.

Combined names are separated by a colon " : ".

- Example: density at a certain temperature.
  - density:temperature
- Example: uncertainty of a material property condition.
  - density:temperature:uncertainty

NOTE: *Nesting "data" column values should derive from the " *name " column of the data sheet. For more information,* see "Individual Excel Sheets".

NOTE: *Nesting "citation", column values should derive from the " *name " column of the Citation sheet. For more information,*See "Individual Excel Sheets" on page 2-17.

ID

The Id feature is used to allow for multiple measurements through time. With Id it is possible to take several measurements through an experiment, and later use nesting to record more details.

If there are multiple densities throughout time and we want to show each of their temperatures (or any other condition), we can use an Id field to differentiate between the different temperatures. We denote an Id with brackets and a number inside such as [1] or [2]. The Id is used to identify distinct properties/conditions of the same type.

1. To identify two density measurements at two different temperatures, we could create the following column headers: [1]density (1), [1]density:temperature (2), [2]density (3), [2]density:temperature (4).

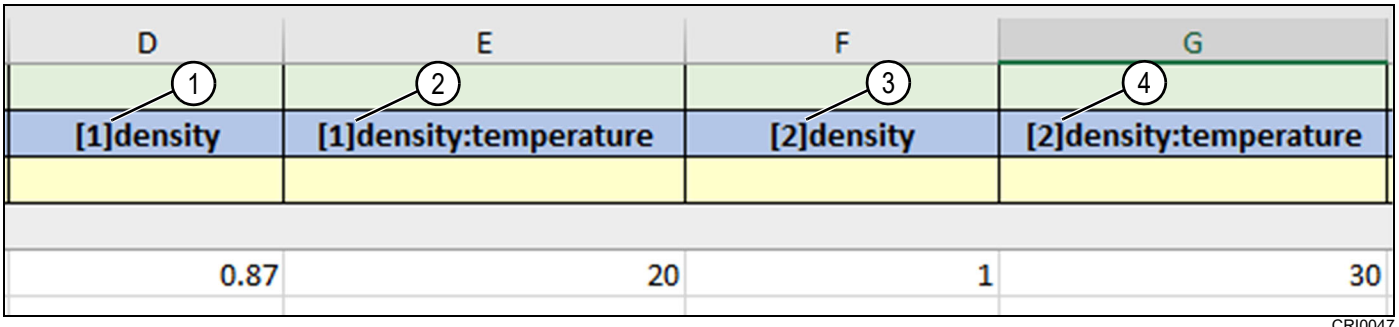


Figure 2-19

ROW 3: UNITS

Defines the units used in the column.

- Some examples include: *Celsius, g/ml.*
- A full list of supported units can be found at: **https://github.com/hgrecco/pint/blob/master/pint/default_en.txt**, or use the following hyperlink "supported units".

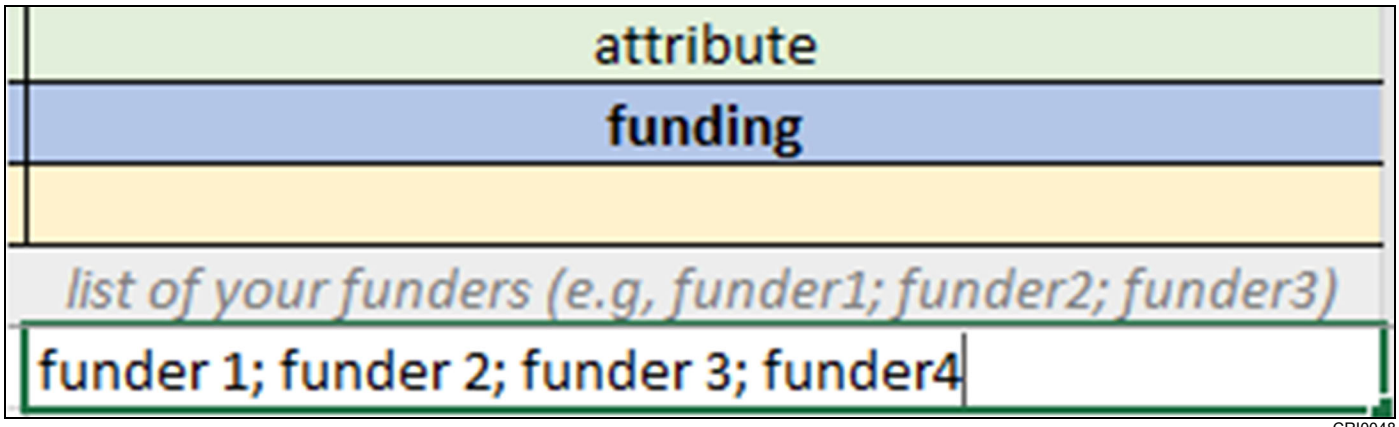
ROW 4: INSTRUCTIONS

Many columns contain instructions on row 4. These clarify the data that should go in that column after the value for row 2 is picked.

ROW 5 AND BEYOND: USER INPUT

These rows are used to enter user data.

- Some columns allow for more than a single value. List values must use a semicolon " ; " as a separator.
- Example of inputting multiple funders.



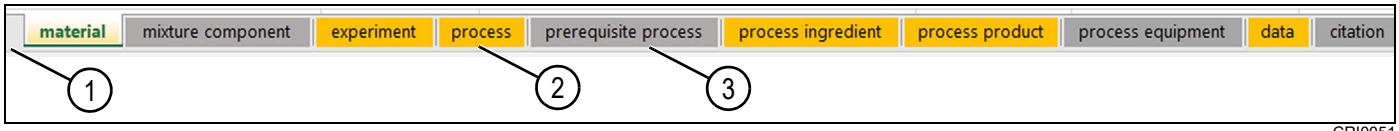
CRI0048

Figure 2-20

INDIVIDUAL EXCEL SHEETS

The Excel Template is comprised of 10 individual sheets in the sheet line (1) for specific characteristics. These are separated into two colored categories. Orange categories (2) are required and Grey categories (3) optional.

NOTE: *Sheets can not be renamed, but can be removed. The optional column's not needed can be left blank or deleted.*



CRI0051

Figure 2-21

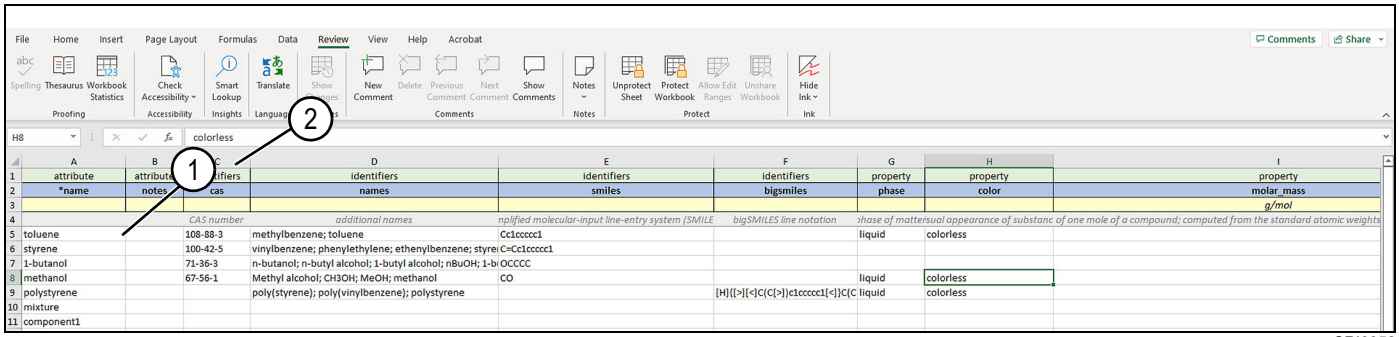
MATERIAL SHEET

The material sheet is used to input the materials at the beginning of the experiment (ingredients), and the material that occurs as a result of the experiment (process product).

1. Enter material's unique name in the first available row (1).

NOTE: *Each material entered must be unique and can not be duplicated.*

2. Generate new column category (2) as needed for additional information. See "Column setup" on page 2-12.



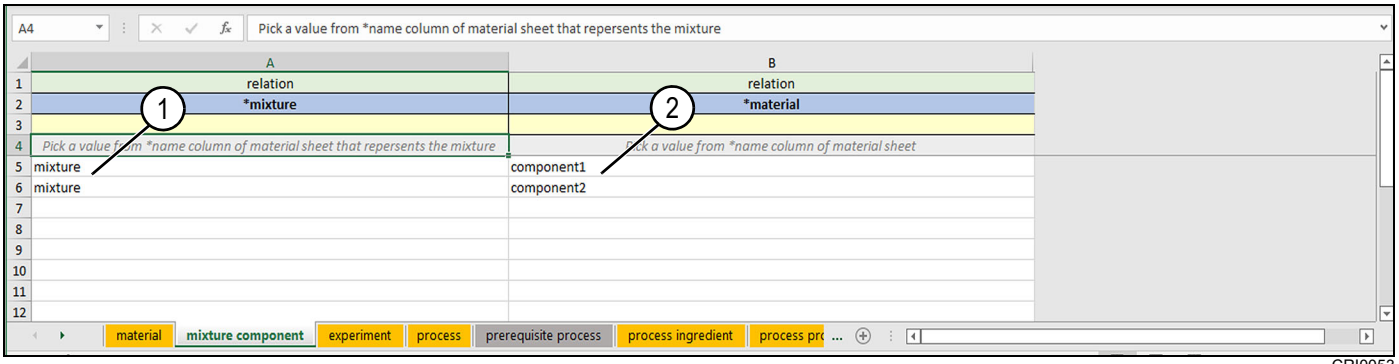
CRI0052

Figure 2-22

MIXTURE COMPONENT SHEET (OPTIONAL)

This sheet defines the components of the mixture materials.

1. The mixture value (1) is derived from the “ *name” column of the material sheet that represents the mixture.
2. The material value (2) is derived from the “ *name” column of the material sheet.



CRI0053

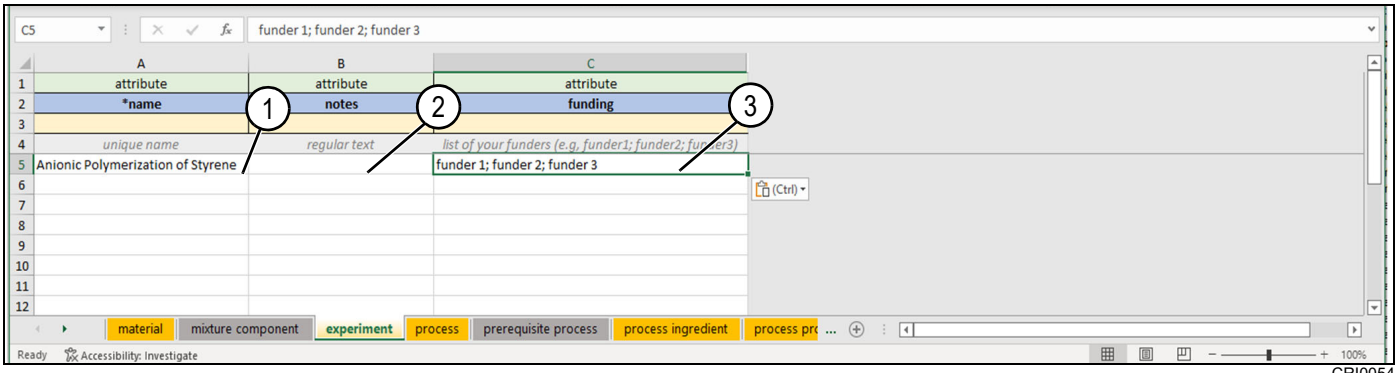
Figure 2-23

EXPERIMENT SHEET

This sheet defines the experiment.

1. Attribute name (1) should be a unique name to describe the experiment.
2. Notes (2) are used to add additional information regarding the experiment.
3. Funding (3) lists the funders for the project.

NOTE: When entering more then 1 funder, a semicolon “ ; “ must be used to separate the names.



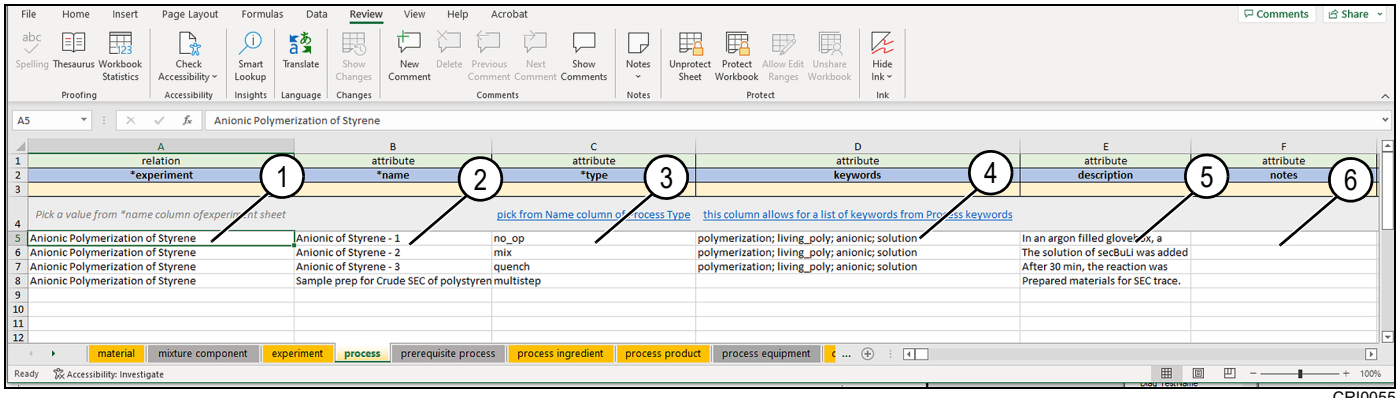
CRI0054

Figure 2-24

PROCESS SHEET

Define the processes of each experiment.

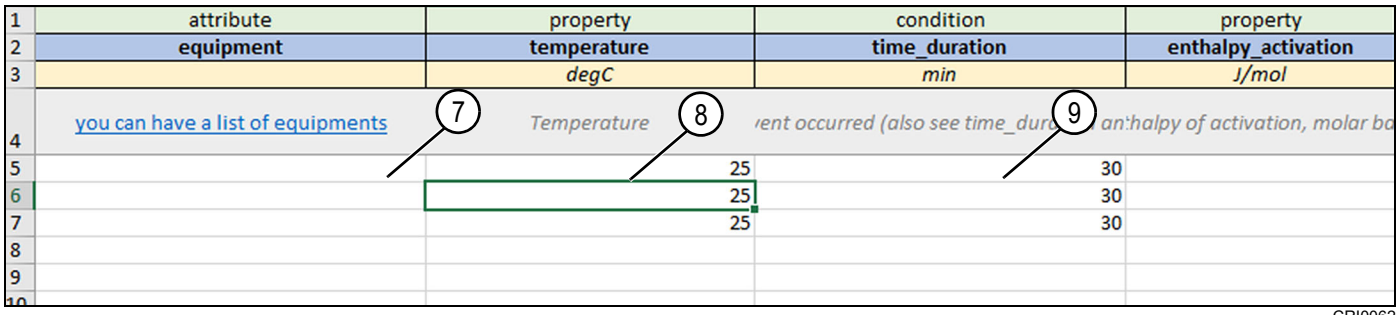
1. Experiment value (1) is derived from the " *name" column of the experiment sheet.
2. Name (2) is a unique name for the process.
3. Type (3) is selected from the name column of Process Type file found at CRIPT Home > Help > Controlled vocabulary > Process type, or use the following hyperlink "Process Type".
4. Keywords (4) is selected from the name column of Process keywords found at CRIPT Home > Help > Controlled vocabulary > Process keyword, or use the following hyperlink "Process keywords".
5. Description (5) describes the process.
6. Notes (6) allows user to enter notes regarding process.



CRI0055

Figure 2-25

7. Equipment (7) is selected from the list of equipment found at CRIPT Home > Help > Controlled vocabulary > Equipment key, or use the following hyperlink "equipments".
8. Columns H and beyond are customizable. Some examples would include.
   - Process property temperature (8) is selected from the list of Process property found at CRIPT Home > Help > Controlled vocabulary > Process property key, or use the following hyperlink "process property".
   - Condition time_duration (9), is selected from the list of Condition Keys at CRIPT Home > Help > Controlled vocabulary > Process property key, or use the following hyperlink "process property".



CRI0062

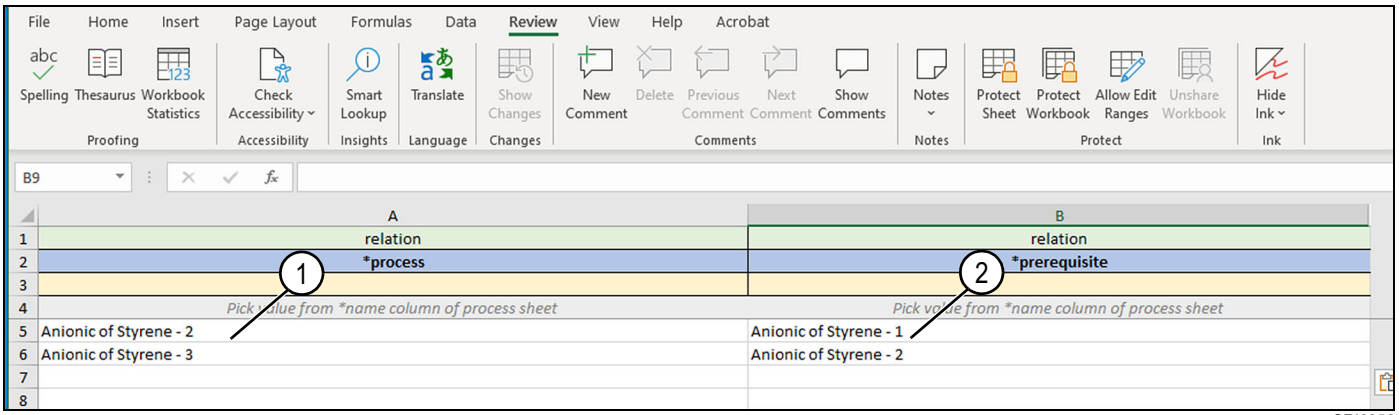
Figure 2-26

PREREQUISITE PROCESS SHEET

Define the immediate prerequisites for each process.

1. Process (1) is derived from the " *name" column of the process sheet.
2. Prerequisite (2) is derived from the " *name" column of the process sheet.

NOTE: *Each prerequisite must be from the previous process.*



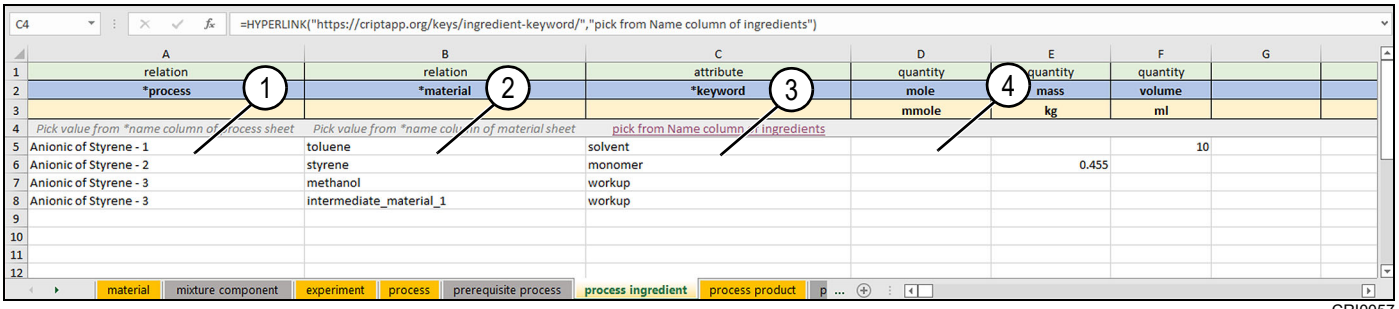
CRI0056

Figure 2-27

PROCESS INGREDIENT SHEET

Define the ingredients for each process and their respective quantities.

1. Process (1) is derived from the " *name" column of the process sheet.
2. Material (2) is derived from the " *name" column of the material sheet.
3. Keyword (3) is selected from the list of ingredients found at CRIPT Home > Help > Controlled vocabulary > Ingredient keyword, or use the following hyperlink "Ingredient keyword".
4. Columns D and beyond are customizable. Some examples would include.
   - Quantity mole (4) is selected from the list of quantity values found at CRIPT Home > Help > Controlled vocabulary > Quantity key, or use the following hyperlink "Quantity key".



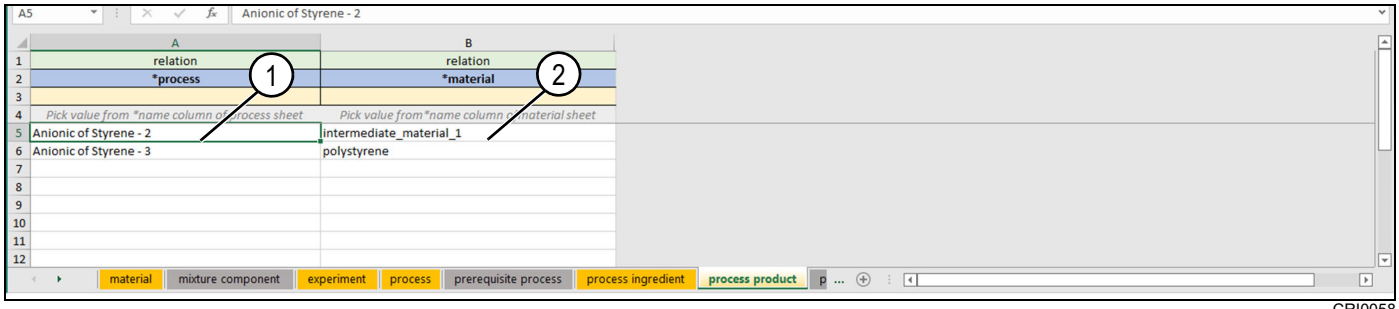
CRI0057

Figure 2-28

PROCESS PRODUCT SHEET

This sheet describes the resulting product after completing a process.

1. Process (1) is derived from the “ *name” column of the process sheet.
2. Material (2) is derived from the “ *name” column of the material sheet.



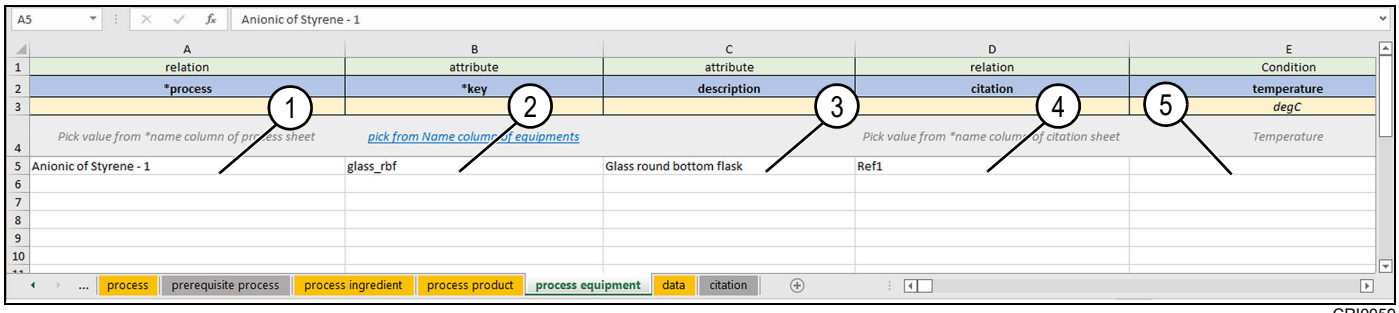
CRI0058

Figure 2-29

PROCESS EQUIPMENT SHEET

Define the equipment used in a process.

1. Process (1) is derived from the “ *name” column of the process sheet.
2. Key (2) is selected from the list of equipment found at CRIPT Home > Help > Controlled vocabulary > Equipment key, or use the following hyperlink “Equipment key”.
3. Description (3) describes the type of equipment.
4. Citation (4) is derived from the “ *name” column of the citation sheet.
5. Condition temperature (5) is selected from the list of conditions found at CRIPT Home > Help > Controlled vocabulary > Condition key, or use the following hyperlink “Condition key”.



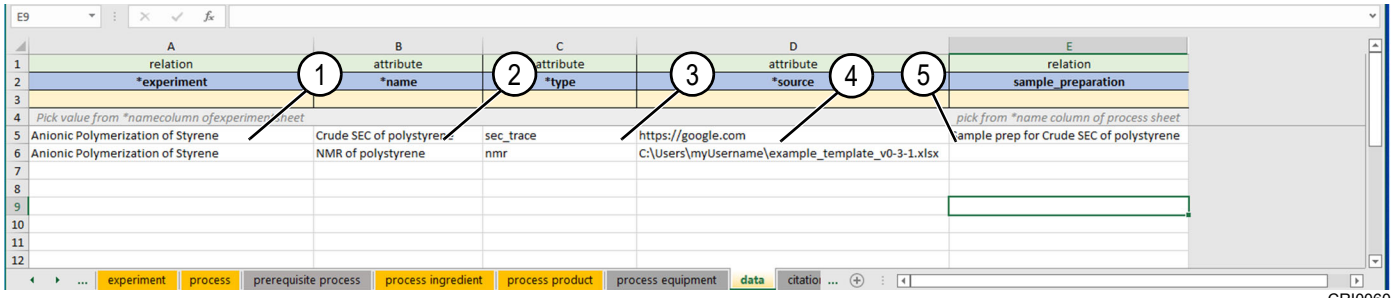
CRI0059

Figure 2-30

DATA SHEET

This sheet defines the data files you want to upload to CRIPT, such as a CSV file from a robot, an image, or any other type of file.

1. Experiment (1) is derived from the " *name" column of the experiment sheet.
2. Name (2) is a unique name for the experiment.
3. Type (3) is selected from the list of data types found at CRIPT Home > Help > Controlled vocabulary > Data type, or use the following hyperlink "Data type".
4. Source (4) can either be a path to a local file on your computer or a url to a website.
5. Sample_preparation (5) is derived from the " *name" column of the process sheet.



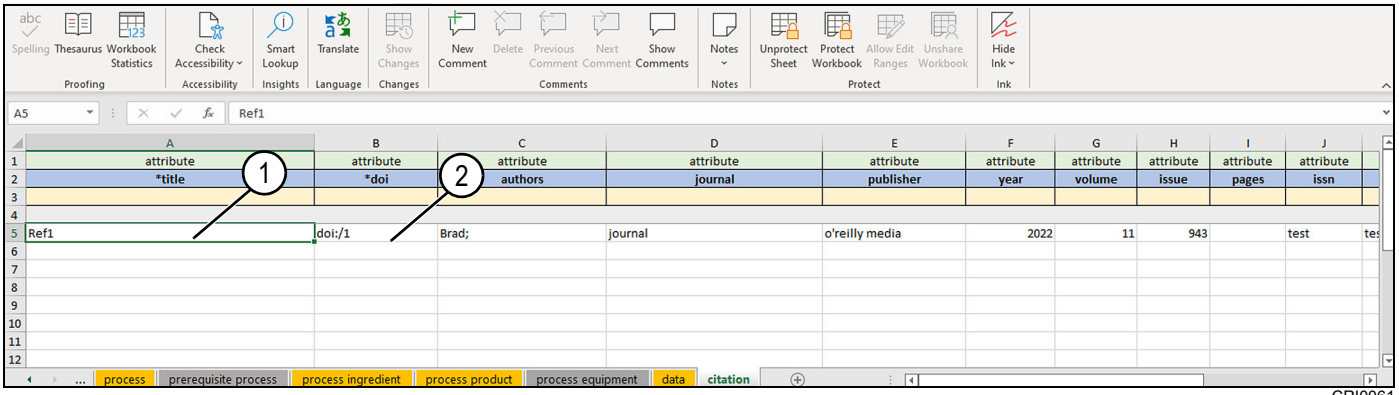
CRI0060

Figure 2-31

CITATION SHEET

This sheet can be used to reference any sources used in the experiments that you want to cite in CRIPT.

1. Required columns are the title (1) and Digital object identifier (doi) (2). Optional information can be added in the additional columns.



CRI0061

Figure 2-32

Chapter 3

Python SDK

QUICKSTART

PREREQUISITES:

- Python version 3.9 or higher
- Internet access
- API Token. For more information, See “API Token” on page 2-3

INSTALLATION

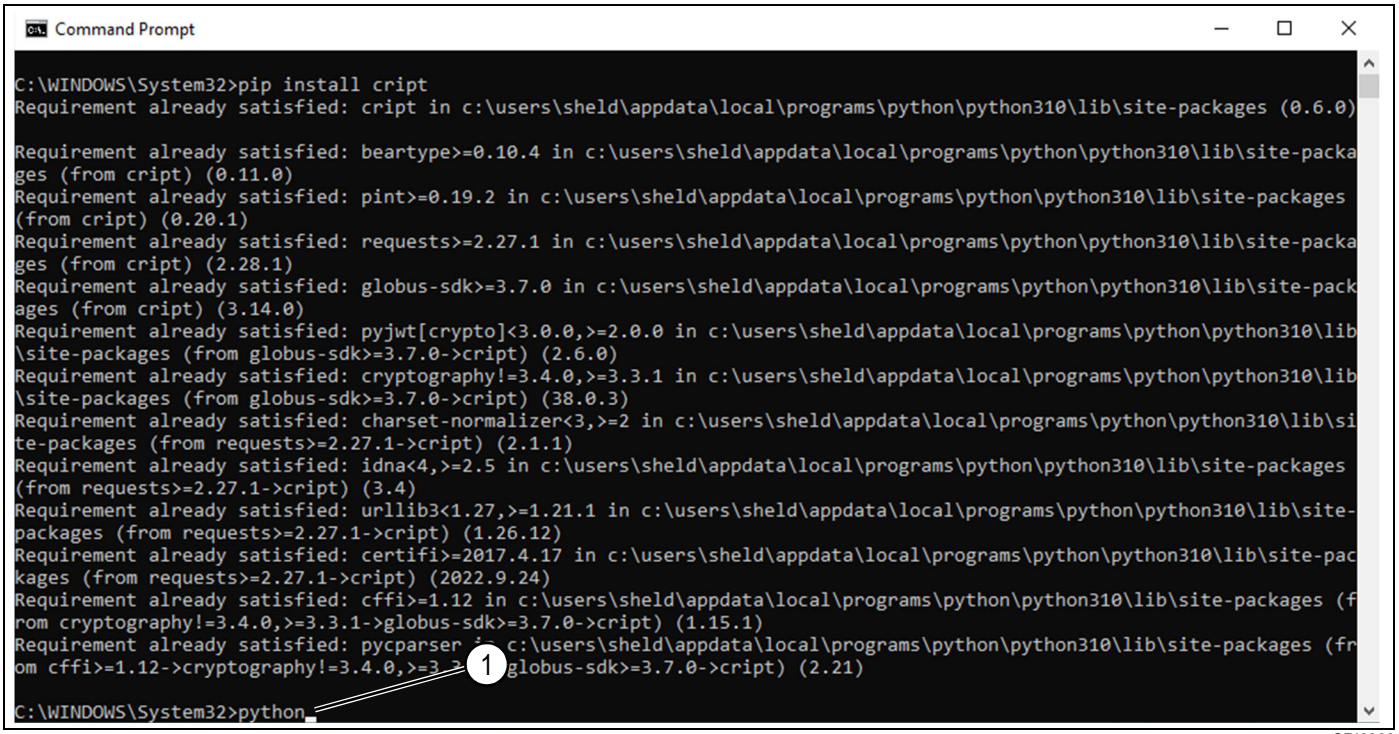
1. Download and install Python version 3.9 or higher.
2. Open the command prompt by selecting the Windows icon and type “CMD”.
3. Enter “pip install cript” to the command line (1) and press enter.

CRI0063

Figure 3-1

CONNECT TO CRIPT

1. Enter “python” at cursor (1) and press enter.



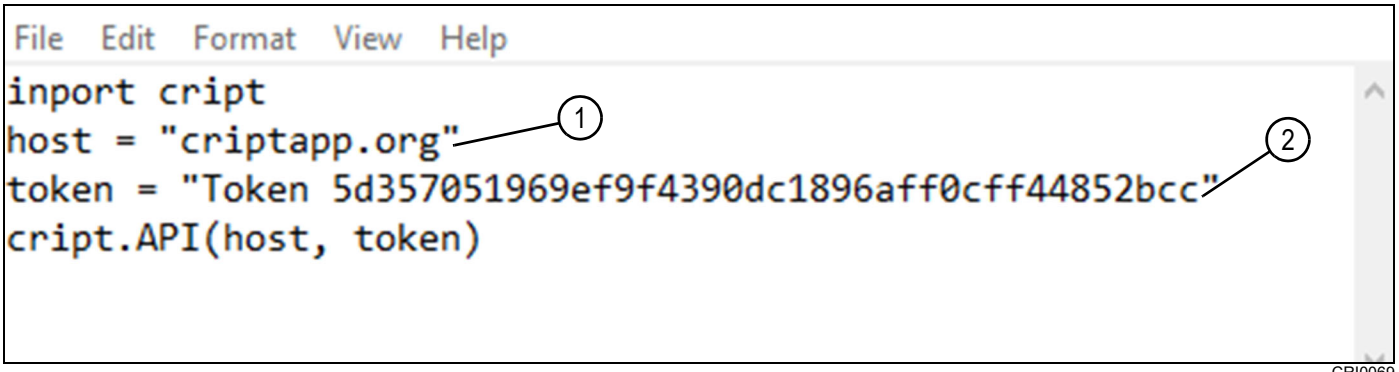
CRI0068

Figure 3-2

2. Open text editor (Microsoft Notepad shown).
3. Copy and paste the example code into notepad to modify.

```
host = "<endpoint_hostname>"  # e.g., criptapp.org
token = "<your_api_token>"
cript.API(host, token)
```

4. Modify the host line (1) with the application programming interface (API) destination.
5. Modify the token line (2) with the user token. See “API Token” on page 2-3.



CRI0069

Figure 3-3

6. Copy and paste modified host line from Notepad to the command prompt cursor (3) and press enter.
7. Copy and paste modified token line from Notepad to the command prompt cursor (4) and press enter.
8. Copy and paste cript.API line from Notepad to the command prompt cursor (5) and press enter.
9. Confirm connected to API site (6).

NOTE: *Each line must be entered individually.*

CRI0067

Figure 3-4

SAMPLE TASKS

The sample tasks are excellent templates to be used to record data. Replace sample tags in quotes with the name that applies to the project.

Create a node:

Create a Project and Collection node.

```
proj = cript.Project(name="MyProject")proj.save()
```

```
coll = cript.Collection.create(project=proj, name="MyCollection")
```

Create an Experiment node.

```
expt = cript.Experiment(
    collection=coll,
    name="Anionic Polymerization of Styrene with SecBuLi"
)
expt.save()
```

NOTE: *The use of create() here, instantiates and saves the object at the same time.*

Update a node:

Update the Project and Collection node created above.

```
proj.name = "OurProject"
proj.save()
```

```
coll.update(name="OurCollection")
```

NOTE: *The use of update() here, updates and saves a node at the same time.*

Delete a node:

Delete the Collection node created above.

```
coll.delete()
```

Get an existing node:

Get the official CRIPT Project node.

```
proj = cript.Project.get(name="CRIPT")
```

Get the official styrene Material node via name.

```
styrene =  cript.Material.get(project=proj.uid, name="Styrene")
```

Get the official styrene Material node via UID.

```
styrene =  cript.Material.get(uid="<material_uid>")
```

Get the official styrene Material node via URL.

```
styrene =  cript.Material.get(url="<material_url>")
```

Get Material nodes:

Get an existing Inventory node from the database.

```
uid = "134f2658-6245-42d8-a47e-6424aa3472b4"
inv = cript.Inventory.get(uid=uid, get_level=1)
```

NOTE: *Get_level is set to 1 so that the Material nodes are auto-generated. This parameter defaults to 0, but can be set to any integer.*

Notice that the Material node objects have been auto-generated.

```
type(inv.materials[0])
# <class 'cript.data_model.nodes.material.Material'>
```

Run a search query:

Search for Material nodes with a molar mass of less than 10 g/mol.

```
res =  cript.Material.search(
    properties = [
        {
            "key": "molar_mass",
            "value__lt": 10,
            "unit": "g/mol"
        }
    ]
)
```

Page through the results.

```
res.json()              # View the raw JSON for the query
res.objects()           # Generate objects for the current page
res.next_page()         # Flip to the next page
res.previous_page()     # Flip to the previous page
```

Upload a file:

User will need a Project and Data node to upload a file.

```
proj = cript.Project.get(uid="<project_uid>")
data = cript.Data.get(uid="<data_uid>")
```

Create a File node that points to a local file.

```
path = "path/to/local/file"
f = cript.File(project=proj, source=path)
file.save()
```

Download a file:

Download the file that was uploaded above.

```
path = "path/to/local/file"
f.download_file(path=path)
```

NOTE: *The default path for a download is in the current directory.*

Create a Process node:

```
prcs = cript.Process(
    experiment=expt,
    name="Anionic of Styrene",
    type = "multistep",
    description = "In an argon filled glovebox, a round bottom flask was filled with 216 ml of dried toluene. The "
                  "solution of secBuLi (3 ml, 3.9 mmol) was added next, followed by styrene (22.3 g, 176 mmol) to "
                  "initiate the polymerization. The reaction mixture immediately turned orange. After 30 min, "
                  "the reaction was quenched with the addition of 3 ml of methanol. The polymer was isolated by "
                  "precipitation in methanol 3 times and dried under vacuum."
)
prcs.save()
```

Add Ingredient nodes to the Process node:

Get the Material nodes from the Inventory node.

```
solution = inv['SecBuLi solution 1.4M cHex']
toluene = inv['toluene']
styrene = inv['styrene']
butanol = inv['1-butanol']
methanol = inv['methanol']
```

Define Quantity nodes indicating the amount of each Ingredient.

```
initiator_qty = cript.Quantity(key="volume", value=0.017, unit="ml")
solvent_qty = cript.Quantity(key="volume", value=10, unit="ml")
monomer_qty = cript.Quantity(key="mass", value=0.455, unit="g")
quench_qty = cript.Quantity(key="volume", value=5, unit="ml")
workup_qty = cript.Quantity(key="volume", value=100, unit="ml")
```

Create Ingredient nodes for each:

```
initiator = cript.Ingredient(
    keyword="initiator",
    material=solution,
    quantities=[initiator_qty]
)
solvent = cript.Ingredient(
    keyword="solvent",
    material=toluene,
    quantities=[solvent_qty]
)
monomer = cript.Ingredient(
    keyword="monomer",
    material=styrene,
    quantities=[monomer_qty]
)
quench = cript.Ingredient(
    keyword="quench",
    material=butanol,
    quantities=[quench_qty]
)
workup = cript.Ingredient(
    keyword="workup",
    material=methanol,
    quantities=[workup_qty]
)
```

Add Ingredient nodes to the Process node:

```
prcs.add_ingredient(initiator)
prcs.add_ingredient(solvent)
prcs.add_ingredient(monomer)
prcs.add_ingredient(quench)
prcs.add_ingredient(workup)
```

Add Condition nodes to the Process node:

```
temp = cript.Condition(key="temperature", value=25, unit="celsius")
time = cript.Condition(key="time_duration", value=60, unit="min")
prcs.add_condition(temp)
prcs.add_condition(time)
```

Add Property node to the Process node:

```
yield_mass = cript.Property(
    key="yield_mass",
    value=0.47,
    unit="g",
    method="scale"
)
prcs.add_property(yield_mass)
```

Create a Material node (process product):

Instantiate the node.

```
polystyrene = cript.Material(project=proj, name="polystyrene")
```

Add Identifier nodes.

```
names = cript.Identifier(
    key="names",
    value=["poly(styrene)", "poly(vinylbenzene)"]
)
bigsmiles = cript.Identifier(
    key="bigsmiles",
    value="[H]{[>][<]C(C[>])c1ccccc1[<]}C(C)CC"
)
chem_repeat = cript.Identifier(key="chem_repeat", value="C8H8")

polystyrene.add_identifier(names)
polystyrene.add_identifier(chem_repeat)
polystyrene.add_identifier(bigsmiles)
```

Add Property nodes.

```
phase = cript.Property(key="phase", value="solid")
color = cript.Property(key="color", value="white")

polystyrene.add_property(phase)
polystyrene.add_property(color)
```

Save the Material and add it to the Process node as a product.

```
polystyrene.save()
prcs.add_product(polystyrene)
```

Save the Process node.

```
prcs.save()
```

Create a File node and upload a file:

Instantiate a File node and associate it with the Data node created above.

```
path = "path/to/local/file"
f = cript.File(project=proj, source=path)
```

NOTE: *The source field should point to a file on the local filesystem.*

NOTE: *Depending on the file size, there could be a delay while the checksum is generated.*

Upload the local file by saving the File node. Follow all prompts that appear.

```
api.save(f)
```

Create a Data node:

```
sec = cript.Data(
    experiment=expt,
    name="Crude SEC of polystyrene",
    type="sec_trace",
)
```

Add the uploaded File to the Data node.

```
sec.add_file(f)
sec.save()
```

Associate a Data node with a Property node:

Create one more Property node for polystyrene.

```
mw_n = cript.Property(key="mw_n", value=5200, unit="g/mol")
```

Add the Data node to the new Property node.

```
mw_n.data = sec
```

Add the new Property node to polystyrene, then save it.

```
polystyrene.add_property(mw_n)
polystyrene.save()
```

Page Intentionally Blank

Intentionally Left Blank Page


{ RIPT }