

NANOSPACE USER GUIDE

1. Create an account and login

Before using Nanospace, it is necessary to register as a user. The work done will then be accessible only to you or to other users you have added. Click on the 'subscribe' button, and fill in the form before registering. You can now come back on the login menu [1] and enter your login / password.

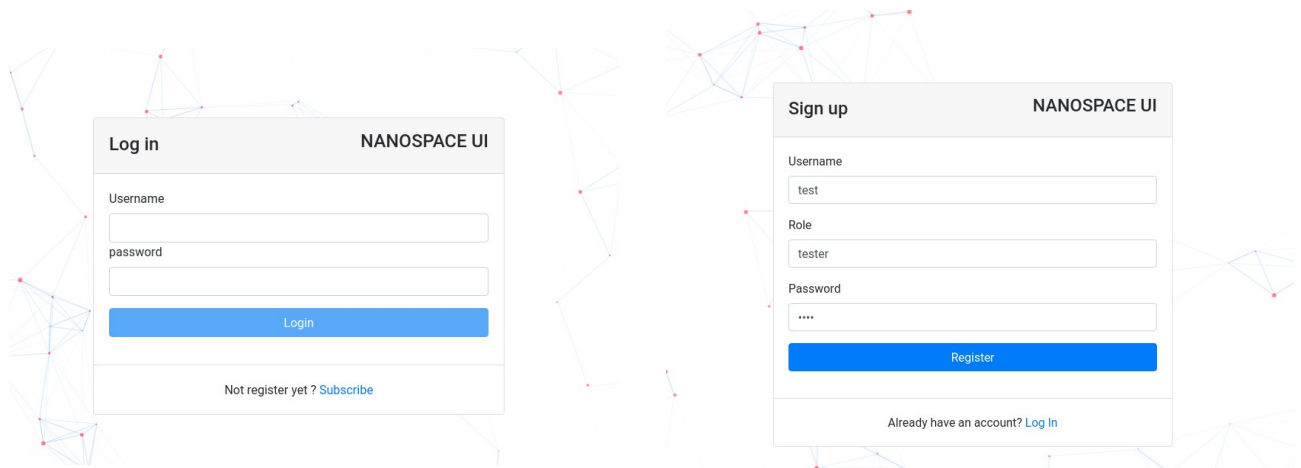


Image 1 - Login and subscribe screen

2. Create or import a project

Click on the *project* menu [2] at the top right of the navbar to access the project creation and import menu. It is also listing all the project in the database you have already access. The project imported **must be** a project created with nanospace which has been exported previously. A notification should confirm the project has been created.

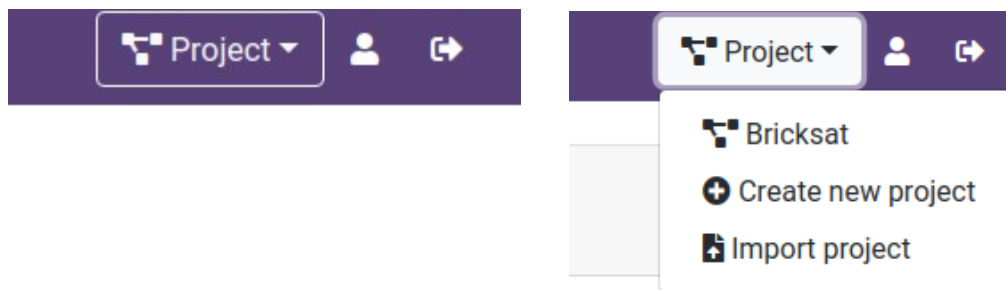


Image 2 – Project menu

3. Manage component and project model

A sidebar should have appeared since you created, imported, or opened the project. This sidebar is representing all the components (or sub-systems) à in a project tree. The top element is your project and the other one are the composition of components. A right-click on one of those entities is opening a context menu [3]. It let you create a children component, remove the component or rename it.

A left-click on a component permitted to select it and to display detail on the left side of the screen. Details are composed with a list of mode and values which are characterizing the mode selected.

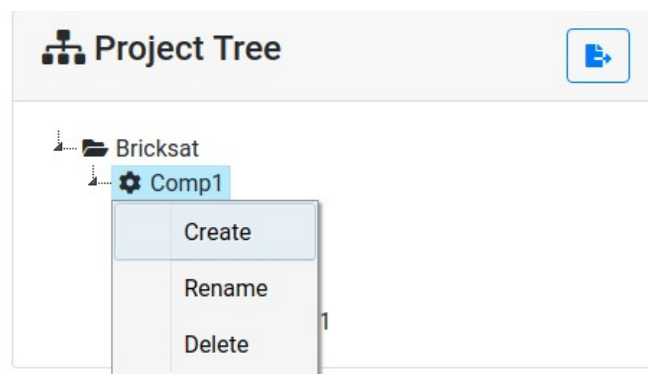


Image 3 – Context-menu on the sidebar ‘Project Tree’

A left-click on the project (top-level entity) permit to have access to a panel [4] to add and delete users.

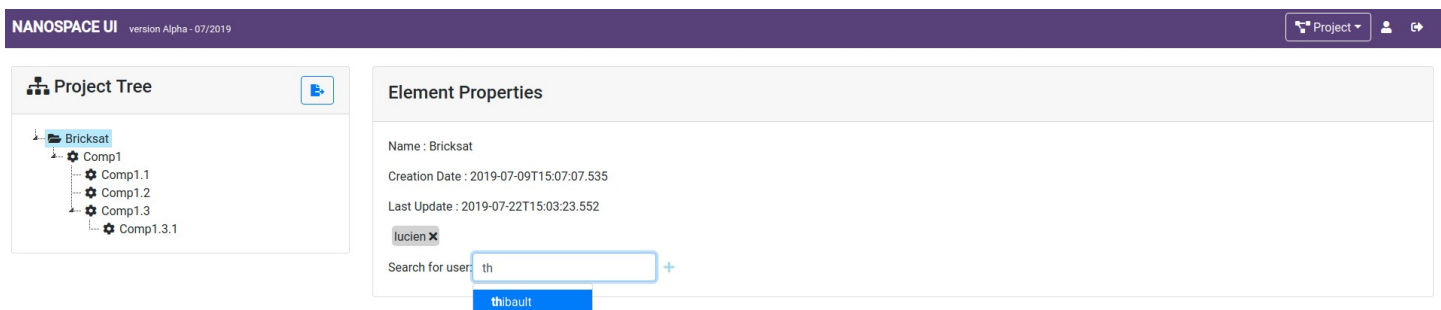


Image 4 – Add users to your project

4. Manage Modes

Like indicated above, modes and values list are accessible by selecting a component. In this view, it is possible to create, rename, and delete modes. The default mode is called 'nominal', it cannot be deleted or renamed. About the others modes, a double click on the mode name permits to edit it. It is possible to create a new mode by clicking on the add button on the right of the mode [5].

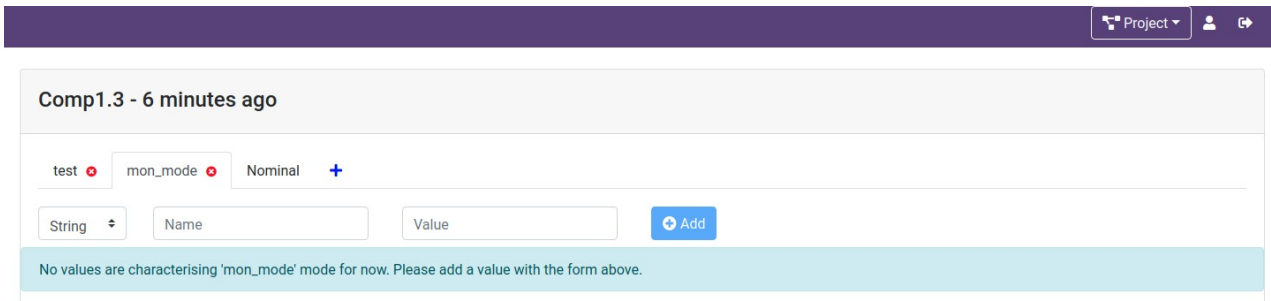


Image 5 – Modes in the project

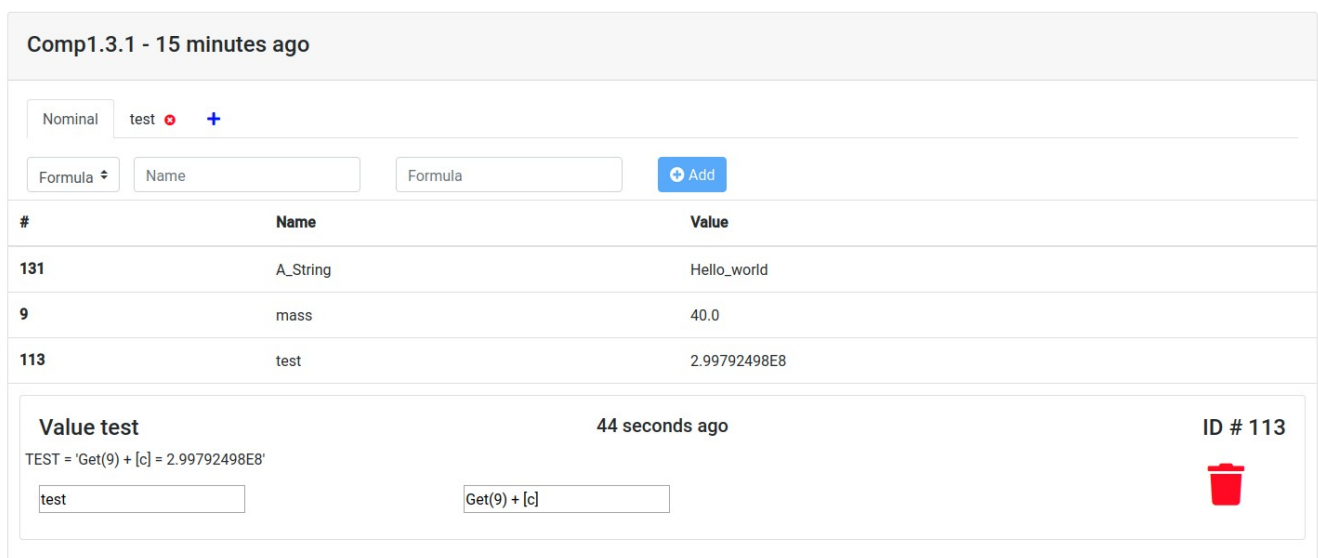
5. Values : String, Matrix, and Formula

You can use the values formula under the modes tabs to add new values to the selected modes.

Several choices of values are possible: String, Matrix, Formula.

You will be required to complete the name of the value, then you can fill the value itself according to his type.

By clicking on the value on the list you will open a new menu which allows you to update or delete it [6].



#	Name	Value
131	A_String	Hello_world
9	mass	40.0
113	test	2.99792498E8

Value test 44 seconds ago ID # 113

TEST = 'Get(9) + [c] = 2.99792498E8'




Image 6 – Value list and value menu.

5.1. String

String is the most basic element. The values are just composed of a chain of character. There is currently no limit to the size of the value for now. Obviously, load a heavy string value is going to be slower than a shorter one.

5.2. Matrix

Matrix are dynamics 2-dimensional arrays. As user, you will need to enter the number of columns, and rows that your matrix needs and then complete the matrix that has been dynamically created to fit with the parameter you asked [8].

#	Name	Value
131	A_String	Hello_world
9	mass	40.0

Image 8 – Matrix form

5.3 Formula

Formula type permit to enter numbers and formula using numbers. It is based on the mXparser syntax. Mains units, mathematical constants, and astronomical constants are implemented in the interpreter. Moreover, two other functionalities have been implemented in order to manipulate data in your project.

The first one is the 'Get()' function which returns the value of the formula passed in parameter. In order to do that you need to pass an Id as the argument of the function [9]. For example Get(153).

The second one is the SoC() function. This function is a bit specific permit to Sum all the data with a common name and modes.

Ex : In a following tree

Bricksat

— *Structure*

--- *Mass : SoC()*

— *Solar Pannel*

— *Mass : 150 * 4*

— *Communication*

— *Antenna :*

— *Mass : 300*

— *Controller :*

— *Mass : 100*

*The SoC() result is going to be : $150 * 4 + 300 + 100$*

Notes that for the moment, it is not possible to make a SoC() of a SoC(). In case of error the answer is going to be NaN.

Comp1.3.1 - 11 minutes ago

Nominal

test ⊖ +

Formula ▾

A_Formula

Get(9) + 80.5 + [c]

+ Add

#	Name	Value
131	A_String	Hello_world
9	mass	40.0

Image 9 – Formula form

6. Export your project in JSON

A button export the project is available on the left side on the top of the project tree [3]. It is important to do not modify the JSON if you want it to be importable later.