Using Workspaces

Full support for workspaces was added in Profound.js 5.0.0.

The ability to create a workspace is *only* available in the Profound.js IDE. This functionality is not available in the Profound UI Visual Designer. You can access the Profound.js IDE using the following URL: http://systemName:port/ide (where '*systemName*' is the name of your system and '*port*' is the port number that your Profound.js instance is installed on).

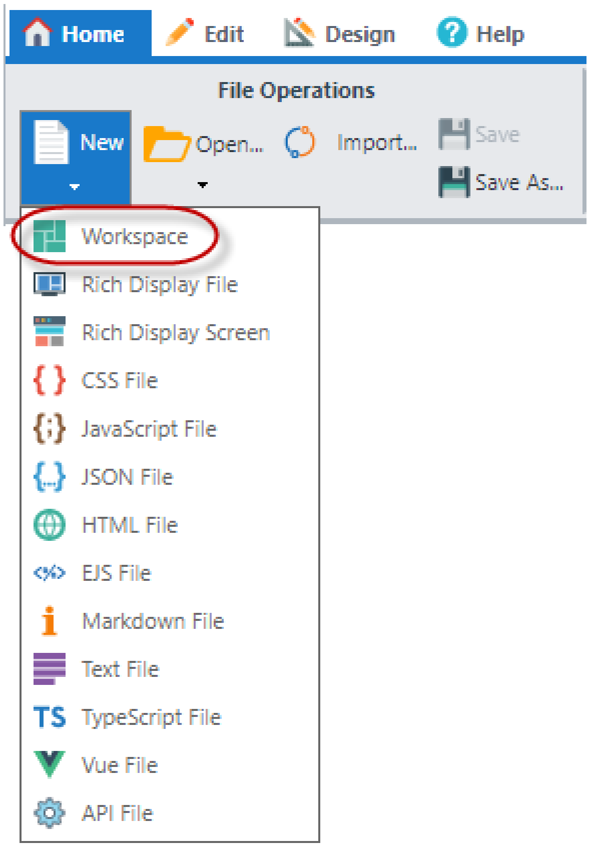
What is a Workspace?

A workspace is a convenient way of organizing your projects. A workspace is a directory on the file system that includes all of the source code and configuration files needed to make an application work, along with a Git repository for source control. Workspace directories are stored within the 'modules' sub-directory of the Profound.js installation directory. Workspaces from Profound.js can easily be deployed in the cloud on Profound.js Spaces and vice versa.

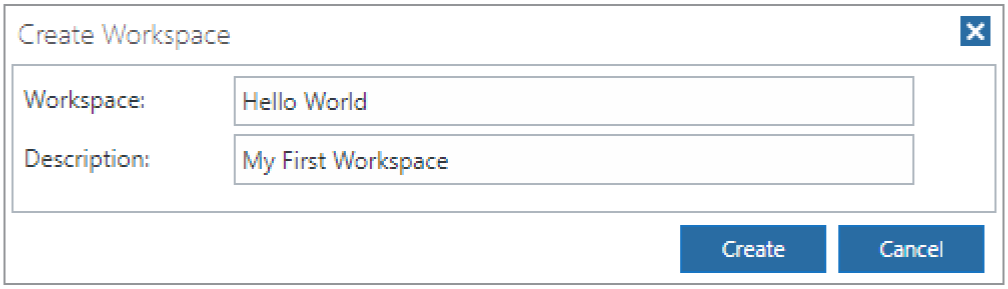
Use of workspaces in Profound.js is optional, but strongly recommended. On Profound.js Spaces, all applications are organized into workspaces.

Creating a Workspace

To create a new workspace, use the option **NewWorkspace** from the **Home** section of the ribbon:



Then enter a workspace name and description on the **Create Workspace** dialog:



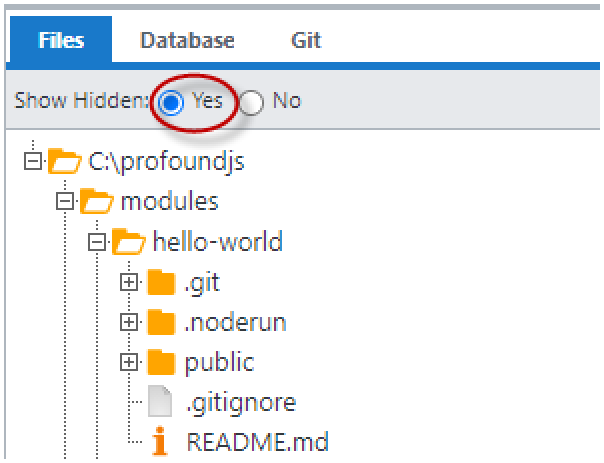
Workspace: Hello World

Description: My First Workspace

The workspace is created and then automatically opened in the IDE.

Workspace Directory Structure

A workspace has the following directory structure. Note that you'll need to select **Show Hidden** in order to show all of the contents.



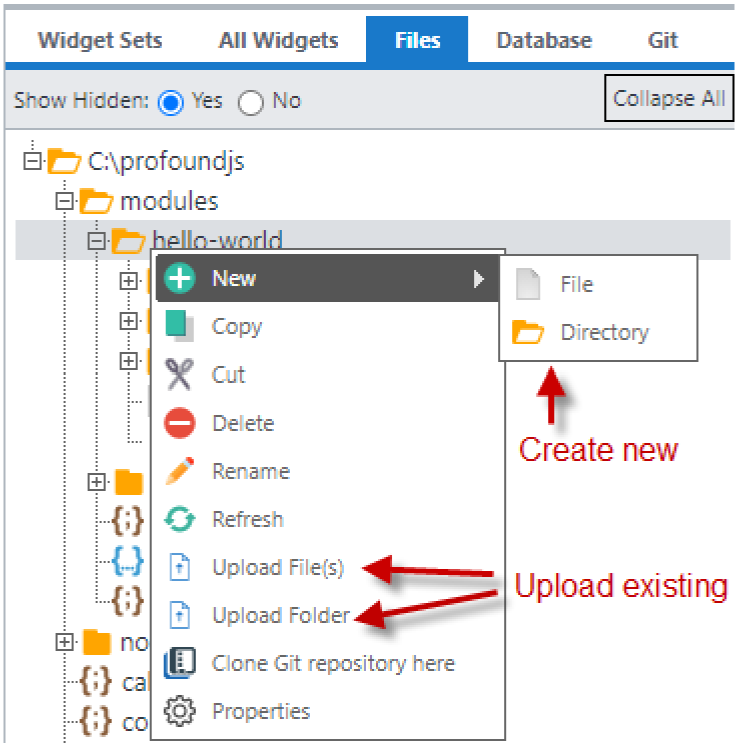
Show Hidden: ● Yes ○ No

```
C:\profoundjs
  modules
    hello-world
      .git
      .noderun
      public
      .gitignore
      README.md
```

- Application source code files can go into the main workspace directory, or into sub-directories that you create. This example shows a new/empty workspace – there are no source code files yet.
- The '.git' directory contains the Git source control repository.
- The '.gitignore' file contains a list of files and directories to be ignored by Git. This is useful when your project contains files that you don't want to commit to source control, such as dependencies from NPM (node_modules), etc.
- The '.noderun' directory contains the workspace settings, which are stored in JSON files.
- The 'public' directory is where you can place static files that are designed to be served via HTTP, such as images, style sheets, client-side JavaScript files, etc.
- The 'README.md' file is a Markdown file that you can use to show a description of your project.

Adding Files and Directories

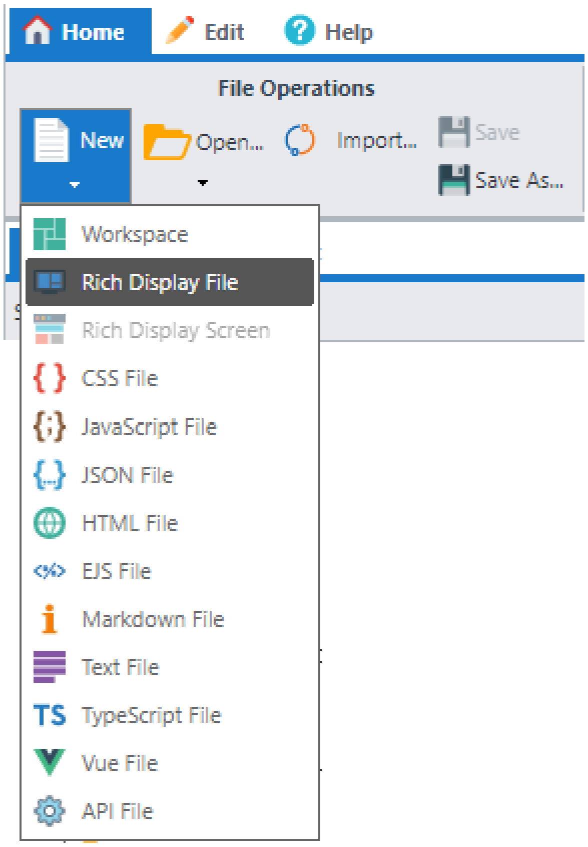
New files and directories can be added to the workspace by using the right-click options on the **Files Tree:**



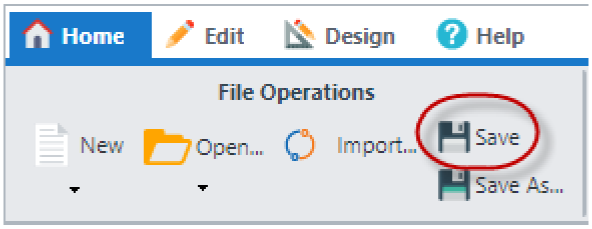
Create new

Upload existing

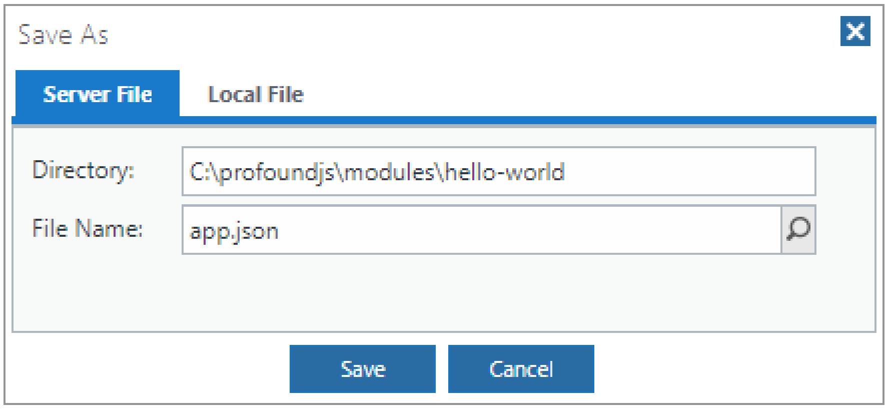
Or by using the **New File** drop down options on the **Home** section of the ribbon:



And then saving the file into the workspace directory or a sub-directory, by clicking the **Save** button in the **Home** section of the ribbon:



When a workspace is open in the IDE, the default save location will be the workspace directory:

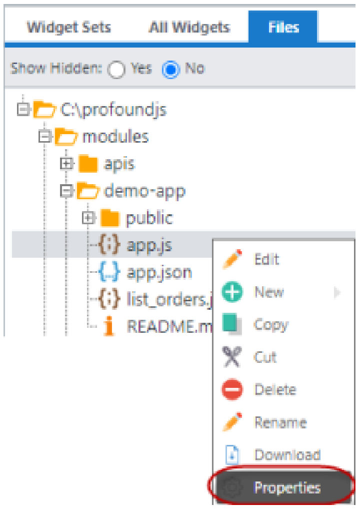


Directory: C:\profoundjs\modules\hello-world

File Name: app.json

Configuring Routes to Run Profound.js Modules

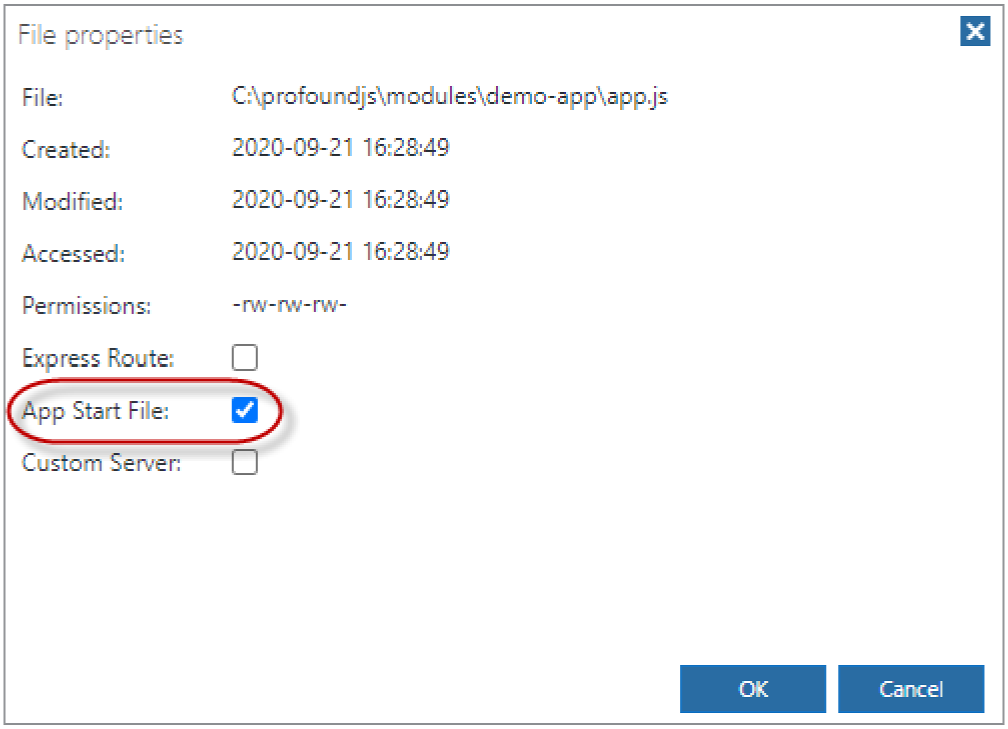
To run an application, you must first create one or more callable Profound.js modules. See here for instructions on creating a callable module. Then, you must configure a route (a URL path) that will be used to call the module. To configure routes, right-click on your Profound.js module file or Rich Display JSON file in the **Files Tree** and select Properties:



Show Hidden: ○ Yes ● No

On the **File Properties** dialog, you can configure two different types of routes:

- An App Start Route. There can be only one App Start Route in each workspace. This route runs a module when the URL specifies just the workspace name. The route can be either a stateful Rich Display program, or a stateless Express program. For a stateful Rich Display program, just click the **App Start** box:



File: C:\profoundjs\modules\demo-app\app.js

Created: 2020-09-21 16:28:49

Modified: 2020-09-21 16:28:49

Accessed: 2020-09-21 16:28:49

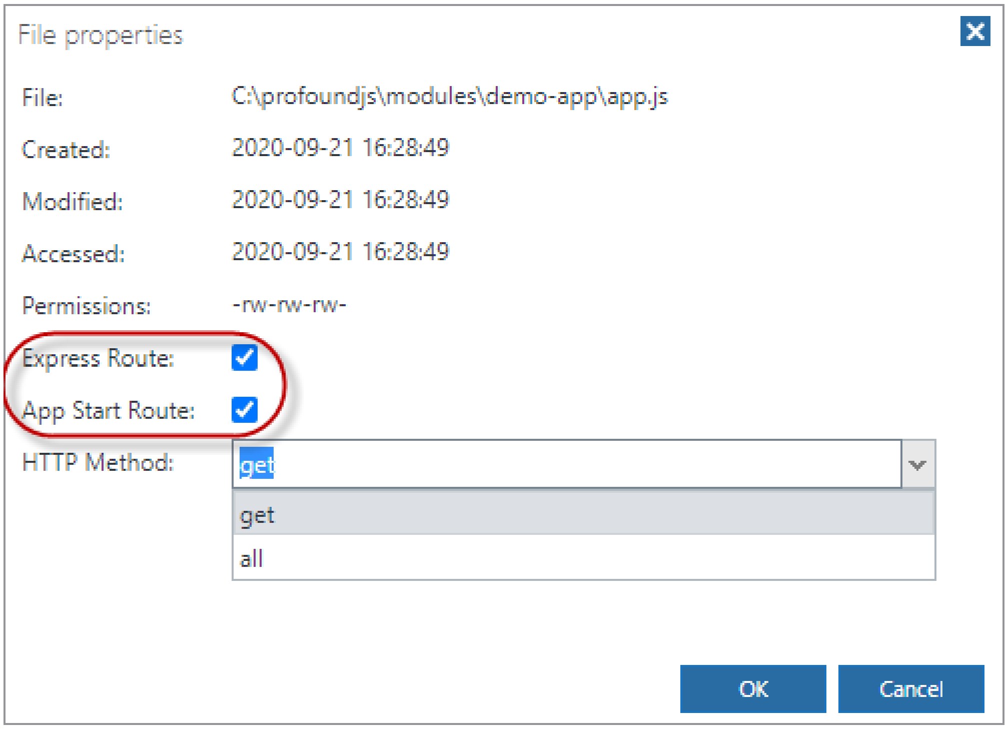
Permissions: -rw-rw-rw-

Express Route: □

App Start File: ☑

Custom Server: □

For a stateless Express program, also check the **Express Route** box and optionally select the desired HTTP method to respond to:



File: C:\profoundjs\modules\demo-app\app.js

Created: 2020-09-21 16:28:49

Modified: 2020-09-21 16:28:49

Accessed: 2020-09-21 16:28:49

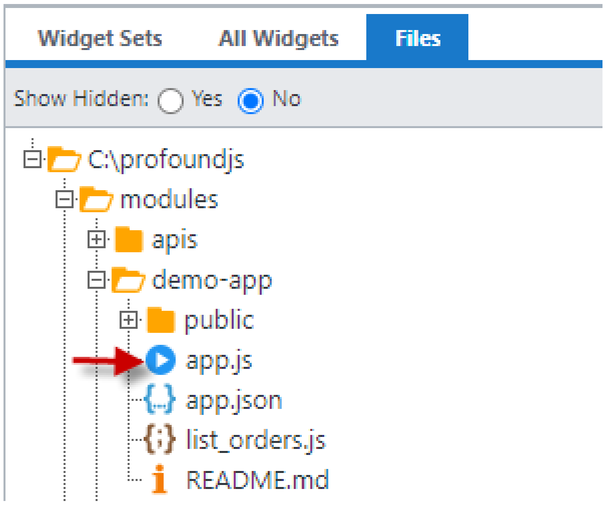
Permissions: -rw-rw-rw-

Express Route: ☑

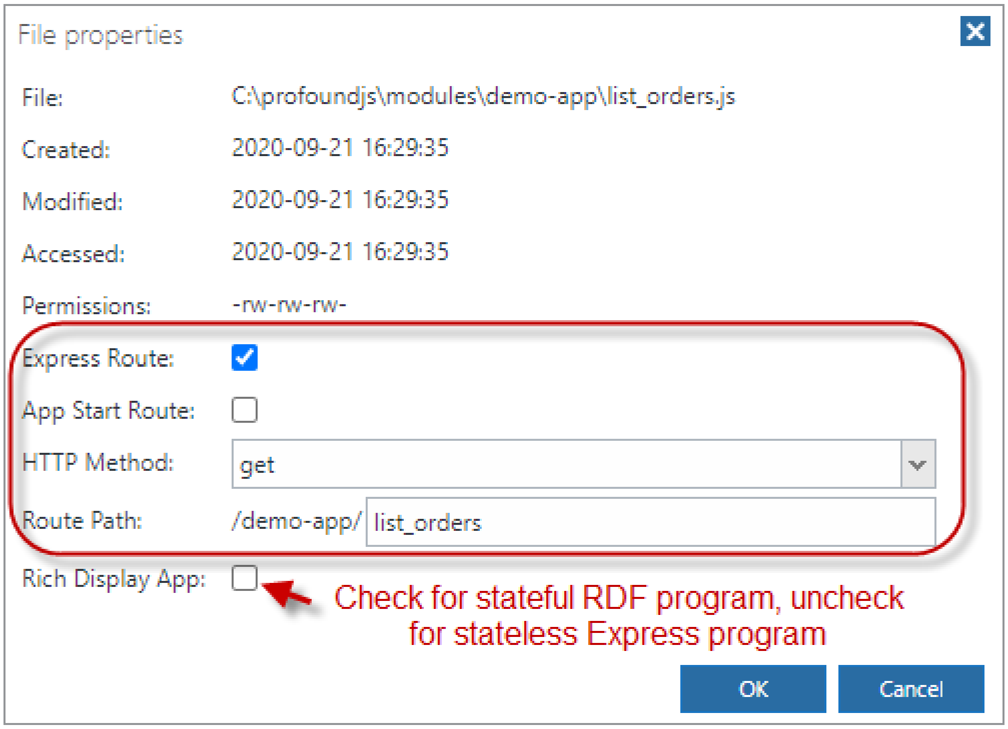
App Start Route: ☑

HTTP Method: get

The **Files Tree** will display an icon next to the file associated with the App Start route, if configured:



- A Named Route. A workspace can contain any number of Named Routes. These routes run when the URL includes a path name, in addition to the workspace name. Named Routes can also be either stateful Rich Display programs, or stateless Express programs. To set up a named route, check the **Express Route** box, uncheck the **App Start Route** box, select the desired HTTP method, and fill in the desired route path. Route paths must be unique within the workspace. For a stateful Rich Display program, set the HTTP Method to 'get' and check the **Rich Display App** box.


File properties

File: C:\profoundjs\modules\demo-app\list_orders.js

Created: 2020-09-21 16:29:35

Modified: 2020-09-21 16:29:35

Accessed: 2020-09-21 16:29:35

Permissions: -rw-rw-rw-

Express Route: ☑

App Start Route: □

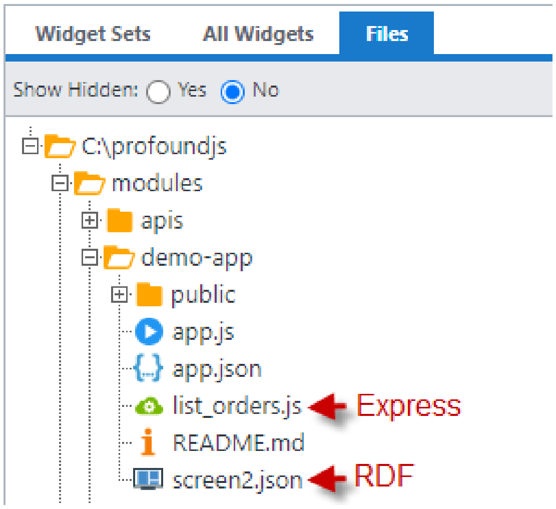
HTTP Method: get

Route Path: /demo-app/ list_orders

Rich Display App: □

Check for stateful RDF program, uncheck for stateless Express program

The **Files Tree** will display an icon next to any files associated with a Named Route:

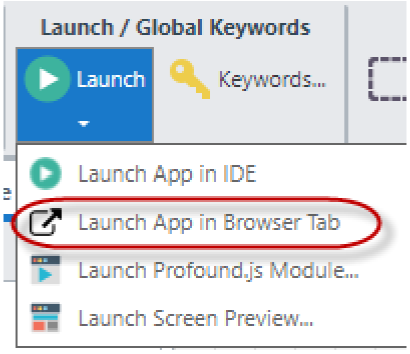


Running Routes

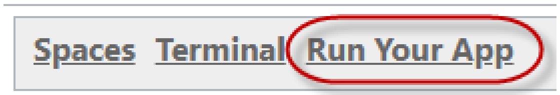
Workspace routes are run using URLs like this:

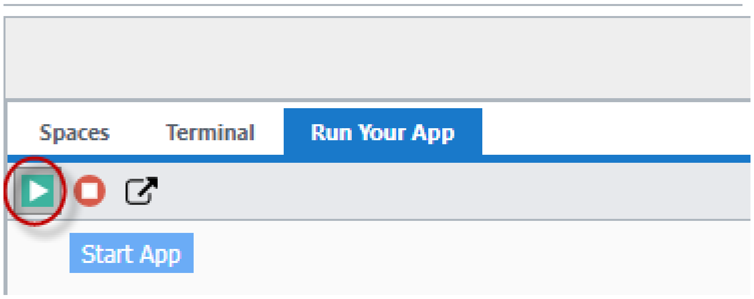
- http://yourServer:port/run/workspace_name/ (App Start Route)
- http://yourServer:port/run/workspace_name/route_path (Named Route)

If an App Start Route is configured, it can be launched in a browser tab by using the option **LaunchLaunch App in Browser Tab** from Home section of the ribbon:



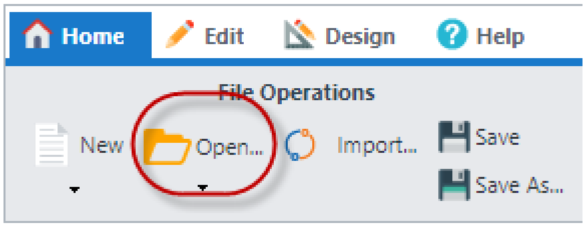
The App Start Route can also be run within the IDE by clicking on **Run Your App** in the bottom bar to expand it, and then using the **Start App** button:



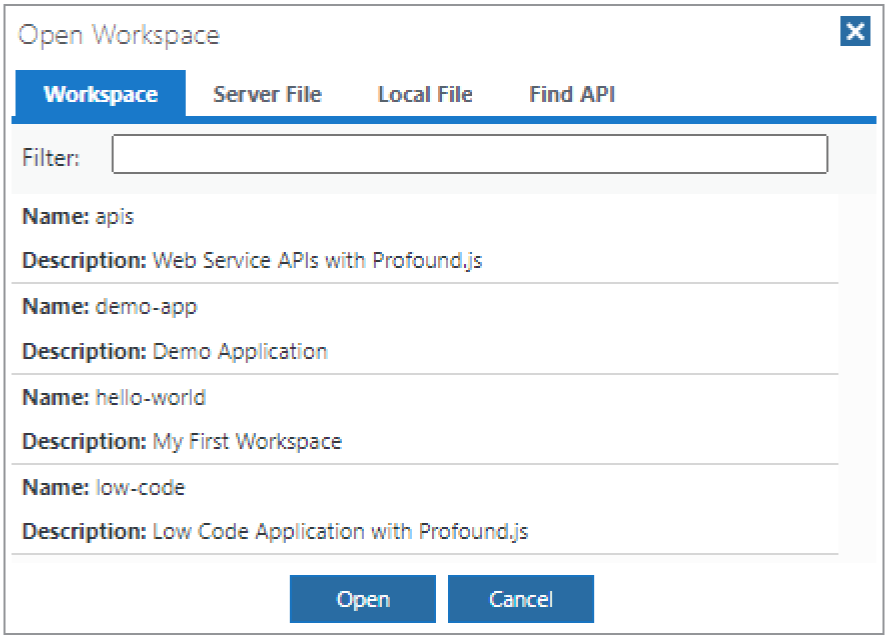


Opening Workspaces

Workspaces can be opened by clicking on the **Open** option on the Home section of the Ribbon:



And then selecting the workspace from the Workspace tab on the **Open Dialog**:



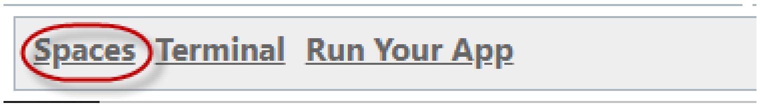
Name: apis
Description: Web Service APIs with Profound.js

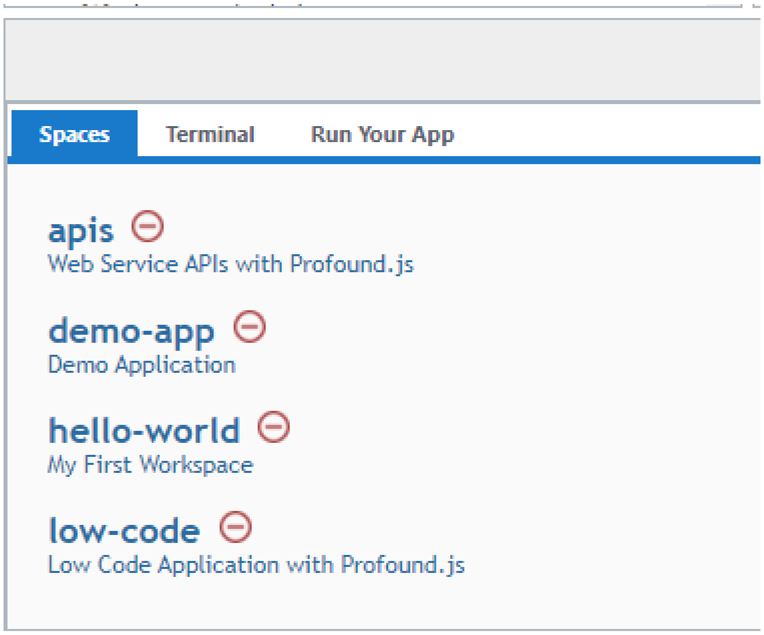
Name: demo-app
Description: Demo Application

Name: hello-world
Description: My First Workspace

Name: low-code
Description: Low Code Application with Profound.js

Workspaces can also be opened using the **Spaces List** in the bottom bar:





apis
Web Service APIs with Profound.js

demo-app
Demo Application

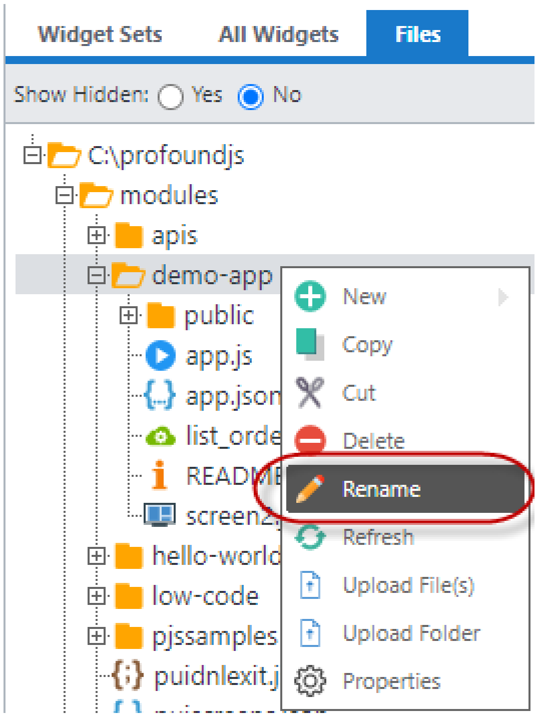
hello-world
My First Workspace

low-code
Low Code Application with Profound.js

Clicking a workspace name in the list will open it in a new browser tab.

Renaming Workspaces

To rename a workspace, use the right-click option in the **Files Tree** to rename the workspace directory:

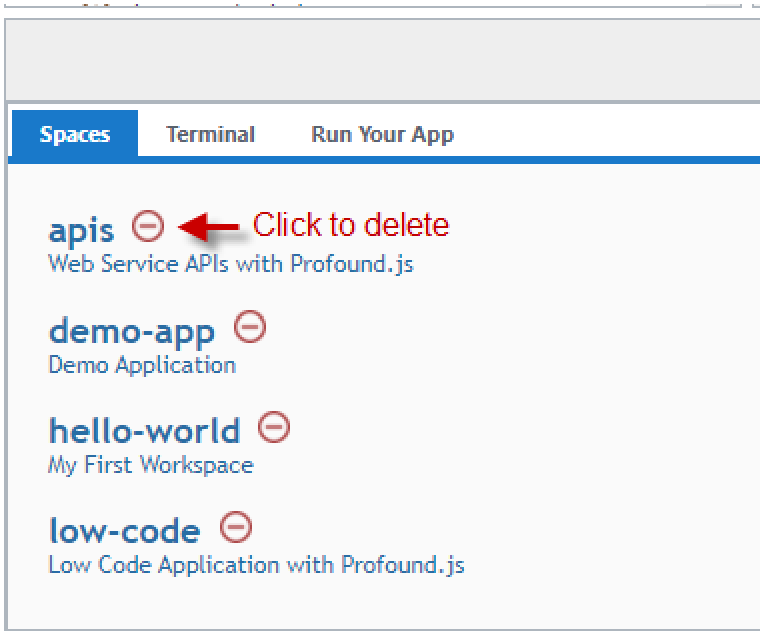


Changing Workspace Descriptions

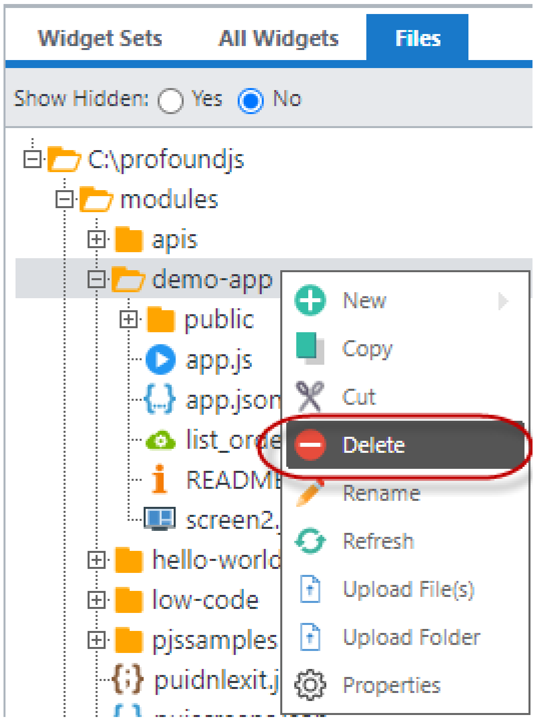
To change a workspace description, open the file '.noderun/settings.json' in the workspace directory, and edit the "description" field.

Removing Workspaces

To remove a workspace, click on the minus icon on the **Spaces List**:



Or use the right-click option in the **Files Tree** to delete the workspace directory:



Customizing HTML Page and Built-in Screens for Rich Display File Apps

Profound.js applications use the Profound UI Rich Display File framework. The following files can be customized in each workspace:

- To customize the Profound UI 'start.html' file, place a copy in the Workspace Directory and name it 'puistart.html'.
- To customize the built-in sign on, error, session timeout, session end, and message box screens, place a copy of the 'puiscreens.json' file in the Workspace Directory.