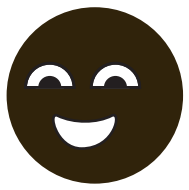
Templating II

There is more we can do!
Wie can also pass params from the adress bar to the template!



INDEX.JS

```
...

router.get('/', (req,res) => {
      const passedVars =
        { headline : "Hello you!",
          cup : req.query.cup
        }
      res.render('examplefor', passedVars);
});

...
```

+

examplefor.pug

```
h1= headline

p In the cup there is #{cup}
```

=

http://localhost:3000?cup=coffee

Hello you!

In the cup there is coffee

Variabels boucing around all over the place!

Basic Layout

You can also create a basic layout that many sites can refer to this is a basic example to start with...

layout.pug

```
doctype html
html
  head
    title= `${title} | ${siteName}`
    link(rel='stylesheet', href='/dist/style.css')
    link(rel="shortcut icon" type="image/png" href="/images/icons/iconimage.png")
    meta(name="viewport" content="width=device-width, initial-scale=1")
  body
    block header
      header
        nav

    block messages

    .content
      block content

    .footer
      block footer
        p © Erik Norden
    block scripts.
```

The line `extends layout` tells our template engine that it shoult take the `layout.pug` a a basis and ...

examplefor.pug

```
extends layout

block content
  h1 Hello there!
  p this is a paragraph
```

... and at **`block content`** line we have our own `block` that should be inserted

since our site is based on **`layout.pug`** and we do have a more information in the **`block footer`** it is displayed on our site...

... and it could be over written!

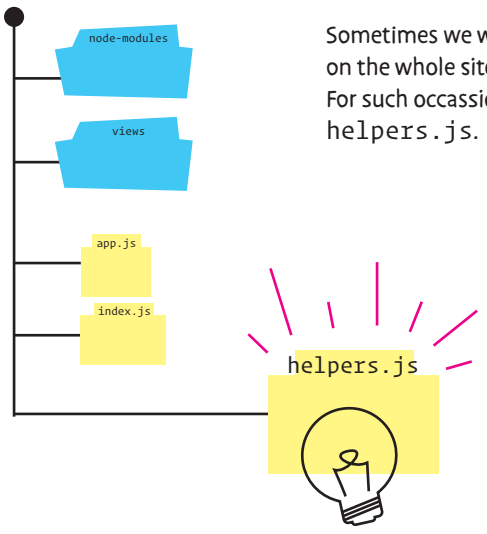
http://localhost:3000

Hello there!

this is a paragraph

© Erik Norden

Sitewide Variables



Sometimes we want to make variables available on the whole site – for example: the site title. For such occassions we create a file called `helpers.js`.

The **idea** of the `helpers.js` is to put there any helper library or any helper data that one needs on the whole site

helpers.js

```
exports.siteName = `our site`;
```

APP.JS

```
...
const helpers = require('./helpers');
...

...
app.use((req,res,next)=>{
  res.locals.h = helpers;
  res.locals.currentPath = req.path;
  next();
});
...
```

Our helpers become available in the **`.h`** of our **`.locals`** of our resonse object

so now we can access the siteName on all of our pages

layout.pug

```
doctype html
html
  head
    title= `${title} | ${h.siteName}`
...
```

examplefor.pug

```
extends layout
...
```

http://localhost:3000

Hello there! | our site

Hello there!

Conclusion

res.locals are librarys, variables and I don't know what, which are available on each response.