For enabling NodeRED (version 0.17.5) to import and export flows to a Sna4City repository, we modified the core file of NodeRED installation in this way:

1. File: /usr/lib/node_modules/node-red/editor/templates/index.mst

We changed the lines:

```
<script src="{{ asset.red }}"></script>
<script src="{{ asset.main }}"></script>
```

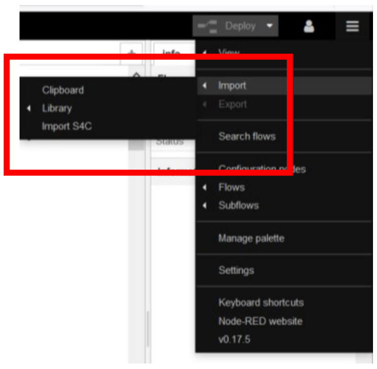
with

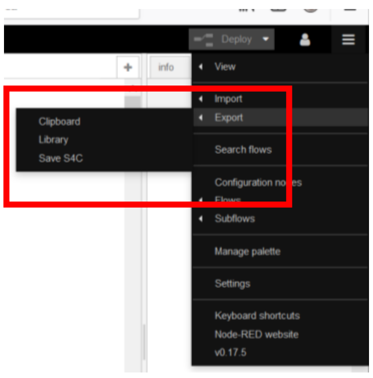
```
<script src="red/red.js"></script>
<script src="red/main.js"></script>
```

In order to load non compressed version of the core files of NodeRED

2. File: /usr/lib/node_modules/node-red/public/red/main.js

In the function loadEditor() we integrated the top-right menu of Nodered (depicted in the following figures) with two voices in the sections "Import" and "Export" for offering the possibility to import/export flows from/to the Snap4City repository.





The following code is used for populating these sections with the voices (in yellow in the code): "Import S4C" and "Save S4C".

```
menuOptions.push({id:"menu-item-import",label:RED._("menu.label.import"),options:[
        {id:"menu-item-import-clipboard",label:RED._("menu.label.clipboard"),onselect:"core:show-import-dialog"},
        {id:"menu-item-import-library",label:RED._("menu.label.library"),options:[]},
        {id:"menu-item-import-cloud",label:RED._("Import S4C"),onselect:"core:show-import-cloud-dialog"},
    ]});
    menuOptions.push({id:"menu-item-export",label:RED._("menu.label.export"),disabled:true,options:[
        /* s4c */
        {id:"menu-item-export-clipboard",label:RED._("menu.label.clipboard"),disabled:true,onselect:"core:show-export-dialog"},
        {id:"menu-item-export-library",label:RED._("menu.label.library"),disabled:true,onselect:"core:library-export"},
        {id:"menu-item-export-cloud",label:RED._("Save S4C"),disabled:false,onselect:"core:show-export-cloud-dialog"}
    ]});
```

3. File: /usr/lib/node_modules/node-red/public/red/red.js

For what concern the Export function we integrated the red.js file with the following code:

```
            {
                id: "cloud-dialog-copy",
                class: "primary",
                text: RED._("save S4C"),
                click: function() {
                   $("#clipboard-export").select();
                   var errorMessage = "";
                   if(
                      $("#clipboard-export").val()==""||
                      $("#clipboard-export").val()=="[]"||
                      $("#node-cloud-export-name").val()=="" ||
                      $("#node-cloud-export-res").val()=="" ||
                      $("#node-cloud-export-cat").val()=="" ||
                      $("#node-cloud-export-desc").val()==""

                   ){
                          errorMessage="All fields are mandatory";
                          $("#node-cloud-export-error-msg").html(errorMessage)
                          return;
                   }
                var cloudMessage = {
                          "name":$("#node-cloud-export-name").val(),
                          "user":RED.settings.user.username,
                          "app_type":"IoTApp",
                          "nature":$("#cloud-data-export-nature").val(),
                          "sub_nature": $('#cloud-data-export-subnature').val(),
                          "format":"json",
                          "licence":"https://creativecommons.org/licenses/by/4.0",
                          "description":$("#node-cloud-export-desc").val(), // da assegnare
                          "data":cloudData
                    }
                var cloudDialog = $( this );
var jqxhr = $.post( "https://processloader.snap4city.org/processloader/api/upload.php", JSON.stringify(cloudMessage),  function( data) {
                     console.log( "success:" + data);
                   })
                     .done(function( data ) {
                          data = JSON.parse(data);
                          console.log( "success" + data);
                          if(data.result=="Ok"){
                             cloudDialog.dialog( "close" );
                          }else{
                             $("#node-cloud-export-error-msg").html(""+data.message)
                          }
                     })
                     .fail(function(data) {
                       //data = JSON.parse(data);
                       console.log( "error: " + data);
                     })
                     .always(function(data) {
                       console.log( "finished" + data );
                     });
                }
            },
```

When the user clicks on the top-right menu “Export > Save S4C”, a dialog box appears that enables the user to export the selected flows, the current flow or all flows. In the text box the JSON of the selected flows are reported. When all mandatory fields of the web-form are filled, the user can click on the button “Save S4C”. This action calls the previous function that checks if all mandatory fields are filled out and then collects the values of these fields and registers the flow calling the “https://processloader.snap4city.org/processloader/api/upload.php” API.

For what concern the Import function we integrated the red.js file with the following code:

```
function cloudImportNodes() {
       if (disabled) {
          return;
       }
    dialogContainer.empty();
     dialogContainer.append($(cloudImportNodesDialog));
     dialogContainer.i18n();
    /* dovnload list of resources */
    var jqxhr = $.get( "https://processloader.snap4city.org/processloader/api/download.php?resource_type=IoTApp",   function(
data) {
      console.log( "success:" + data);
    })
      .done(function( data ) {
           data = JSON.parse(data);
           if(data.result=="Ok"){
               if(data.hasOwnProperty('files')){
                   var cloudImportData = data.files;
                   var cloudImportListOptions= "";
                   cloudImportData.forEach((element, index) => {
                       var elName = element.name.substring(0, element.name.indexOf('.json'));
                       var elId =  element.id;
                       cloudImportListOptions+='<li><a href="javascript:void(0)" data-val="'+elId+'">'+elName+'</a></li>';
                   });
               }else if(data.hasOwnProperty('message')){
                    $("#node-cloud-import-error-msg").html(""+data.message);
               }
               $('#cloudImportList').append(cloudImportListOptions);
               $('#cloudImportList a').click(function (evt) {
                      evt.preventDefault();
                    console.log($(this).data('val'));
                    var id= $(this).data('val');
                    // download single resource init
                    var jqxhr = $.get( "https://processloader.snap4city.org/processloader/api/download.php?id="+id,
                                    function( singleResData) {
                      //console.log( "success:" + singleResData);
                      }).done(function( singleResData ) {
                              singleResData = JSON.parse(singleResData);
                              console.log( "success single res" + singleResData);
                              //console.log( "second success" + data.result);
                              //console.log( "second success" + data.message);
                              if(singleResData.result=="Ok"){
                                  if(singleResData.hasOwnProperty('data')){
                                      $('#clipboard-import').val(JSON.stringify(singleResData.data));
                                      $('#clipboard-import').trigger('keyup');
                                  }else if(data.hasOwnProperty('message')){
                                       $("#node-cloud-import-error-msg").html(""+singleResData.message);
                                  }
                              }
                              else{
                                  $("#node-cloud-import-error-msg").html(""+singleResData.message)
                              }
                       })
                       .fail(function(data) {
                      console.log( "error: " + data);
                       })
               })
            // download single resource end
             }else{
                  $("#node-cloud-import-error-msg").html(""+data.message)
           }
       })
      .fail(function(data) {
       console.log( "error: " + data);
      })
        $("#clipboard-dialog-ok").show();
        $("#clipboard-dialog-cancel").show();
        $("#clipboard-dialog-close").hide();
        $("#clipboard-dialog-copy").hide();
        $("#cloud-dialog-copy").hide();
        $("#clipboard-dialog").parent().find(".ui-dialog-titlebar").html("Import s4c");
        $("#clipboard-dialog-ok").button("disable");
        $("#clipboard-import").keyup(validateImport);
        $("#clipboard-import").on('paste',function() { setTimeout(validateImport,10)});
        $("#import-tab > a").click(function(evt) {
            evt.preventDefault();
            if ($(this).hasClass('disabled') || $(this).hasClass('selected')) {
                return;
            }
            $(this).parent().children().removeClass('selected');
            $(this).addClass('selected');
        });
        dialog.dialog("option","title",RED._("clipboard.importNodes")).dialog("open");
    }
```

When the user clicks on the top-right menu “Import > Import S4C”, a box appears with a list of public flows that are available to import. The list is loaded querying the https://processloader.snap4city.org/processloader/api/download.php API.
When a flow is selected the function retrieves it by querying the previous API by passing to it the ID of the flow.

The code we integrated in the core files of NodeRED are marked by using the comments

/* s4c init */

/* s4c end */

Respectively, at the beginning and at the end of the inserted code.