Thermostat Installation

One way to do this is to try to adjust the ***temperature*** on your old thermostat. If you don’t hear the system turn on within 5 minutes, air does not come out from the vents or the old **thermostat** does not even power on, then you have successfully turned off the `power` to your system. Activites:

- L1
    1. Las
    2. Bay
- L2 Steps:

1. Hello
    - Tomato
        - list 4
    - Potato
2. Hi
3. asdjgk
4. asmnsdnf
5. snmsfnl
6. nadslgldf
7. Dank u wel
8. sfdgjsdfjkfn
9. sdnfjksfn
10. sdfnn
11. sdfnjkndf
12. sdfkl

Participant	Name	City
one	two	three
four	five	six
seven	eight	nine

Note: Happy Weekend

> *Note: Happy Weekend*

```
{
    "status":"OK",
    "data":
        {
            "message": "Welcome, world!"
        }
}
```

Gireesh  Gireesh

H2

H3

H4

H5

H6

The Last Markdown Editor, Ever

Dillinger is a cloud-enabled, mobile-ready, offline-storage compatible, AngularJS-powered HTML5 Markdown editor.

- Type some Markdown on the left
- See HTML in the right
- ✨Magic ✨

Features

- Import a HTML file and watch it magically convert to Markdown
- Drag and drop images (requires your Dropbox account be linked)
- Import and save files from GitHub, Dropbox, Google Drive and One Drive
- Drag and drop markdown and HTML files into Dillinger
- Export documents as Markdown, HTML and PDF

Markdown is a lightweight markup language based on the formatting conventions that people naturally use in email. As John Gruber writes on the Markdown site

> *The overriding design goal for Markdown's formatting syntax is to make it as readable as possible. The idea is that a Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions.*

This text you see here is *actually- written in Markdown! To get a feel for Markdown's syntax, type some text into the left window and watch the results in the right.

Tech

Dillinger uses a number of open source projects to work properly:

- AngularJS - HTML enhanced for web apps!
- Ace Editor - awesome web-based text editor
- markdown-it - Markdown parser done right. Fast and easy to extend.
- Twitter Bootstrap - great UI boilerplate for modern web apps
- node.js - evented I/O for the backend
- Express - fast node.js network app framework @tjholowaychuk
- Gulp - the streaming build system
- Breakdance - HTML to Markdown converter
- jQuery - duh

And of course Dillinger itself is open source with a public repository on GitHub.

Installation

Dillinger requires Node.js v10+ to run.

Install the dependencies and devDependencies and start the server.

```
cd dillinger
npm i
node app
```

For production environments...

```
npm install --production
NODE_ENV=production node app
```

Plugins

Dillinger is currently extended with the following plugins. Instructions on how to use them in your own application are linked below.

Plugin	README
Dropbox	plugins/dropbox/README.md
GitHub	plugins/github/README.md
Google Drive	plugins/googledrive/README.md
OneDrive	plugins/onedrive/README.md
Medium	plugins/medium/README.md
Google Analytics	plugins/googleanalytics/README.md

Development

Want to contribute? Great!

Dillinger uses Gulp + Webpack for fast developing. Make a change in your file and instantaneously see your updates!

Open your favorite Terminal and run these commands.

First Tab:

```
node app
```

Second Tab:

```
gulp watch
```

(optional) Third:

```
karma test
```

Building for source

For production release:

```
gulp build --prod
```

Generating pre-built zip archives for distribution:

```
gulp build dist --prod
```

Docker

Dillinger is very easy to install and deploy in a Docker container.

By default, the Docker will expose port 8080, so change this within the Dockerfile if necessary. When ready, simply use the Dockerfile to build the image.

```
cd dillinger
docker build -t <youruser>/dillinger:${package.json.version} .
```

This will create the dillinger image and pull in the necessary dependencies. Be sure to swap out `${package.json.version}` with the actual version of Dillinger.

Once done, run the Docker image and map the port to whatever you wish on your host. In this example, we simply map port 8000 of the host to port 8080 of the Docker (or whatever port was exposed in the Dockerfile):

```
docker run -d -p 8000:8080 --restart=always --cap-add=SYS_ADMIN --name=dillinger
<youruser>/dillinger:${package.json.version}
```

> *Note: `--capt-add=SYS-ADMIN` is required for PDF rendering.*

Verify the deployment by navigating to your server address in your preferred browser.

```
127.0.0.1:8000
```

License

MIT

Free Software, Hell Yeah!