mdmon

You can watch a quick demo of mdmon in action here.



Editing markdown in Visual Studio Code is awesome - it can even preview your mardown for you (crt1-k, v). The downside is you can't pop this preview screen on to another monitor!

mdmon allows you to review your rendered markdown in your

browser of choice on your monitor of choice using a familiar GitHub style theme.

It works by using browser-sync in combination with express to

monitor and reload your files when they change. Markdown rendering is performed by marked.

Syntax highlighting is provided with Prism with all languages

Syntax Highlighting

supported.

For example to show some c# code:

```
```csharp
var something = "somevalue";
```
```

To install run the following command:

Install

npm install -g mdmon

'~/somefolderpath' to specify a path.

① localhost:5000/ □ ☆

```
Usage
```

Type $\mbox{ mdmon }$ to run from the current terminal path or $\mbox{ mdmon }$ -p

want to monitor.

 \Box localhost \times + \leftarrow \bigcirc \bigcirc

mdmon

Browsing

Mdmon has simple folder browsing capability so you can point to the

base folder of your project and navigate around to find the file you

```
C:\Temp\some path\Bot Boiler\
.env
.git
.gitignore
.vscode
docs
gulpfile.js
LICENSE
package.json
README.md
src
tests
Powered by mdmon

New Tab

Keep an eye out for the New Tab button at the bottom to open up
```

Create dialogs, use LUIS or QnA maker, call databases and other services, add authentication and a range of other bot functionality in a nice loosely coupled composable

Inject functionality in to your dialogs as required to reduce complexity and increasability.

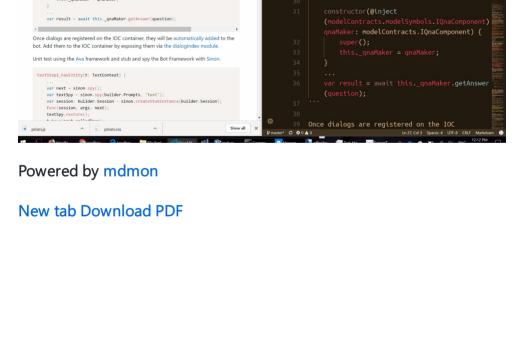
ort default class qnaDialog extends serviceBase imp private _qnaMaker: modelContracts.IQnaComponent; constructor(@inject(modelContracts.modelSymbols.IQ

another browser tab at the root of your selected folder!

x 📭 📭 🛭 0 i

Samples

@injectable()



export default tlass qnaDialog extends serviceBase implements contracts.IDialog{