

How to connect the Project

Web: <https://dev2-openebench.bsc.es/>

Login (for testing → test):

- **Username:** test@bsc.es
- **Password:** ask someone

Open VM in Visual Studio in your laptop:

- > ssh -f -N -L 2249:10.32.1.249:22 user@oeb-login.bsc.es
- > sshfs -o port=2249 user@localhost:/home/LOCAL_FOLDER
- Once we have this locally, we access to the folder [openEbench-frontend/tree/development/src/app](#) and we can see all the components.

To do the build and see all on the website:

> npm run build-raw -- --optimization=false



User section → New Benchmarking Event

Create a form from a specific JSON Schemas to create a new benchmarking event (all automatically).

A screenshot of a web form titled 'Benchmarking Event concept (ELIXIR-EXCELERATE WP2 Benchmarking schemas)'. The form contains several input fields and a checkbox. The fields are: 'Benchmarking Event ID *' with an example 'Benchmarking_event_322', 'The benchmarking event original id' with an example 'ORIG_ID_1999', 'Name *' with an example 'Some Event Name', 'Description' with an example 'This is the description about the benchmarking event', and 'Community ID' with a dropdown menu showing 'Quest for Orthologs'. There is also a checkbox labeled 'Is this an automated event?' which is currently checked. The form is displayed in a browser window with a gray border.

Where the JSON Schemas are?

<https://github.com/inab/OpEB-VRE-schemas/tree/frontend-schema>

<https://github.com/inab/OpEB-VRE-schemas/tree/frontend-mode-schema>

These second schemas are needed for the form to be created correctly since it has a lot of \$references to other files. José María made a parser to make it work correctly. This parser needs this structure between schemas.

What library is used for the form?

<https://formly.dev/>

<https://formly.dev/examples/advanced/json-schema> → example about JSON Schemas with forms and the correspondent library.

The form creates everything but the selections with information from the API.

Where is the code in which the form is created?

<src/app/BDMCreator> The code is commented.

The form has types. This means that if the schema has element X is type = object, here appears an object element, etc. These types are in <src/app/BDMCreator/schema-types>

Where the select and “new” button are created?

<src/app/BDMCreator/schema-types/BDMLoader>

BDMLoader:

- **How does the form know that a select should appear?** To make a selection appear in a desired place on the form, it is with type = BDMLoader inside the “widget” and the “formlyConfig” section, as in the following image:

```
"community_id": {
  "title": "Community ID",
  "description": "The community id where this benchmarking event belongs",
  "type": "string",
  "foreign_keys": [
    {
      "schema_id": "Community",
      "members": [ "." ]
    }
  ],
  "minLength": 1,
  "widget": {
    "formlyConfig": {
      "type": "BDMLoader",
      "templateOptions": {
        "short_name": "community",
        "title": "Community ID",
        "BDMType": "community_id",
        "multiple": false,
        "url_select": {
          "url_staged": "https://dev2-openebench.bsc.es/api/scientific/staged/Community?privileged=true"
        }
      }
    }
  }
},
```

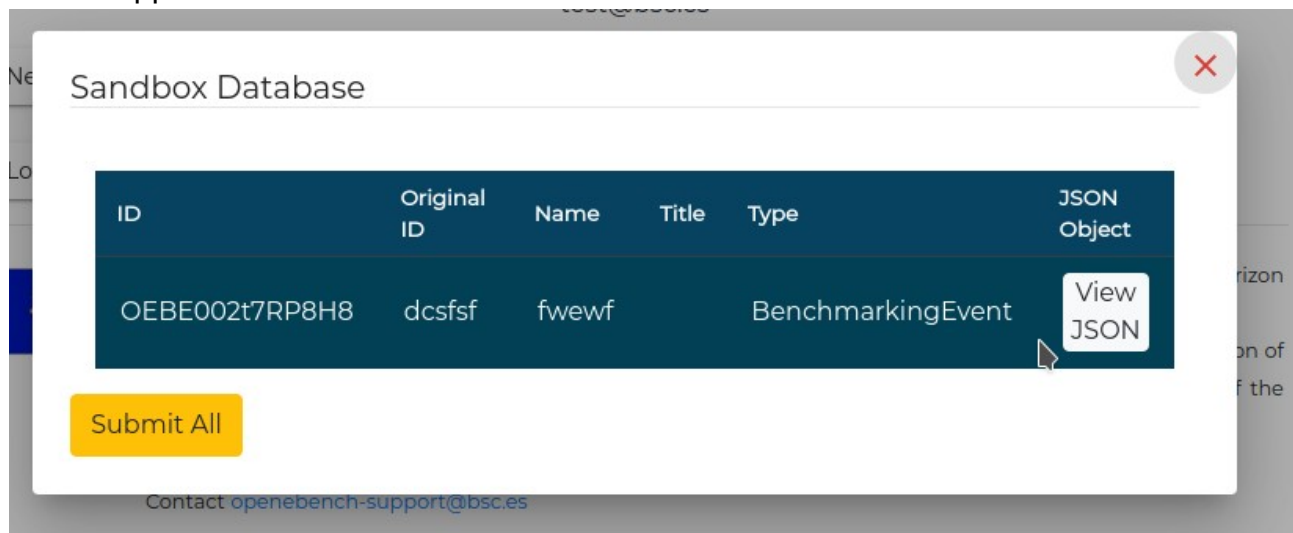
templateOptions from BDMLoader (JSON Schema):

1. **short_name:** It must be the same as the corresponding name of the file <https://github.com/inab/OpEB-VRE-schemas/tree/frontend-mode-schema>
2. **title:** Title that I want to appear when opening the corresponding form (if it has a new button).

3. **BDMType:** It must match the name of that attribute in the JSON Schema, in this case it is community_id.
4. **multiple:** If I want a multiple select this attribute is true, if not false
5. **new_button:** If it is = true, exist a "New" button to be able to create a new element of the same type. If it is = false or not exist, there is no "New" button.
6. **url_select:** It is used to get the information by making a GET to the API, it is the URL that it will use. There may be objects that can have staged and sandbox information. "Url_select" allows have two attributes, staged and sandbox. If either of them is not there, it simply does not read the correspondent.
 1. **url_staged:** If it is from the staged database.
 2. **url_sandbox:** If it is from the sandbox database.

User section → **Show sandbox**

A table appears with the information of the elements that are in sandbox database.



The screenshot shows a window titled "Sandbox Database" with a close button (X) in the top right corner. Inside the window is a table with the following data:

ID	Original ID	Name	Title	Type	JSON Object
OEBE002t7RP8H8	dcsfsf	fwewf		BenchmarkingEvent	View JSON

Below the table is a yellow button labeled "Submit All". At the bottom of the window, there is a text link: "Contact openebench-support@bsc.es".

Where is the code?

In [src/app/sandbox-selector](#)