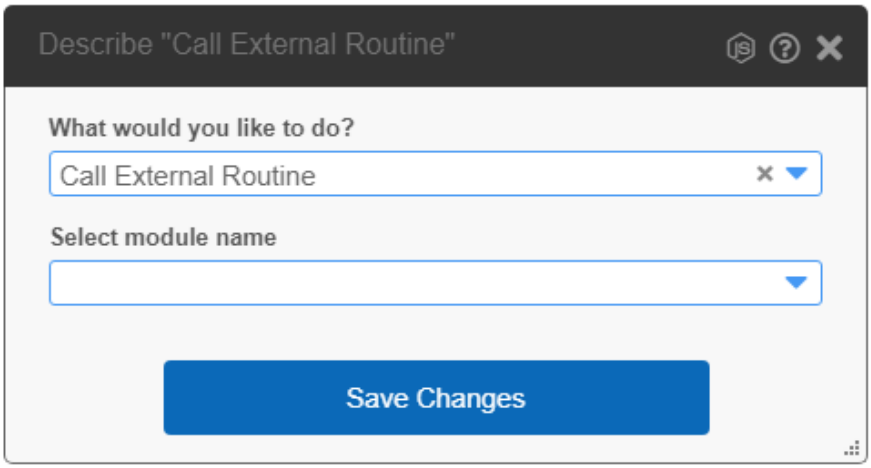
Call Low-code Modules

There are several ways you can call these Low-code Modules routines:

From a Low-code Rich Display, API, or other Low-code Modules:

- Using the Low-code plugin named "Call External Routine", it will walk you though selecting the module, routine as well as setting the input and output parameters.



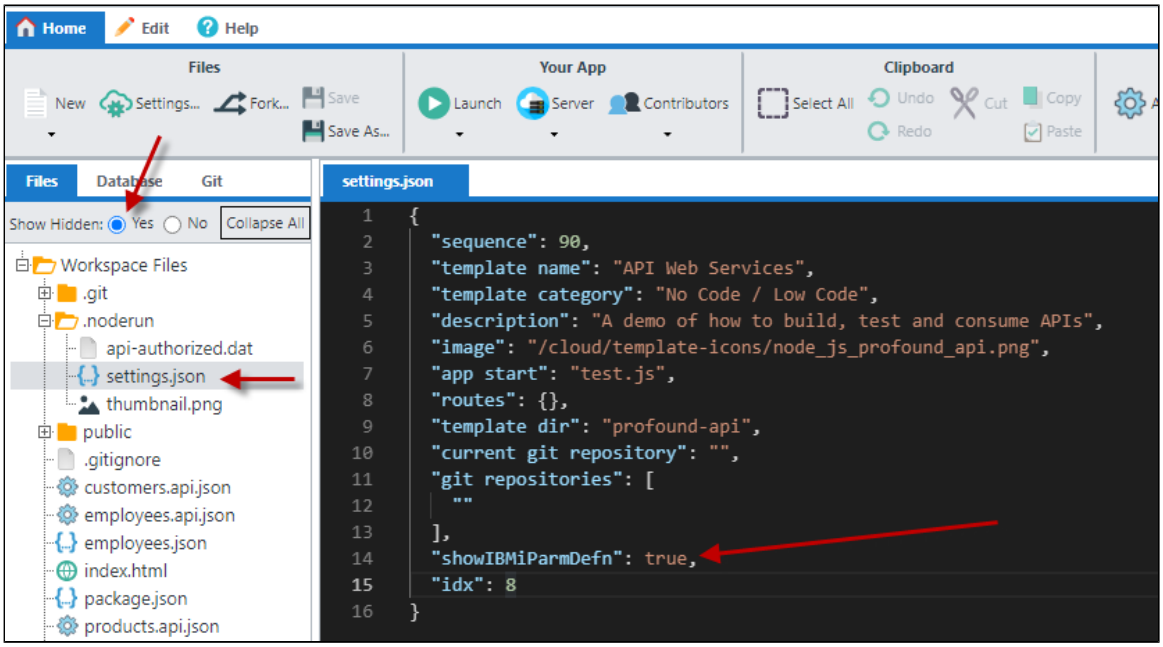
From a javascript file:

```
let input = { postalCode: "90210" };

let weatherModule = pjs.require("Weather.module.js");
let output = weatherModule["Get weather"](input);

console.log("Current weather for 90210 is:", output);
```

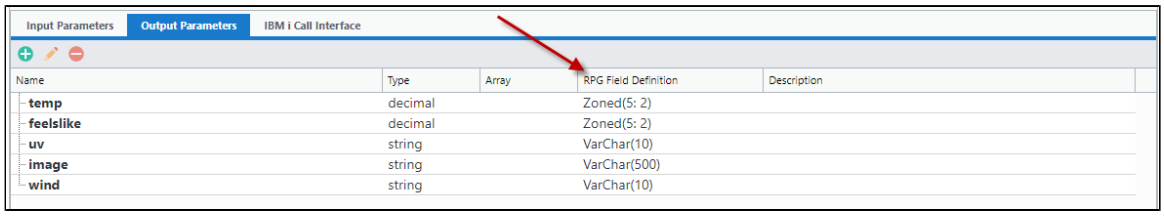
From a RPG program:

To enable Low-code Modules to be called from the IBM i, you must turn on a setting **showIBMiParmDefn** in the configuration.

- For standard Profound.js installations- see here
- For Cloud Profound.js spaces:
  - Edit the hidden settings.json file within the .noderun folder
  - Add a new property named "showIBMiParmDefn" and set the value to true
  - Then reload your browser tab.



Notice, in the **Input** and **Output Parameter** tabs it shows a new column for RPG Field Definition.



If you do not see this column either the **showIBMiParmDefn** flag is not set to true or the Profound.js instance needs to restarted.

Set the **Input** and **Output parameter** definitions to be correct: such as datatype, length, decimal places and array size.

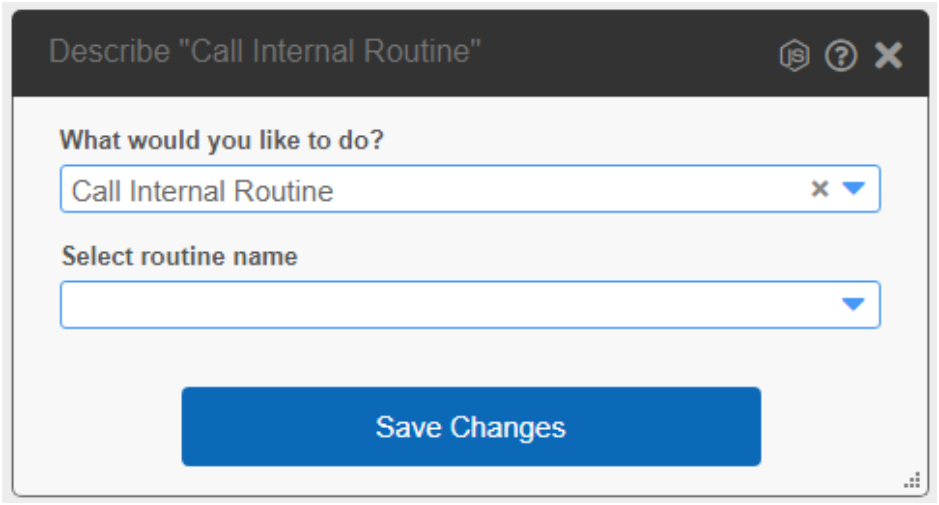
Next, select the **IBM i Call Interface** tab next to the Output Parameters tab.

This is the exact RPG code snippet you can copy and paste right into the IBM Rational IDE.

In your RPG program, all you need to do is set the input parameters (if any defined) and after the call you can access those output parameters (if any defined).

Calling private (Internal) routines:

- Using the Low-code plugin named "Call Internal Routine", it will walk you though selecting the routine as well as setting the input and output parameters.


Describe "Call Internal Routine"
What would you like to do?
Call Internal Routine
Select routine name
Save Changes