

back-end

Templating

lab-2b

Stand-up!

*Show what
you did*

today

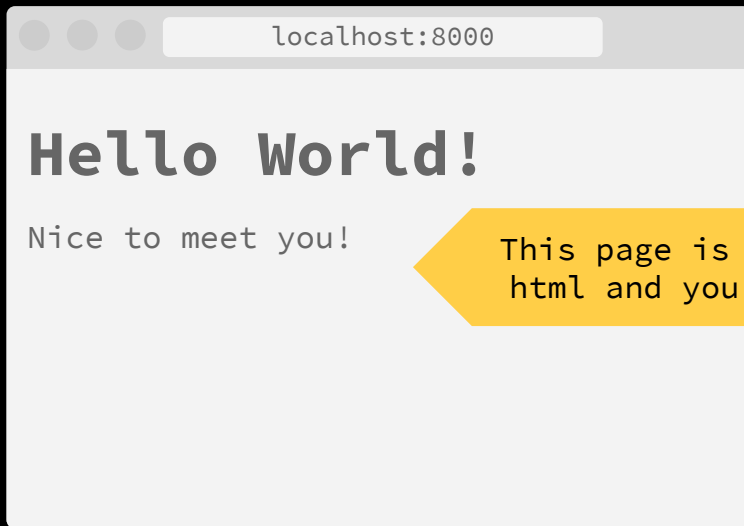
~~I. Stand up~~

II. Templating

Templating

templating

static



This page is static, you type
html and you get the output.

templating

dynamic

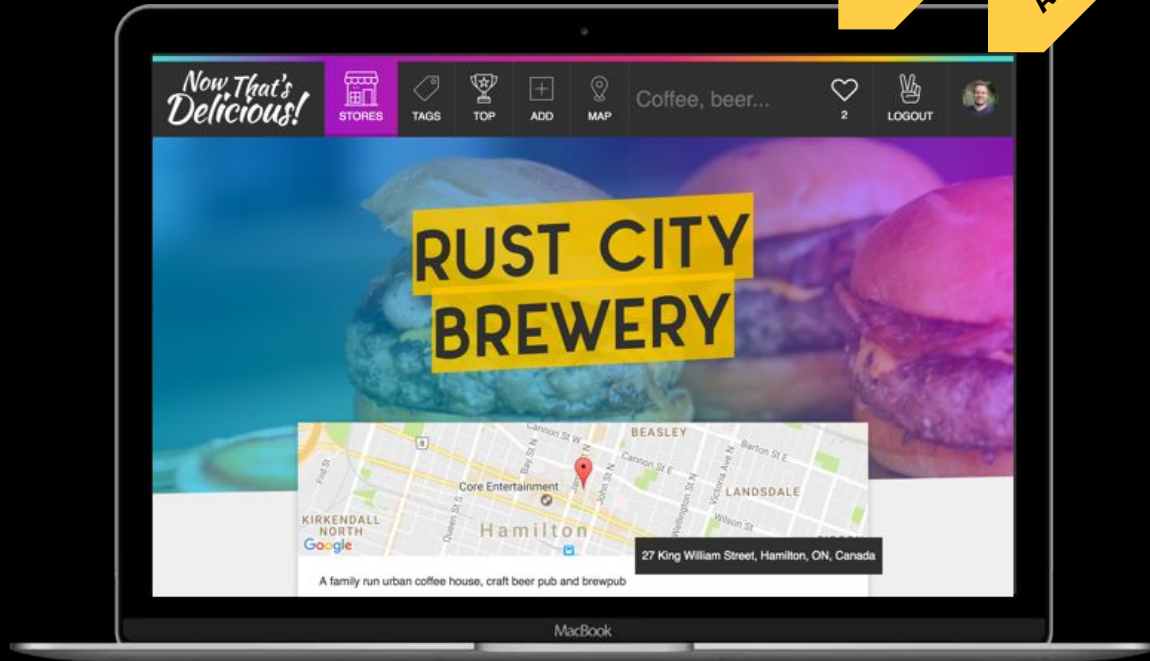


templating

Likes

Avatar

dynamic



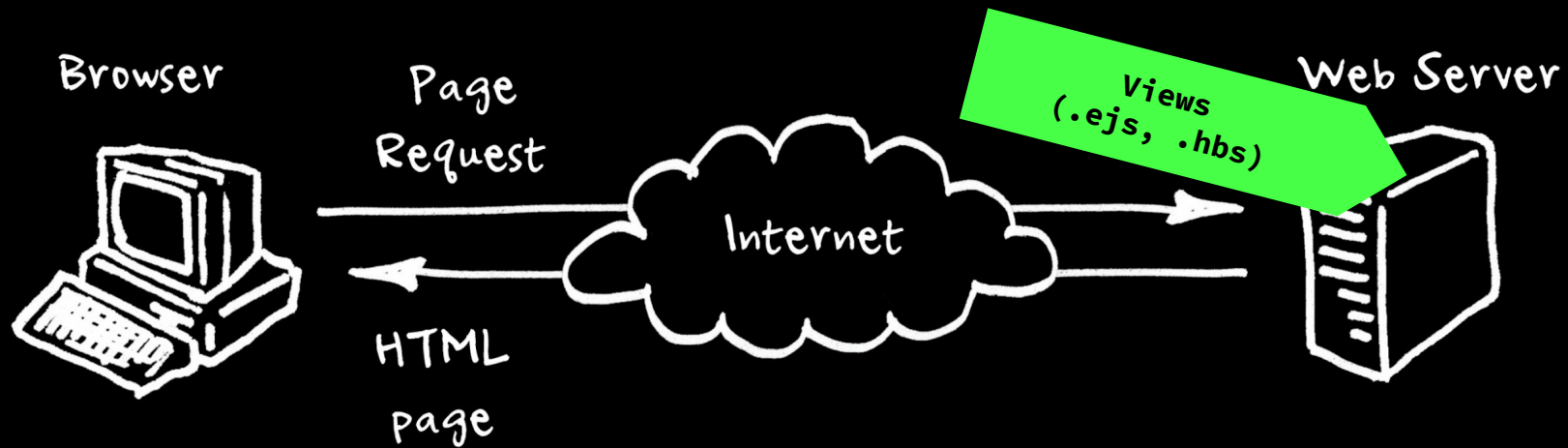
templating

?

A template engine enables you to use static template files in your application. [...] the template engine replaces variables in a **template file with actual values, and transforms the template into an HTML file sent to the client.**

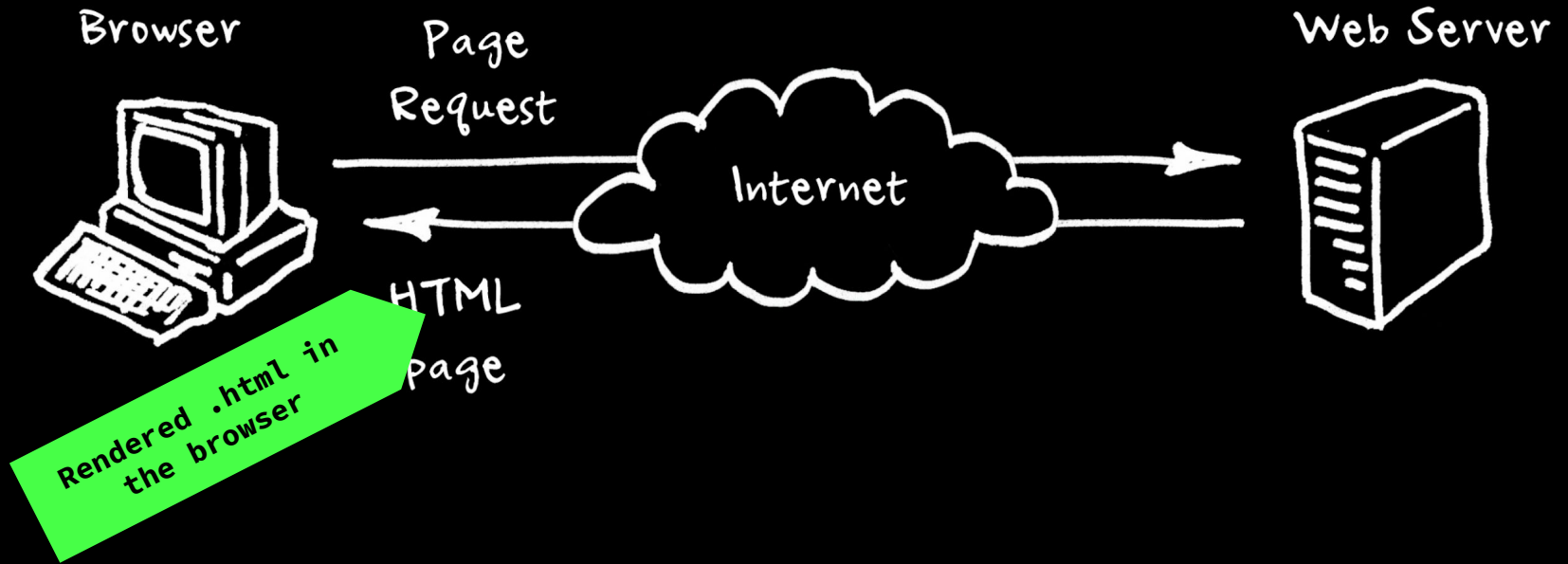
Express

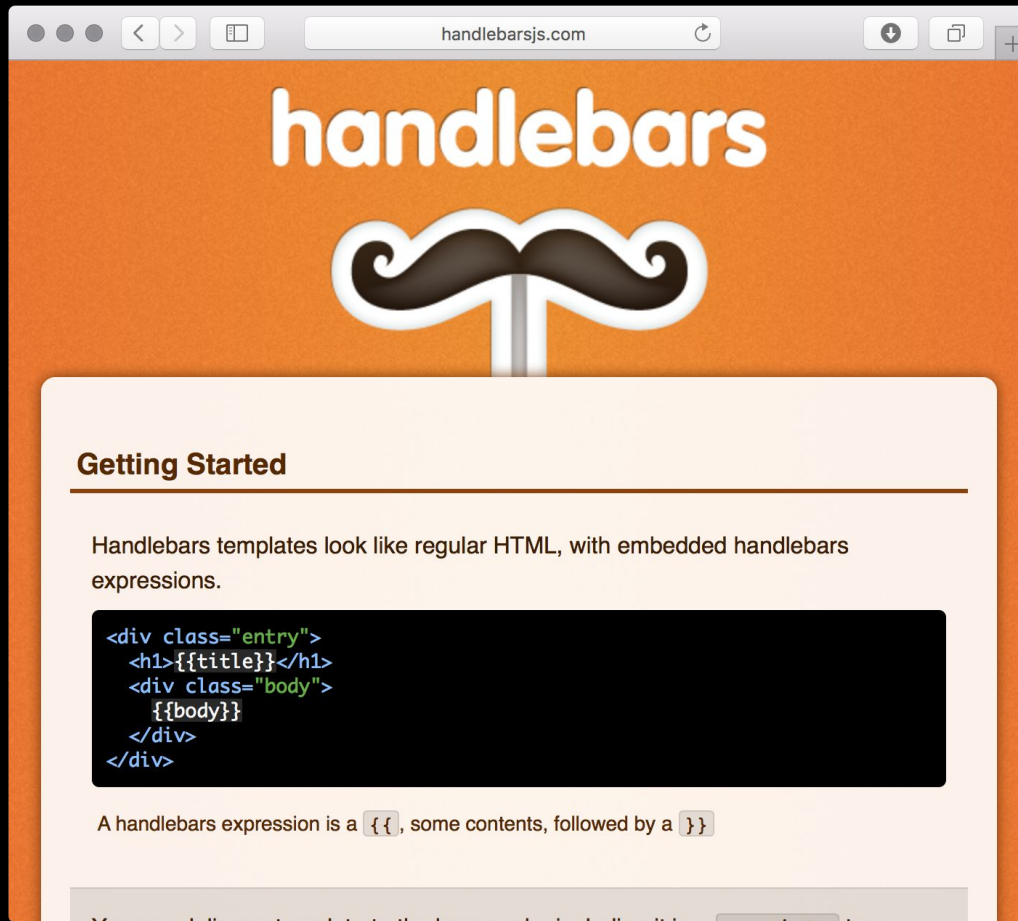
expressjs.com





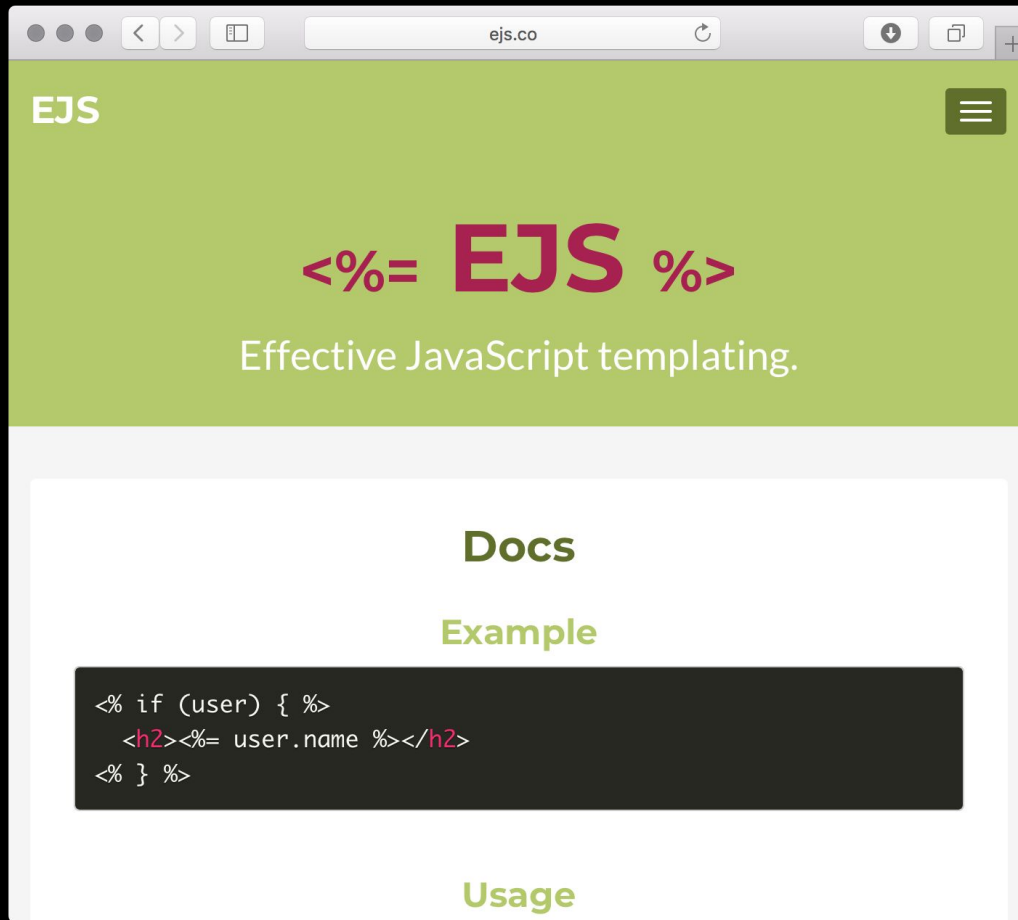
Note: instead of storing .html files you store 'views' (templating files)





handlebarsjs.com

Express



Express

ejs.co

templating

views

```
// Files
express-server/
├─ node_modules/
├─ static/
│   └─ index.css
├─ view/
│   ├── detail.ejs
│   ├── list.ejs
│   └─ not-found.ejs
├─ index.js
└─ package.json
```

```
bash
$ npm install ejs

+ ejs@2.5.7
added 1 package in 0.811s

$
```

Express

templating

```
view/not-found.ejs
<!doctype html>
<link
  rel=stylesheet
  href=/index.css
>
<title>Not found - My movie
website</title>
<h1>Not found</h1>
<p>Uh oh! We couldn't find this
page!</p>
```

```
view/not-found.ejs
<!doctype html>
<link
  rel=stylesheet
  href=/index.css
>
<title><%= data.title %> - My movie
website</title>
<h1><%= data.title %></h1>
<p><%= data.description %></p>
<p><a href="/">Back to list</a></p>
```

<%= is JavaScript with output

%> stops JavaScript

templating

partials

```
// Files
express-server/
├─ node_modules/
├─ static/
│   └─ index.css
├─ view/
│   ├── detail.ejs
│   └─ head.ejs
│   ├── list.ejs
│   └─ not-found.ejs
├─ index.js
└─ package.json
```

Express

templating

partials

```
view/head.ejs
<!doctype html>
<link
  rel=stylesheet
  href=/index.css
>
```

```
view/not-found.ejs
<% include head.ejs %>
<title>Not found - My movie
website</title>
<h1>Not found</h1>
<p>Uh oh! We couldn't find this
page!</p>
```

Express

index.js

```
...

express()
  .use(express.static('static'))
  .set('view engine', 'ejs')
  .set('views', 'view')
  ...

function movies(req, res) {
  res.render('list.ejs', {data: data})
}

function movie(req, res, next) {
  ...
  res.render('detail.ejs', {data: movie})
}

function notFound(req, res) {
  res.status(404).render('not-found.ejs')
}
```

Express

```
index.js

...

express()
  .use(express.static('static'))
  .set('view engine', 'ejs')
  .set('views', 'view')
  ...

function movies(req, res) {
  res.render('list.ejs', {data: data})
}

function movie(req, res, next) {
  ...
  res.render('detail.ejs', {data: movie})
}
```

Note: Views are **rendered** (combined with data) on the server.
The resulting HTML is sent to the client.



templating

Learn how to use a templating engine to dynamically render data and create components for your matching application.

Synopsis

- **Time:** 6:00h
- **Goals:** subgoal 4
- **Due:** before week 3

Description

We are slowly going to build the interface and components for your matching application. You already have a server up and running, now it's time to actually send dynamic HTML to the client using a templating engine.

1. Research different templating engines and read their documentation such as [pug](#), [ejs](#) or [handlebars](#). Document your research in your wiki. Pick one and install it in your project.
2. Then, create views and try to render a page using the templating engine. Start with simple HTML pages and make sure you get

work on **templating**



exit;

see you in lab-3a!