| Library Navigator | Model1 | | Configuration | | | |

⊕ 🔍 🏠

Library name   `-` ▼

Configure new   `-` ▼

Project name   `-`

Model name   `-`

Configuration form (scrollable div)

This form is generated by the configuration widget based on the #data corresponding to the system model selected in `Configure new`.

| Update model | Instantiate model |
| Reset configuration | Delete model |

`Update model`
Implement the configuration options specified by the user in the `Configuration form` by instantiating and connecting objects in the file `#library/User/#project/#model`.

`Instantiate model`
Update and instantiate the class `#library/User/#project/#model` in the model loaded in the active tab of the main panel (here `Model1`).

`Reset configuration`
Reset the configuration options to the default values.

`Delete model`
Delete the class `#library/User/#project/#model` and all its instances in the model loaded in the active tab of the main panel (here `Model1`).

When no object is selected in the diagram view this is the default view for the `Configuration` panel.
The `Library name` is the last value selected (further referenced as `#library`). The drop down menu allows selecting between loaded libraries. The `Library name` is used to 1) load the configuration data stored in `#library/Configuration` directory, 2) define the root path of the directory where the built models will be saved i.e. `#library/User/*/`.
The `Configure new` drop down menu allows selecting the type of system model to configure. The menu is populated by `#data/#system.value` for all Configuration data files in `#library/Configuration`.
The `Project name` is the last value entered (further referenced as `#project`). A real-time form test is required to validate the user input against syntax requirements and avoid duplicate in `#library/User`. The path of the directory where the built models will be saved is `#library/User/#project`.
The `Model name` is by default `#data/#name.value` (further referred to as `#model`). It can be modified by the user (call a `rename_class` function if the model has already been saved). A real-time form test is required to validate the user input against syntax requirements and avoid duplicate in `#library/User/#project`.