

JSON guide for LCAbyg 2023 (5.3.1.0)

Creation of JSON projects in LCAbyg - scenarios

Title	JSON guide for LCAbyg version 5
Subtitle text	Creation of JSON projects in LCAbyg - scenarios
Published	December 2022
Last update	-
Authors	Terese Pagh, Christian Grau Sørensen, Anne Mathilde Gøtke
Keywords	json, JavaScript Object Notation, LCAbyg
Publisher	BUILD - Institut for Byggeri, By og Miljø, Aalborg Universitet, A.C. Meyers Vænge 15, 2450 København SV
Email	lcabyg@build.aau.dk
Website	https://www.lcabyg.dk/da/

Contents

1	Building scenarios	2
1.1	Activate and create building scenarios in LCAbyg	2
2	Create building scenarios in JSON	4
2.1	Update your engine.yalm file	4
2.2	Create the project.json file with several scenarios	5
2.3	Create the building.json file with several scenarios	5
2.4	Create the remaining json file with several scenarios	7
3	Select which components to include in a scenario	8
4	Export project with several scenarios from LCAbyg to JSON	9

Disclaimer

LCAbyg is under constant development and for that reason, there may be changes in the JSON files. Always remember to download the latest version of LCAbyg and download the latest version of the JSON guide, which you can find here <https://www.lcabyg.dk/download-program>. Sign up for our newsletter here <https://www.lcabyg.dk/news-letter>, if you do not want to miss the latest updates and bug fixes.

Readme

Before starting, you should read and understand the basics in the **JSON guide for LCAbyg 5 Creation of JSON projects in LCAbyg - import** which can be found here <https://www.lcabyg.dk/da/usermanual/brugervejledning-andre-vaerktojer/>.

1 Building scenarios

In both LCAbyg and json you have the option to clone your entire building model (MainBuilding) by creating a new scenario of the building model (ExtraBuilding). The new clone can be modified to represent a different scenario of the project.

The naming convention in json and LCAbyg is shown in Figure 1.

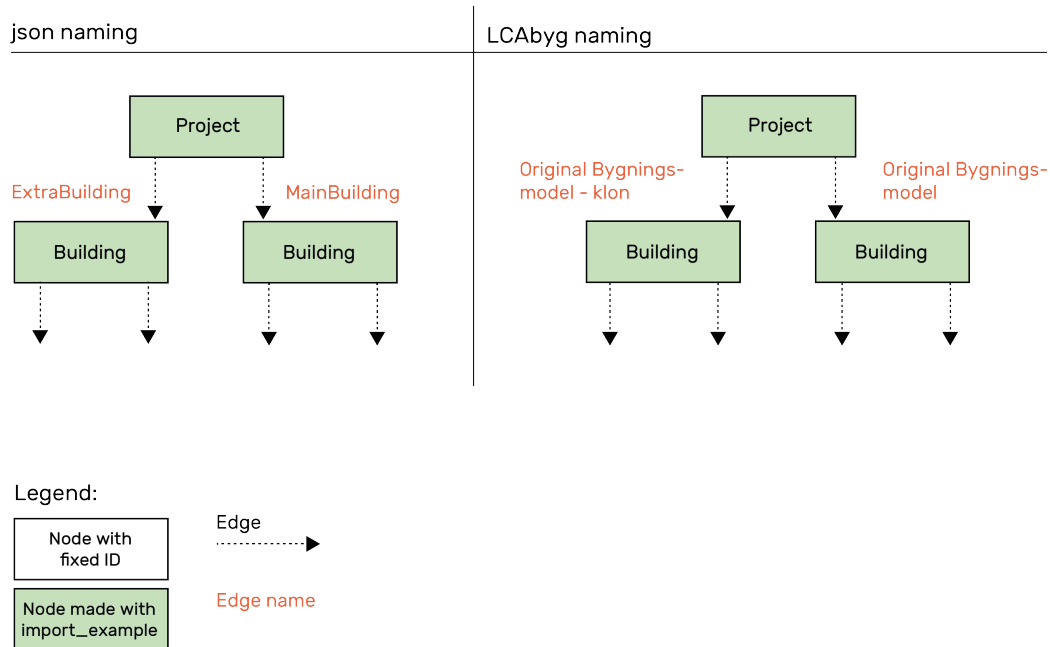


Figure 1: Naming convention in JSON vs. LCAbyg for the Building Model with two scenarios "ExtraBuilding" and "MainBuilding"

1.1 Activate and create building scenarios in LCAbyg

Inside the LCAbyg application, a new scenario is activated by clicking on the activation button found at the menu "Bygning og drift", see Figure 2. When activating scenarios, a new menu then appears to the right inside the "Buildings model" menu.

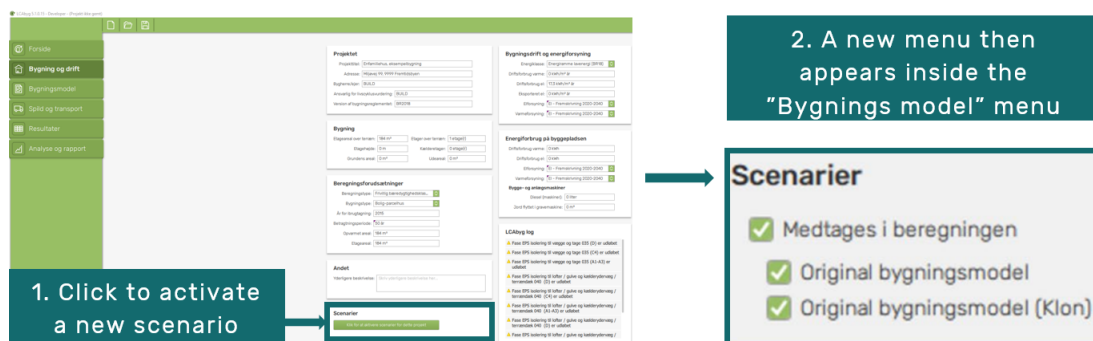


Figure 2: Activate building scenario inside LCAbyg

It is possible to create more than two scenarios by right clicking one of the scenarios as shown in Figure 3. Per default, each new building scenario will have clone, added to its original name. In JSON this corresponds to a new ExtraBuilding with a new ID. Other functionalities found here are:

- renaming a scenario
- locking a scenario so that it cannot be edited
- duplicating a scenario
- organizing the order of the scenarios
- and deleting a scenario and organization.

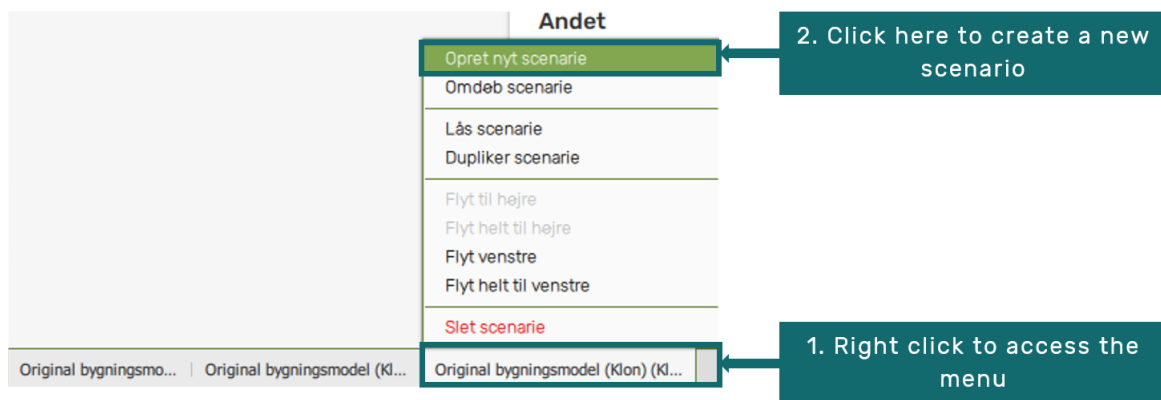


Figure 3: Create a new building scenario inside LCAbyg

2 Create building scenarios in JSON

You can create a new project template like the `import_example`. By creating a new project template, you can change e.g. the defaults for a project in LCAbyg, which can be beneficial if you intend to use the same template for multiple projects.

In this example, the project template named `import_example_scenarios` has been created, which can be used and modified by the user. The project template already contains several JSON files that can be modified and added to. There are two ways to import the example in LCAbyg, which is described in the **JSON guide for LCAbyg 5 Creation of json projects in LCAbyg - import** chapter 2 and chapter 4.

2.1 Update your engine.yalm file

When creating a new project always remember to update the `engine.yalm` file. See example for `import_example_scenarios` in figure 3.

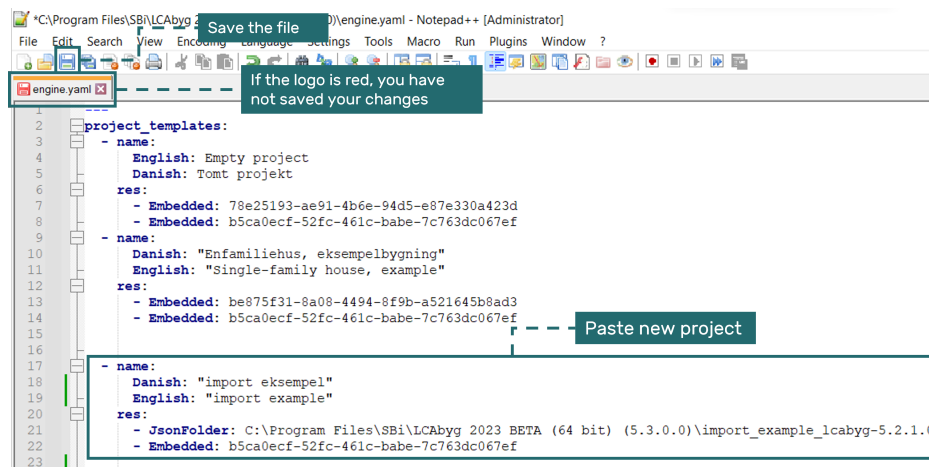


Figure 4: Remember to update and save the `engine.yaml` file after placing and editing the new project template

Note: You can also open new project using the debug version of LCAbyg named `lcabyg_debug.exe`.

You can check if the new project template appears in LCAbyg by opening the program as usual. Click on the "Opret nyt projekt" logo in the menu in the upper left corner. Click on "Ny" and then check if your new project template appears.

2.2 Create the project.json file with several scenarios

Below is an example of the project.json file found in the import_example_scenarios folder. In this file, you can find the building ID of each building scenario listed in the order they have been created in LCAbyg.

```
1 [
2   {
3     "Node": {
4       "Project": {
5         "id": "e9e6e798-390e-4419-a1fa-3b46a8ba5b8d",
6         "name": {
7           "Danish": "Enfamiliehus, eksempelbygning"
8         },
9         "address": "Miljøvej 99, 9999 Fremtidsbyen",
10        "owner": "BUILD",
11        "lca_advisor": "BUILD",
12        "building_regulation_version": "BR2018"
13      }
14    },
15  },
16  {
17    "Edge": [
18      {
19        "ExtraBuilding": "169a1f82-fcc1-4897-acce-242e1f981d86"
20      },
21      "e9e6e798-390e-4419-a1fa-3b46a8ba5b8d",
22      "915c5162-93bc-49c0-bdaf-9f470b4c7996"
23    ]
24  },
25  {
26    "Edge": [
27      {
28        "MainBuilding": "3556869a-7e28-4ada-91bb-d535b1604d37"
29      },
30      "e9e6e798-390e-4419-a1fa-3b46a8ba5b8d",
31      "af026bf8-689a-4bb6-ba1f-d49d9444acb1"
32    ]
33  }
34 ]
```

Listing 1: Example code for the Project.json

2.3 Create the building.json file with several scenarios

```
1 [
2   {
3     "Node": {
4       "Building": {
5         "id": "915c5162-93bc-49c0-bdaf-9f470b4c7996",
6         "scenario_name": "Original bygningsmodel",
7         "locked": "Unlocked",
8         "description": "",
9         "building_type": "ResidentialBuildingSingleFamilyHouse",
10        "heated_floor_area": 184.0,
11        "gross_area": 184.0,
12        "gross_area_above_ground": 184.0,
13        "storeys_above_ground": 1,
14        "storeys_below_ground": 0,
15        "storey_height": 0.0,
16        "initial_year": 2015,
17        "calculation_timespan": 50,
18        "calculation_mode": "SC",
19        "outside_area": 0.0,
20        "plot_area": 0.0,
21        "energy_class": "LowEnergy"
22      }
23    }
24  ]
```



```

22     }
23   }
24 },
25 {
26   "Node": {
27     "Building": {
28       "id": "af026bf8-689a-4bb6-ba1f-d49d9444acb1",
29       "scenario_name": "Original bygningsmodel (Klon)",
30       "locked": "Unlocked",
31       "description": "",
32       "building_type": "ResidentialBuildingSingleFamilyHouse",
33       "heated_floor_area": 184.0,
34       "gross_area": 184.0,
35       "gross_area_above_ground": 184.0,
36       "storeys_above_ground": 1,
37       "storeys_below_ground": 0,
38       "storey_height": 0.0,
39       "initial_year": 2015,
40       "calculation_timespan": 50,
41       "calculation_mode": "SC",
42       "outside_area": 0.0,
43       "plot_area": 0.0,
44       "energy_class": "LowEnergy"
45     }
46   },
47 },
48 {
49   "Edge": [
50     {
51       "BuildingToRoot": "bc35b94d-b8c0-4b8c-9bf3-3f63acc94063"
52     },
53     "915c5162-93bc-49c0-bdaf-9f470b4c7996",
54     "216cf5d6-3e9d-43ec-b0d8-5aee02240c28"
55   ]
56 },
57 {
58   "Edge": [
59     {
60       "BuildingToOperation": "0c35b94d-b8a0-4bec-92f3-3463acc94064"
61     },
62     "915c5162-93bc-49c0-bdaf-9f470b4c7996",
63     "0338d31e-3876-440d-a88c-2daa2dd81942"
64   ]
65 },
66 {
67   "Edge": [
68     {
69       "BuildingToDGNBOperation": "47d464b8-ce4e-4351-a0d1-0f69dc70cec1"
70     },
71     "915c5162-93bc-49c0-bdaf-9f470b4c7996",
72     "3a96ed87-8a5c-49b8-82d6-2fd56962e4bb"
73   ]
74 },
75 {
76   "Edge": [
77     {
78       "BuildingToDGNBOperation": "df716186-4d42-4be0-8321-e91c138dc166"
79     },
80     "af026bf8-689a-4bb6-ba1f-d49d9444acb1",
81     "a7d9206f-5345-4748-9e81-2882908df72e"
82   ]
83 },
84 {
85   "Edge": [
86     {
87       "BuildingToOperation": "1003f6f0-b3aa-4dc8-919c-26d7df936d73"

```

```

88     },
89     "af026bf8-689a-4bb6-ba1f-d49d9444acb1",
90     "1e4fd404-1c1c-42fc-9b85-e0a3e878b587"
91 ]
92 },
93 {
94     "Edge": [
95         {
96             "BuildingToRoot": "7a61df12-3311-4950-9c1e-3c2be1a53dd6"
97         },
98         "af026bf8-689a-4bb6-ba1f-d49d9444acb1",
99         "ed03c5a3-5f10-47f3-a7b2-d2bc93ac08ab"
100     ]
101 }
102 ]

```

Listing 2: Example code for the Building.json

2.4 Create the remaining json file with several scenarios

The JSON files for the other nodes edges must be listed chronologically accordingly to the order of your building scenarios. If you decide to list the ExtraBuilding before the MainBuilding, then you should keep the order for all the components for the two building scenarios.

3 Select which components to include in a scenario

For building scenarios, the default setting is that all elements, constructions, and products are included. This means that you only have to deactivate the items that you wish to exclude from the scenario.

Figure 5 shows an example of an element node that is included in the imagined scenario. If the key "excluded_scenarios" holds an empty list, indicated with [], as a value, then the specific construction is included in the scenario.

Example include element in scenarios	Comment
{	Begin dictionary
"Node": {	Begin node
"Element": {	Begin element
"id": "654d4204-990f-4b34-a74d-47e87bcd0d16",	Existing ID of the element
"name": {	
"Danish": "Randfundamenter",	
"English": "",	
"German": ""	
},	
"source": "User",	
"comment": "",	
"enabled": true,	
"excluded_scenarios": []	The empty list indicates that the elements is included in the scenario.
}	
}	
}	

Figure 5: Example on how to include a component in a scenario

Figure 6 shows an example of an element node that is excluded in the imagined scenario. If the key "excluded_scenarios" contains an ID, then it means that the construction is excluded from that specific scenario.

Example include element in scenarios	Comment
{	Begin dictionary
"Node": {	Begin node
"Element": {	Begin element
"id": "654d4204-990f-4b34-a74d-47e87bcd0d16",	Existing ID of the element
"name": {	
"Danish": "Randfundamenter",	
"English": "",	
"German": ""	
},	
"source": "User",	
"comment": "",	
"enabled": true,	
"excluded_scenarios": "ea222293-b17d-490c-b483-be22ab2ae440"	The existing ID of the specific building scenario the element is part of.
}	
}	
}	

Figure 6: Example on how to exclude a component in a scenario

4 Export project with several scenarios from LCAbyg to JSON

The entire project, can be exported from LCAbyg to JSON. This procedure is identical to the export of the project with one scenario described in the **JSON guide for LCAbyg 5 Creation of JSON projects in LCAbyg - export**.

The folder structure of the JSON output for more than one scenario follows the structure of a single project exported to JSON – described in detail in **JSON guide for LCAbyg 5 Creation of JSON projects in LCAbyg - import**. The cloned data is simply added to the existing JSON files.

2. Export results to json

The screenshot shows the LCAbyg 5.2.0.0 interface. The left sidebar contains navigation buttons: Forside, Bygning og drift, Bygningsmodel, Spild og transport, Resultater, and Analyse og rapport. The main area displays a table of results for various building components and scenarios. The right sidebar shows a 'Resultater' table with columns for Name, Value, and Unit. At the bottom, there are buttons for 'Original bygningsmodel (A1-A3)' and 'Original bygningsmodel (B1-B3)'. A green arrow points from the '2. Export results to json' box to the 'Eksporter json' button in the top toolbar. Another green arrow points from the '1. Select scenario to export to json results' box to the 'Original bygningsmodel (A1-A3)' button at the bottom.

1. Select scenario to export to json results

Figure 7: How to export results and quantities for each scenario to json