- Home
- Setup FormR Server
  - Vultr Ubuntu VM
  - NodeJS
  - MySQL
- Markdown Cheat Sheets
- Setup Docsify

Hello From Docs/Index.Html

This is a Docsify SPA (Single Page Application) containing Markdown pages describing the building of a VPS (Virtual Private Server) to run our FormR client / server application to quickly and easily edit database tables.

Docsify was discovered by Fillip, and implemented at http://awesome.imade3d.com. He introduced us to writing Markdown pages that are stored in a central repository at GitHub. That turned out t be a great solution for multiple people authoring documentation for IMADE3D printers.

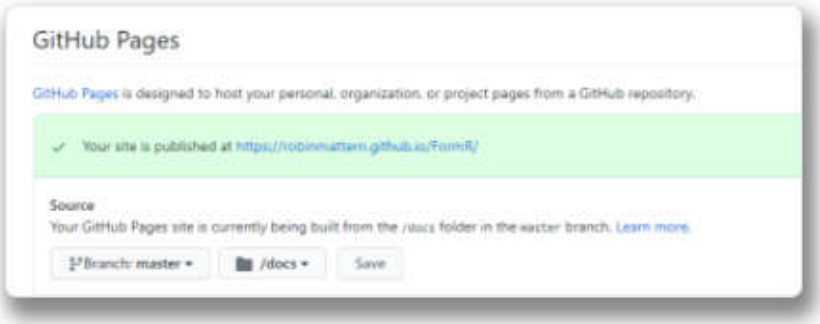
While there is some magic to moving the edited Markdown pages from GitHub to the IMADE3D server, it turns out that

- GitHub can publish the Docsify pages in a webserver available for eacl repository, and
- Docsify can render the documentation from with VSCode while Markdown pages are being edited

All that is needed is to put three initial files into a folder named docs.

```
./docs/index.html -- contains links to the Docsify .css and .js code
./docs/README.md -- the main documentation page
./docs/_sidebar.md -- contains links to all other Markdown files
```

Then after the repository is pushed up to Github, all that is needed is to enable GitHub pages in the `Settings` page for the repository. Note, it ca only be done by the owner of the repository.



And here is what Docsify looks like in GitHub Pages.

- Home
- Setup FormR Server
  - Vultr Ubuntu VM
  - NodeJS
  - MySQL
- Markdown Cheat Sheets
- Setup Docsify

FormR

FormR is a simple easy to use application that uses

- A NodeJS ExpressJS server to receive API calls that retreive and store data to a any database using Sequelize.
- A React-Admin client that provides authenticated users create, retrieve, update and delete (CRUD) records for any database table.

The magic occurs by just providing a connection to a remote or local database, and having the application know how to perform the CRUD operations on the columns for any and all table. FormR also can be customized when the default API actions or column schema needs to be customized.

This example connects to an IODD database server that contains a database containing informaton related to developers, members of the Institute of Database Developers, who can provide expert advise on building business database application.

The documentation herein also provides step by step instruction on how to setup a Virtual Private Server that can connect to a database server and publish the FormR application.

- Home
- Setup FormR Server
  - Vultr Ubuntu VM
  - NodeJS
  - MySQL
- Markdown Cheat Sheets
- Setup Docsify

Setup Instructions For VM With NodeJS On Linux

1. References

See Video: https://www.youtube.com/watch?v=FanoTGjkxhQ&t=1135s
Blog: https://jasonwatmore.com/post/2019/11/18/react-nodejs-on-aws-how-to-deploy-a-mern-stack-app-to-amazon-ec2

2. Login To Linux Server

1. Open Bitvise and Bitvise Profile for Server

2. Login An SFTP window will open for editing and moving files to and from the remote Linux server. An SSH window will also open for executing commands in the Linux console.

A local folder for the VM should exist and be set as the local initial directory under the SFTP tab. This folder should contain the top level folders on the remote server that you will be working with.

3. Prepare Scripts Directory

1. In the Windows File Explorer, copy two linux shell scripts from the `.\FormR\Master\_docs` folder into the local SFTP folder, `..\VMs\et217\home\_0\bin` . The two scripts are

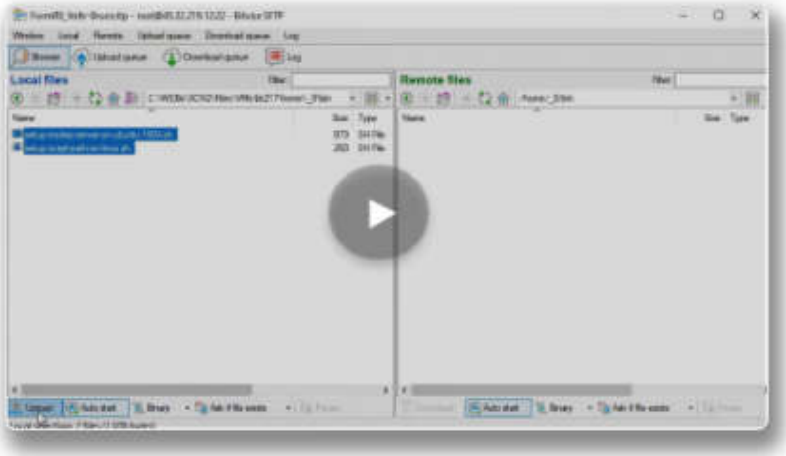
```
setup-nodejs-server-on-ubuntu-1804.sh
setup-script-path-on-linux.sh
```

2. In the remote side of the SFTP window, navigate to the `/home` directory and create a folder, `/home/_0/bin` , for executing scripts in the Linux server.

- This can be done by right clicking anywhere in the `/home` directory and selecting the `Create folder` command and entering `_0` for its name. Then create a folder named, `bin` in the `_0` folder.
- This can also be done in the SSH window by executing the following command in the `/root` directory.

```
$ mkdir /home/_0; mkdir /home/_0/bin
```

3. From the local side of the SFTP window, upload the two scripts from the local `.\home\_0\bin` folder to the remote `/home/_0/bin` folder

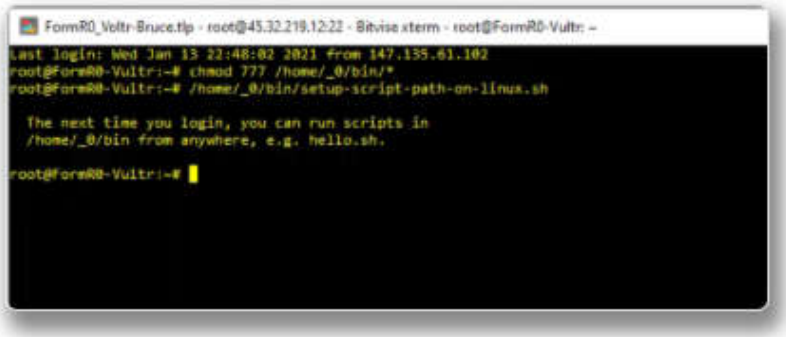


4. To run the scripts you need to make them executable by issuing the `chmod` command in the remote SSH console. You'll then run the script, `setup-script-path-on-linux.sh`, which will create a test script, `hello.sh`, in the `/home/_0/bin` folder and put it all three scripts into the path for all users.

```
$ chmod 777 /home/_0/bin
$ /home/_0/bin/setup-script-path-on-linux.sh
```

5. One final step to prepare the scripts directory. You need to logout of the remote server back in the Bitvise profile window, and login again. Then you should be able to run the test script, hello.sh, from any directory in the remote SSH console:

```
$ hello.sh
```



4. Run The Script To Install NodeJS, NGinx And PM2

1. Execute the script written by Jason Watmor. I have edited out the installation of the MongoDB server program.

```
$ setup-nodejs-server-on-ubuntu-1804.sh
```