

To use submodules to include the `MyFirstFullStackApp-Frontend` and `MyFirstFullStackApp-Backend` repositories as subcomponents within another repository, follow these steps:

1. Navigate to the parent repository.
2. Add the submodules.
  - To add the `MyFirstFullStackApp-Frontend` repository as a submodule, run the following command:

```
git submodule add
https://github.com/diptomistry/MyFirstFullStackApp-Frontend.git frontend
```

- To add the `MyFirstFullStackApp-Backend` repository as a submodule, run the following command:

```
git submodule add
https://github.com/diptomistry/MyFirstFullStackApp-Backend.git backend
```

3. Initialize and update the submodules.
  - Run the following command to initialize and update the submodules:

```
git submodule update --init --recursive
```

4. Commit the changes.
  - Commit the changes to your main repository to include the submodules:

```
git add .
git commit -m "Add frontend and backend submodules"
git push
```

5. Work with the submodules.
  - You can now work with the submodules as if they were separate repositories. Changes made within the submodules can be committed and pushed from within each submodule's directory.
6. Update the submodules.
  - To update the submodules to the latest commits, navigate into each submodule's directory and use `git pull` or other necessary

commands.

Remember:

- When others clone the main repository, they will need to use `git submodule update --init --recursive` to initialize and update the submodules.
- Working with submodules can be a bit of a single repository. Make sure to read up on submodules in the Git documentation to understand their behavior and how to manage them effectively.

I hope this is helpful! Let me know if you have any other questions.