back-end

Forms & Connect

lab-3b

Stand-up!

Show what you did

today

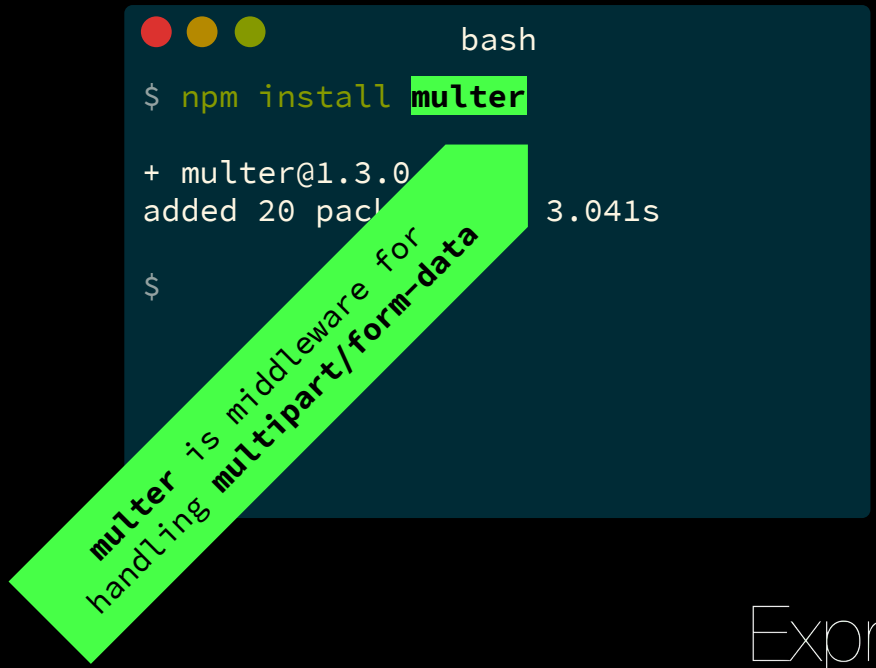
I. ~~Stand up~~
II. Files
III. Connect

Files

files

multer

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



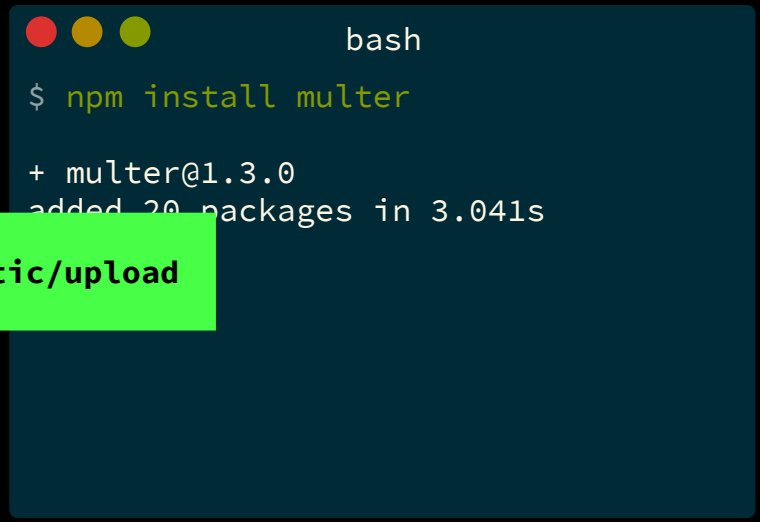
Express

files

folder

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

We'll upload files to **static/upload**



Express

static/index.css

```
…

img {
  max-width: 100%;
  max-height: 15em
}
```

view/detail.ejs

```
…
<h1><%= data.title %></h1>
<% if (data.cover) { %>
<img
  alt
  src="/upload/<%= data.cover %>"
>
<% } %>
<p><%= data.description %></p>
…
```

Express

view/add.ejs

```
<% include head.ejs %>
<title>Add a movie - My movie website</title>
<h1>Add a new movie</h1>
<form
  action=/
  method=post
  enctype=multipart/form-data
>
  <label>Title <input name=title></label>
  <label>
    Cover
    <input name=cover type=file accept=image/*>
  </label>
  <label>
    Plot (short)
    <input name=plot>
  </label>
  …
  <button>Add</button>
</form>
<% include tail.ejs %>
```

Needed for **input[type=file]**

Express

view/add.ejs

```
<% include head.ejs %>
<title>Add a movie - My movie website</title>
<h1>Add a new movie</h1>
<form
  action=/
  method=post
  enctype=multipart/form-data
>
  <label>Title <input name=title></label>
  <label>
    Cover
    <input name=cover type=file accept=image/*>
  </label>
  <label>
    Plot (short)
    <input name=plot>
  </label>
  …
  <button>Add</button>
</form>
<% include tail.ejs %>
```

Accept only images

Express

view/add.ejs

```
<% include head.ejs %>
<title>Add a movie - My movie website</title>
<h1>Add a new movie</h1>
<form
  action=/
  method=post
  enctype=multipart/form-data
>
  <label>Title <input name=title></label>
  <label>
    Cover
    <input name=cover type=file accept=image/*>
  </label>
  <label>
    Plot (short)
    <input name=plot>
  </label>
  …
  <button>Add</button>
</form>
<% include tail.ejs %>
```

Accept only images

Express

index.ejs

```
…
var multer = require('multer')

…

var upload = multer({dest: 'static/upload/'})

express()
  …
  .post('/', upload.single('cover'), add)
  …

function add(req, res) {
  …
  data.push({
    …
    cover: req.file ? req.file
    …
  })
}

…
```

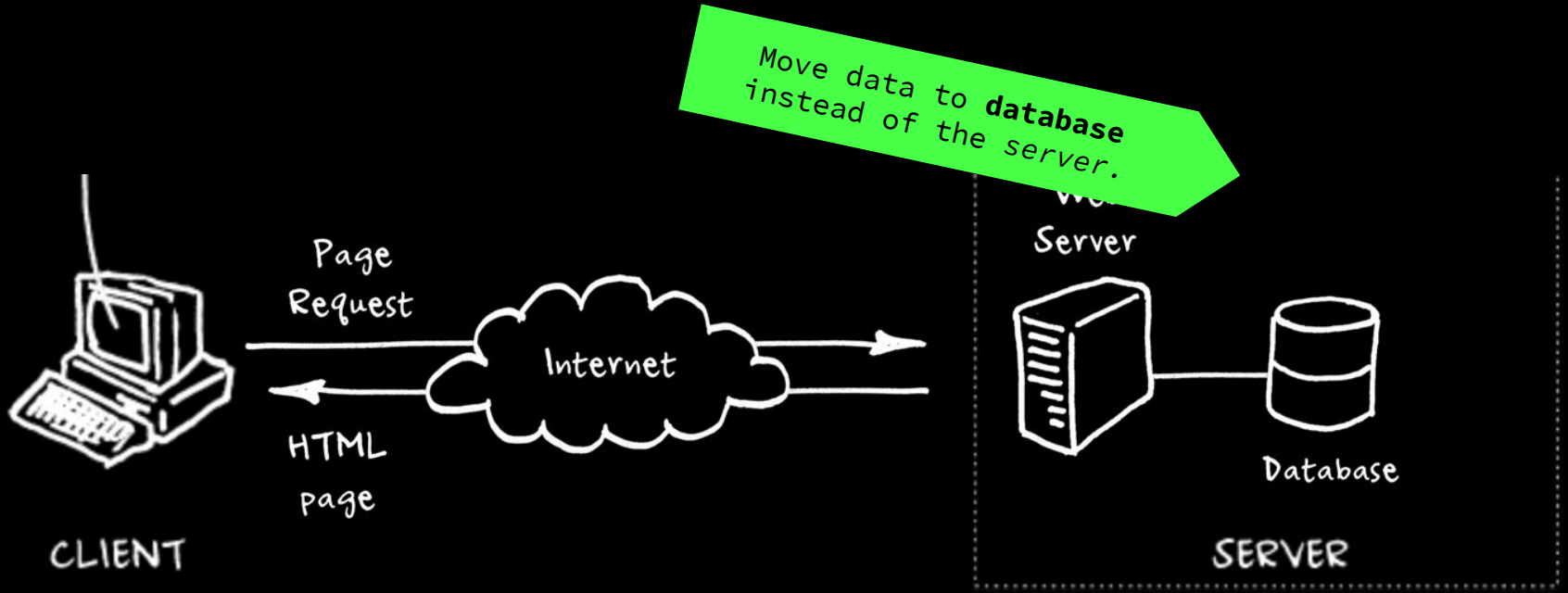
multer sets **req.file**

Express


localhost:8000/add
Add a new movie
Wonder Woman
Diana, an Amazonian warrior…
Cover
wonder-woman.jpg
When a pilot crashes and tells of conflict in the outside world, Diana, an Amazonian warrior in training, leaves home …
We can add files!

Express

Connect


Move data to **database** instead of the server.
Page Request
HTML page
Internet
Server
Database
CLIENT
SERVER

connect

mongodb

MongoDB (from hu**mong**ous) is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is developed by MongoDB Inc. […]

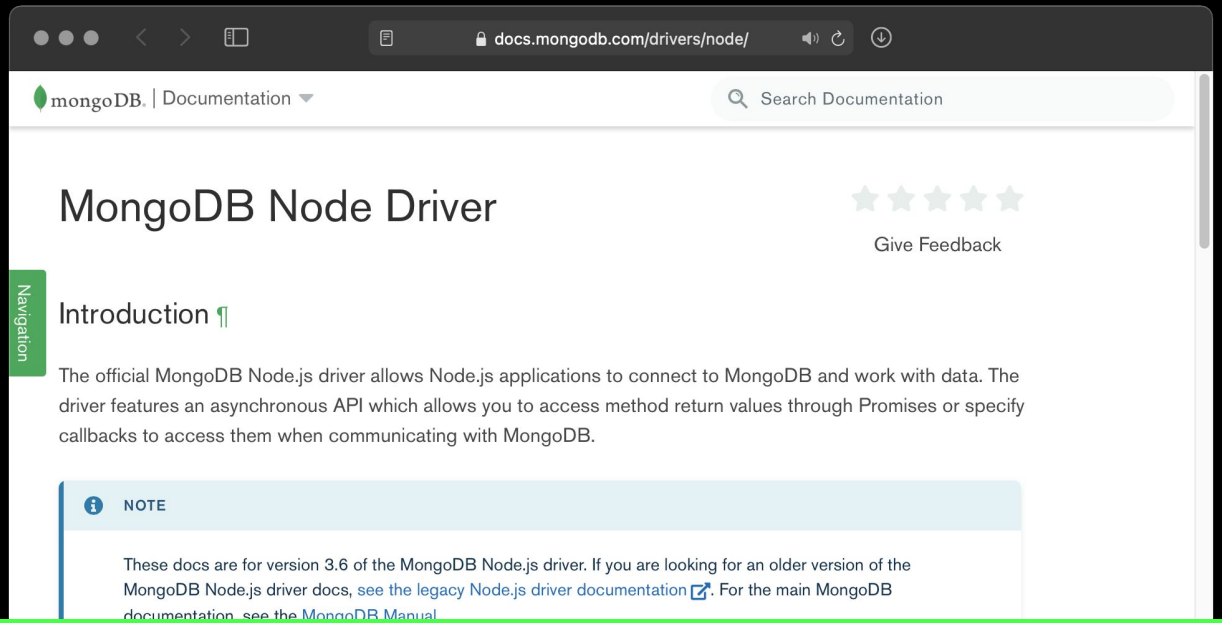
wikipedia.org

connect

mongodb

- JavaScript can be used in queries, aggregation function
- Map-reduce can be used for batch processing of data and aggregation operations
- Manage massive increases in new, rapidly changing data types

https://www.mongodb.com/compare/mongodb-mysql



Note: there are a lot of small steps involved. Read the Mongo guides very carefully. If you miss a step everything will be broken.

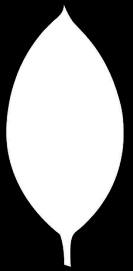
bash

```
$ npm install mongodb

+ mongodb@2.2.33
+ dotenv@4.0.0
added 11 packages in 4.022s

$
```

mongodb wraps **MongoDB** for **Node**



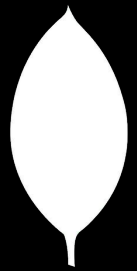
connect

mongodb

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

.env

```
DB_HOST=localhost
DB_PORT=27017
DB_NAME=mymoviewebsite
```



connect

mongodb

```
// Files
mongodb-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
```

.env

```
DB_HOST=localhost
DB_PORT=27017
DB_NAME=mymoviewebsite
```

Note: Never ever put your **host and password in code or on GitHub**! People will be able to access your database!

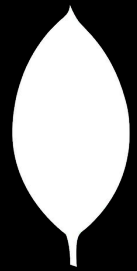
connect

mongodb

```
// Files
mongodb-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
│  ├─ not-found.ejs
│  └─ tail.ejs
├─ .env
├─ .gitignore
├─ index.js
└─ package.json
```

.gitignore

```
node_modules/
.DS_Store
.env
```



II MONG

MONGODB

index.js

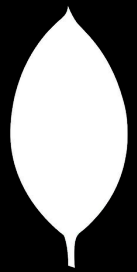
```
…
var multer = require('multer')
var mongo = require('mongodb')

require('dotenv').config()

var db = null
var url = 'mongodb://' + process.env.DB_HOST + ':' +
process.env.DB_PORT

mongo.MongoClient.connect(url, function (err, client) {
  if (err) throw err
  db = client.db(process.env.DB_NAME)
})

…
```



Connect

connect

> Create a database and set-up your remote connection

Synopsis

- **Time**: 2:00h
- **Goals**: subgoal 7, subgoal 8
- **Due**: before week 4

Description

We'll cover actually storing the user input and learning about databases the next lesson but what you already do is create a MongoDB atlas account and set-up your mongodb connection.

All of your data is going to be stored in MongoDB. Take ± 2 hours to set-up your database and connect to it. There are roughly two options: go

work on **connect**

Connect

connect

> Create a database and set-up your remote connection

Synopsis

- **Time**: 2:00h
- **Goals**: subgoal 7, subgoal 8
- **Due**: before week 4

Description

Note: If you are ready, **begin on week-4** because after the holiday we ‘only’ have one week left before A1.

exit;

see you in `lab-4a`!