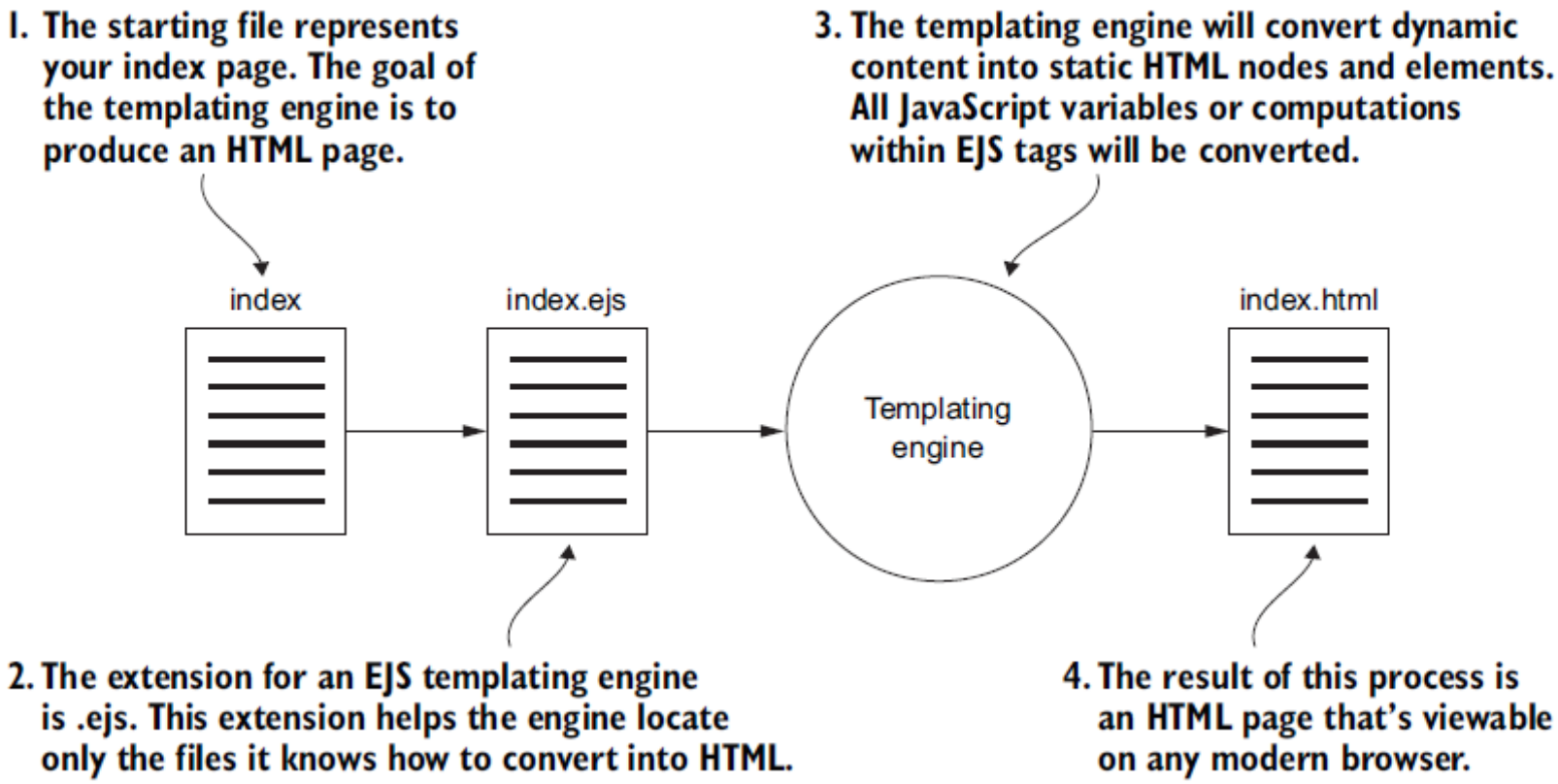
EJS

A template engine for Express

Templating

- You can extend Express to use a templating engine that allows you to insert dynamic data in the views.
- EJS is a template engine that proves to be the simplest templating language to learn for people with moderate experience with HTML.

Converting EJS to HTML



Configuring an Express App with ejs

- Just enter express at the command line
  - You need to have express-generator installed globally for this to work

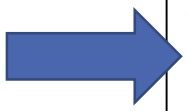
To use ejs and init a git repo:

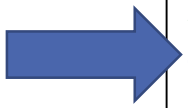
```
express --view=ejs --git app_name
```

- For an existing express app you can install ejs with npm

```
npm i ejs -S
```

app.js





```
var createError = require('http-errors');
var express = require('express');
. . .
var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
var app = express();
// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');
// middleware setup
app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
. . .
// routes setup
app.use('/', indexRouter);
app.use('/users', usersRouter);
// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
// set locals, only providing error in development
. . .
module.exports = app;
```

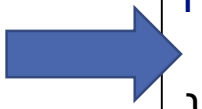
Respond with a view

index.js

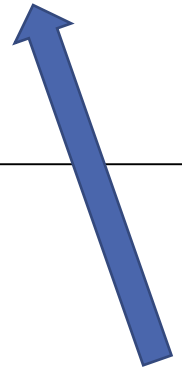
```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});

module.exports = router;
```



Name of view file

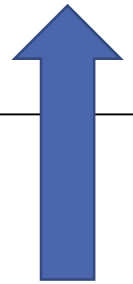


Object with data to pass to the view

The view

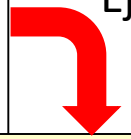
index.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1><%= title %></h1>
    <p>Welcome to <%= title %></p>
  </body>
</html>
```



The value of the title property will be inserted here

Ejs view engine



index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Express</title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1>Express</h1>
    <p>Welcome to Express</p>
  </body>
</html>
```

Ejs layouts

- A *layout* is a template in which your views are rendered
- To use this feature with ejs you must install the express-ejs-layouts package

```
npm i express-ejs-layouts -S
```

- And require and use in app.js:

```
const layouts = require("express-ejs-layouts")
app.use( layouts );
```

Ejs layout template file

layout.ejs

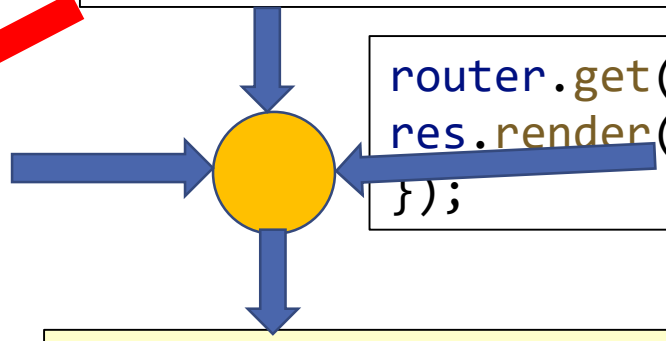
```
<!DOCTYPE html>
<html>
<head>
  <title><%= title %></title>
  <link rel='stylesheet'
   href='/stylesheets/style.css' />
</head>
<body>
  <div id="nav">NAVIGATION</div>
  <%- body %>
  <div id="footer">FOOTER</div>
</body>
</html>
```

index.ejs

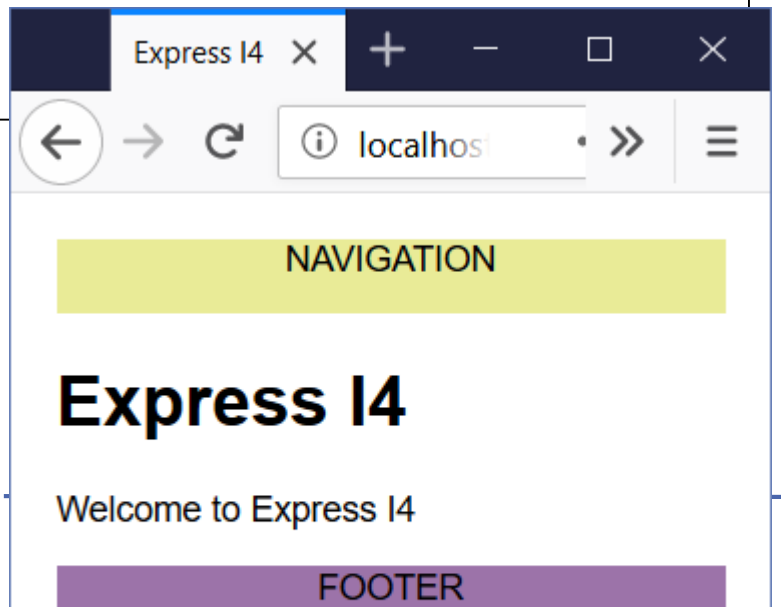
```
<h1><%= title %></h1>
<p>Welcome to <%= title %></p>
```

index.js

```
router.get('/', function(req, res, next)
res.render('index', { title: 'Express' })
});
```

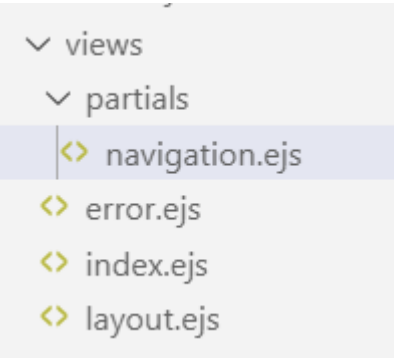


```
<!DOCTYPE html>
<html>
 <head>
  <title>Express</title>
  <link rel='stylesheet' href='/stylesheets/style.css' />
 </head>
 <body>
  <div id="nav">NAVIGATION</div>
  <h1>Express</h1>
  <p>Welcome to Express</p>
  <div id="footer">FOOTER</div>
 </body>
</html>
```



Partials

- Are snippets of view content that can be included in other views.
- You add a partial view to another EJS view by using the **include** keyword and the path to the partial view
- And it is recommended to keep you partial views in a separate sub folder of the View folder – often called partials




```
<body>
<div id="nav"><% include partials/navigation %></div>
<%- body %>
<div id="footer">FOOTER</div>
</body>
```

navigation.ejs

```
<h3>Navigation area</h3>
```

Loops

```
exports.someStudents = (req, res) => {
res.render('shortList', {
title: '3 students',
students
});
};
```

```
let students = [{
  studentNo: '1',
  givenName: 'Brendan',
  sirName: 'Eich'
  }, …
```

```
exports.listStudents = (req, res) => {
  res.render('listStudents', {
  title: 'Students List',
  students
  });
};
```

```
<h1>3 first students in ITTWEB</h1
<table>
<% for(let i=0; i < 3; i++) { %>
  <tr>
    <td><%= students[i].studentNo %></td>
    <td><%= students[i].givenName %></td>
    <td><%= students[i].sirName %></td>
  </tr>
<% } %>
</table>
```

```
<h1>Students in ITTWEB</h1>
<% students.forEach(student => { %>
  <strong> <%= student.studentNo %>
  </strong>
  <span> <%= student.givenName %>
  <%= student.sirName %></span>
  <br />
<% }); %>
```

Conditionals

```
<h1>Students in ITTWEB</h1>
<% students.forEach(student => { %>
  <%= student.studentNo %>
  <%= student.givenName %> <% if (student.sirName == 'Eich') { %> <strong> <% } %>
  <%= student.sirName %> <% if (student.sirName == 'Eich') { %> </strong> <% } %>
<br />
<% }); %>
```

References &Links

- Get programming with Node.js chapter 10.
- https://ejs.co/
- YouType videos – search for ”intro ejs”