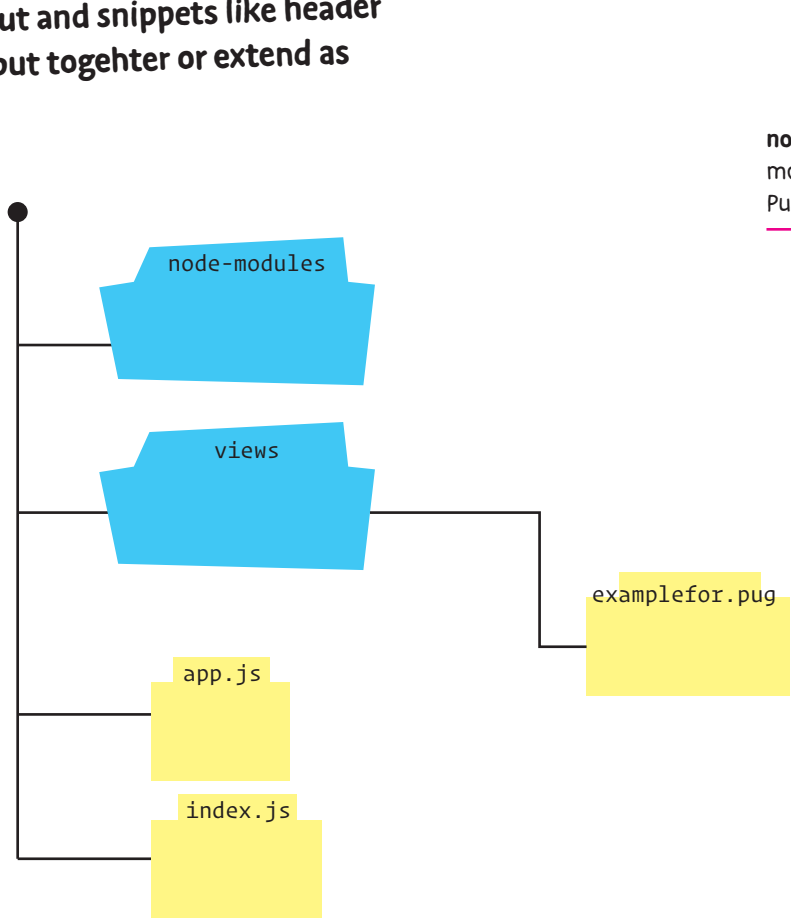


Templating

installing a template engine called pug...

```
npm install pug --save
```

creating html pages via a template engine makes it possible to difine a basic layout and snippets like header that you put togehter or extend as needed



no need to load the pug module.
Pug is loaded app wide!

just don't do it.

```
APP.JS
...
// load all you modules, but not pug
...

//create our app
const app = express();

//define the template folder and the engine
app.set = ('views', './views');
app.set = ('view engine', 'pug');
...
```

create the folder “views” that we set to hold our template files.
Since you are already on it, create a an ‘examplefor.pug’ file as well.

```
examplefor.pug
h1 Hello there!
p this is a paragraph
```

As already explained in **Routing** we now have to set up a route to show and render our file to the viewer.

So wie have to add this to our existing set up.

And our routing file says:

“okay, yea, if you hit our root folder (which is “/”) then...”

“I am going to render smth, and since my render engine is pug, I only need the name of the file. Ah! and that’s examplefor. Good! one second then...”

Then examplefor.pug gets rendered when you visit the root-folder.
Like this...

```
INDEX.JS
...
router.get('/', (req,res) => {
  res.render('examplefor');
});
...
```

```
APP.JS
...
app.use('/', routes);
...
```

forwarded to ourer router file

Existing setup

bam!



But the cool thing is...

we can pass variabels as well!

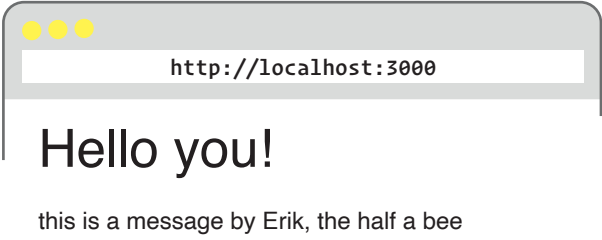


```
INDEX.JS
...
router.get('/', (req,res) => {
  const passedVars = {
    headline : "Hello you!",
    author : "Erik, the half a bee"
  }
  res.render('examplefor', passedVars);
});
...
```

+

```
examplefor.pug
h1= headline
p this is a message by ${author}
```

=



see, there are two ways to use a variable.
h1= works as well as \${author}

Conclusion

pug is a template engine not a dog.