

Setting up a Git repo for Profound.js Modules on IBM i

Maintaining versions of your code is important in a modern development environment. Git can be installed via a PTF: [5733OPS](#), [option 6](#).

This guide will show you two methods of development with Git:

1. All development is done from the IBM i - single repository over the Profound.js modules folder (non-bare) where developers make individual commits from the same folder.
2. All development is done locally and changes are pulled into the IBM i - remote repository hosted on cloud/local-network server where developers develop on their local machines and make commits to the remote repository. Some cloud services provide private repositories.

Option 1 - Develop on the IBM i

This option is the easiest to setup. Via pase:

1. Change your current directory to your Profound.js modules folder (or one of the subfolders): `cd /profoundjs/modules`
2. Create the non-bare Git repository: `git init`
3. If there are existing files in the folder
 - a. Index and stage all the existing files: `git add --all`
 - b. Commit the staged files to the repository: `git commit -m "YOUR MESSAGE HERE"`

When people are creating, deleting or modifying files, each individual should make commits to the files they change.

Video Tutorial

Option 2 - Develop Locally with a Cloud Repository

This option is best for distributed development. First, you will want to setup either a [local-network repository](#) or use a cloud hosting service such as GitHub or BitBucket.

1. Create a repository on your local-network or on a cloud service
2. Install Profound.js on your local machine
3. Clone your repo within your Profound.js installed on your local machine, with a final parameter of 'modules - which tells the clone to 'clone' into the modules folder: `git clone https://github.com/user/myproductmodules.git modules`
4. Run through Step 1 and Step 2 in the [Profound.js Debugging with VS Code](#) guide to run code locally
5. Make commits back to the repo as normal.

Note: I found that adding "modules" on the git clone command is problematic. Instead first change directory into the modules folder and then run the clone command `git clone https://github.com/user/myproductmodules.git`

This will create the git repository in a sub-folder within modules.

Pulling the changes onto your IBM i

- Clone your repo within your Profound.js installed on your IBM i, with a final parameter of 'modules - which tells the clone to 'clone' into the modules folder: `git clone https://github.com/user/myproductmodules.git modules`
- Use `git pull` every time you'd like to pull new commits down to your IBM i.