

back-end

Database

lab 4/8



Show what
you did

Stand-up!

Review

In the lab you will have time to have a look at the assessment checklist and grading fellow students based on the rubric. Ask your teacher what teachers expect from the project and do some technical troubleshooting.

A1 peer review

This is the peer review document you will fill-in as duo. You'll perform the checks on the project of your fellow student. It's a good last check to see if everything is in order.

→ [Peer Review](#)

A1 rubric:

This is the rubric your teacher will grade you on during the assessment. Ask yourself upon completion if everything listed on the rubric is clear and that you understand each row and column, if not ask your teacher on MS Teams!

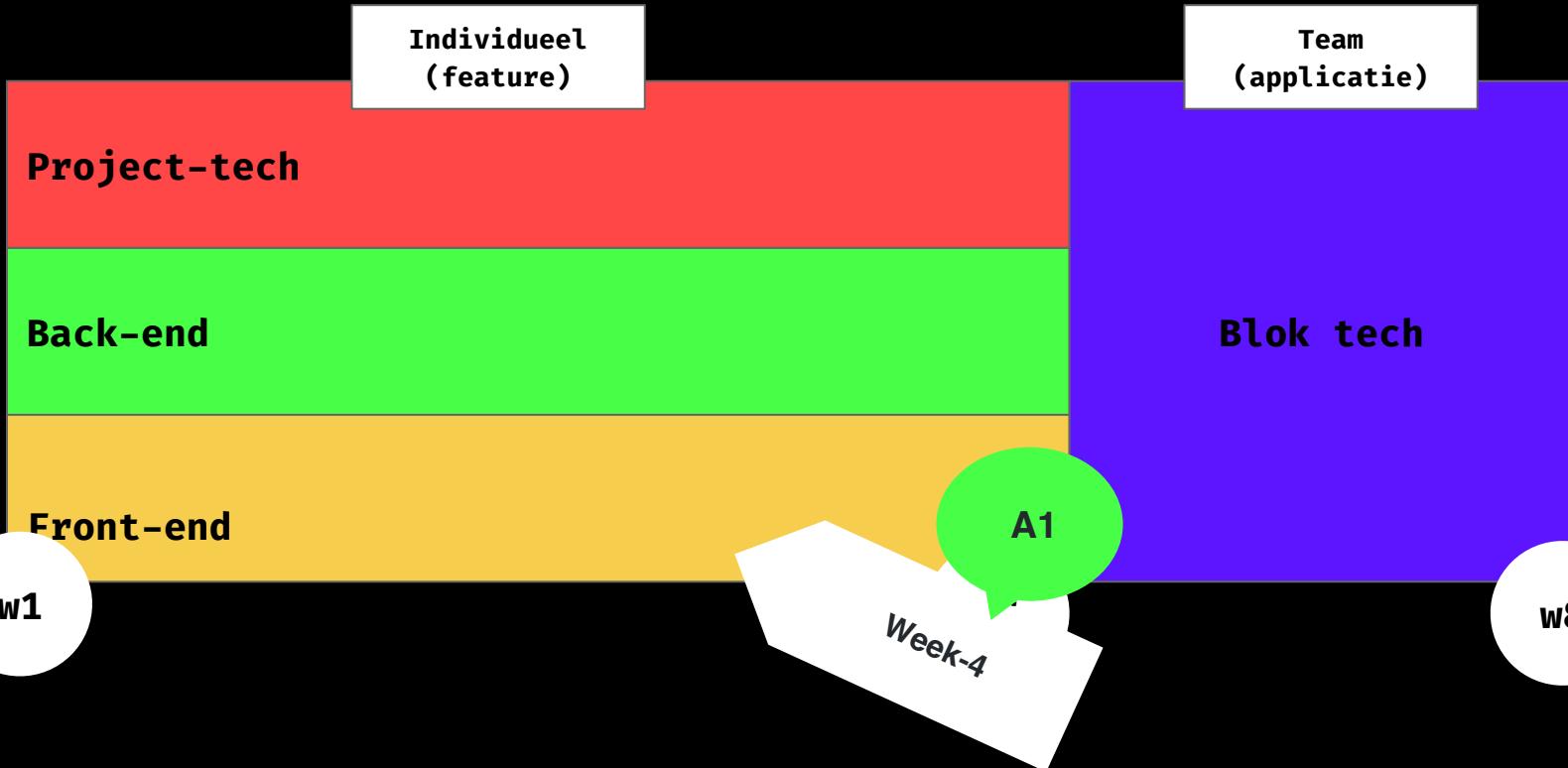
→ [Rubric](#)

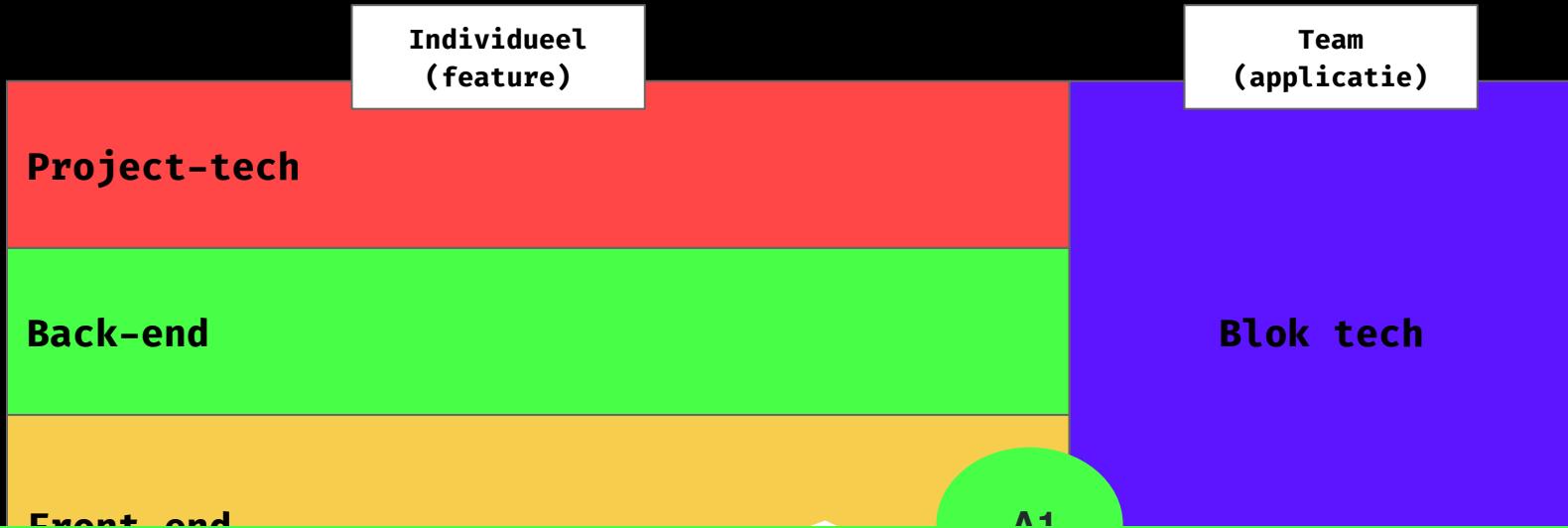
Questions:

If you have any question you can reach the student-assistants and teachers on MS Teams. Make sure you are prepared:

**Make a
copy of
the file**

peer review





Note: last class before assessment.

Next week is 'hemelvaart' and the week after is A1.

today

I. ~~Stand-up~~

II. A1 Grading

III. SQL vs NoSQL

IV. Connect

V. Crud

A1

course

goals

- ❖ You can build web apps with Node and use packages from NPM
- ❖ You can communicate over HTTP and understand client/server flow
- ❖ You can render data server-side with a templating engine
- ❖ You can store data in a database and update that data
- ❖ You can write documentation that other developers understand
- ❖ You can explain your code and the cohesion of your application

course

deliverables

- ❖ **Individual Prototype:** working **interactive feature** for serious relationships
- ❖ **Process book (wiki):** that provides insight into the weekly iterative process and your research

Assessment 1 - Individual

You've worked iteratively (*formative*) on your product and finish with an oral test (*summative*). You'll **show the feature** you've built based on your code in your repository and live version. *A teacher will try out your feature and look at the code.*



You will show you can create a quality project in which you apply the subject matter of this course and that you understand it.

You will answer questions in such a way as to demonstrate sufficient knowledge of our goals.

This is an individual assessment, so tests will be conducted between one teacher and one student.

This is an assessment, not another moment for feedback. So you will be graded. There isn't much time for additional feedback or troubleshooting technical issues.

Preparation

Since we have limited time make sure you come to the assessment prepared:

- Bring your computer and make sure it's *charged* and connected to *Wifi*.
- Make sure your *webcam, microphone* and *screen sharing works in MS Teams*
- Have the latest version of your feature **ready in your browser**.
- Have the latest version of your code **ready on GitHub**.

let's look at the **rubric for A1**

SQL & NoSQL

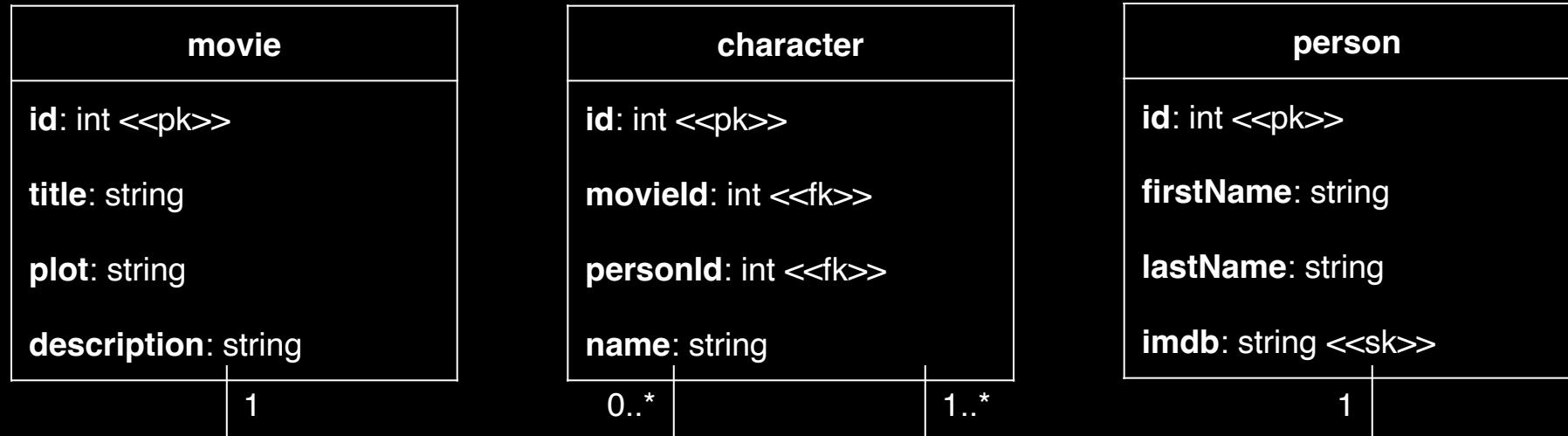
sql

?

SQL (Structured Query Language) is a [...] language used in programming and designed for managing data held in a relational database [...].

sql

structure



nosql

?

A NoSQL (originally referring to “non SQL” or “non relational”) database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. [...]

nosql

?

[...] Such databases have existed since the late 1960s, but did not obtain the “NoSQL” moniker until a surge of popularity in the early twenty-first century, triggered by the needs of Web 2.0. [...]

mongodb

?

MongoDB (from **humongous**) 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. [...]

mongodb

?

MongoDB (from **humongous**) 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](https://en.wikipedia.org/wiki/MongoDB)

DATABASES: 5 COLLECTIONS: 17

+ Create Database

NAMESPACES

- sample_airbnb
- sample_geospatial
- sample_mflix**
 - comments
 - movies**
 - sessions
 - theaters
 - users
- sample_training
- sample_weatherdata

REFRESH

sample_mflix.movies

COLLECTION SIZE: 61.82MB TOTAL DOCUMENTS: 45993 INDEXES TOTAL SIZE: 37.95MB

Find Indexes

INSERT DOCUMENT

FILTER {"filter":"example"}

Find Reset

QUERY RESULTS 1-20 OF MANY

```
_id: ObjectId("573a1390f29313caabcd4132")
title: "Carmencita"
year: 1894
runtime: 1
> cast: Array
  poster: "http://ia.media-imdb.com/images/M/MV5B..."/>
  plot: "Performing on what looks like a small stage, Carmencita (Lola Albright) and her troupe..."/>
  fullplot: "Performing on what looks like a small stage, Carmencita (Lola Albright) and her troupe..."/>
  lastupdated: "2015-08-26 00:03:45.040000000"
  type: "movie"
> directors: Array
> imdb: Object
> countries: Array
  rated: "NOT RATED"
> genres: Array
```

Keys & values

DATABASES: 5 COLLECTIONS: 17

+ Create Database

NAMESPACES

- sample_airbnb
- sample_geospatial
- sample_mflix**
 - comments
 - movies**
 - sessions
 - theaters
 - users
- sample_training
- sample_weatherdata

REFRESH

sample_mflix.movies

COLLECTION SIZE: 61.82MB TOTAL DOCUMENTS: 45993 INDEXES TOTAL SIZE: 37.95MB

Find Indexes

INSERT DOCUMENT

FILTER {"filter": "example"}

Find Reset

QUERY RESULTS 1-20 OF MANY

```
_id: ObjectId("573a1390f29313caabcd4132")
title: "Carmencita"
year: 1894
runtime: 1
> cast: Array
  poster: "http://ia.media-imdb.com/images/M/MV5B..."/>
  plot: "Performing on what looks like a small stage, Carmencita (Lina Bas奎尔) is performing a..."/>
  fullplot: "Performing on what looks like a small stage, Carmencita (Lina Bas奎尔) is performing a..."/>
  lastupdated: "2015-08-26 00:03:45.040000000"
  type: "movie"
  > directors: Array
```

Keys & values

Think about how you want to structure your data.
This is called **data modelling**.

mongodb

atlas

The screenshot shows the MongoDB Atlas landing page. At the top, there's a navigation bar with the MongoDB logo, Cloud, Software, Learn, Solutions, Docs, a search icon, Contact, Sign In, and a prominent green 'Try Free' button.

The main heading is 'MongoDB Atlas'. Below it, a sub-headline reads: 'Move faster with a cloud MongoDB service. Built for agile teams who'd rather spend time building apps than managing databases. Available on AWS, Azure, and GCP.' A large green 'Start free' button is visible.

Below the headline, a message says: 'Already have an account? [Log in here](#) →'. To the right, a 'Cloud Provider & Region' configuration panel is open. It shows 'aws' selected as the provider and 'AWS, N. Virginia (us-east-1)' as the region. The panel includes sections for 'NORTH AMERICA', 'EUROPE', 'ASIA', and 'SOUTH AMERICA' with various region options listed. A note at the top of the panel says: 'Configure a free tier cluster by first selecting a region labeled with [FREE TIER AVAILABLE](#), then choose the M0 option on the Cluster Tier below.'

At the bottom of the main content area, a summary statement reads: 'MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class'.

Below this summary, there are links for 'Pricing', 'Getting started', 'Migrate to MongoDB Atlas', and 'Frequently Asked Questions'.

<https://www.mongodb.com/cloud/atlas>

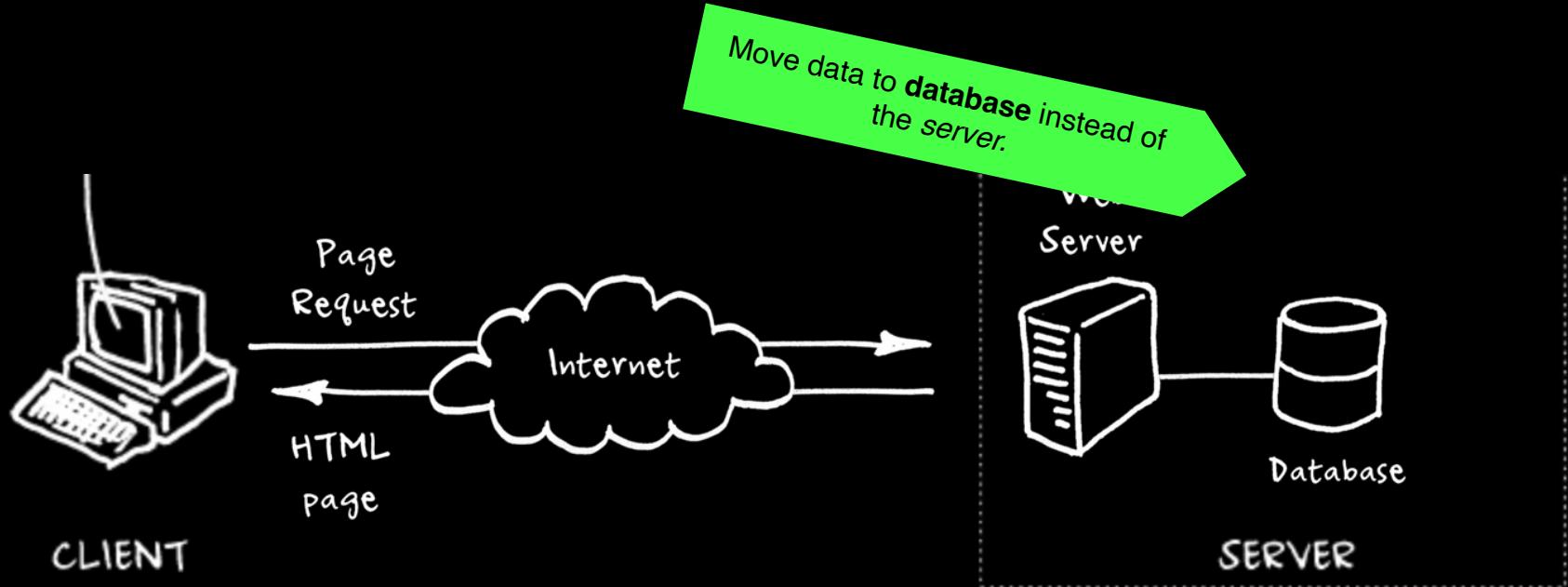
Assignment ($\pm 20m$)

Data model your database and draw / write how you are going to structure your database.

- Look at your interface, what inputs are you using?
- What data types do you need?
- What relations between tables do you need?



Connect



connect

mongodb

MongoDB (from **humongous**) 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. [...]

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>

The screenshot shows a web browser displaying the MongoDB Node Driver documentation at docs.mongodb.com/drivers/node/. The page title is "MongoDB Node Driver". On the left, there's a vertical navigation bar with a green header labeled "Navigation". The main content area features a section titled "Introduction" with a subtitle "NOTE". The "NOTE" section contains a message about the version of the documentation.

MongoDB Node Driver

mongDB | Documentation ▾

Search Documentation

Give Feedback

★★★★★

Introduction ¶

NOTE

These docs are for version 3.6 of the MongoDB Node.js driver. If you are looking for an older version of the MongoDB Node.js driver docs, see the legacy Node.js driver documentation [↗](#). For the main MongoDB documentation, see the [MongoDB Manual](#).

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

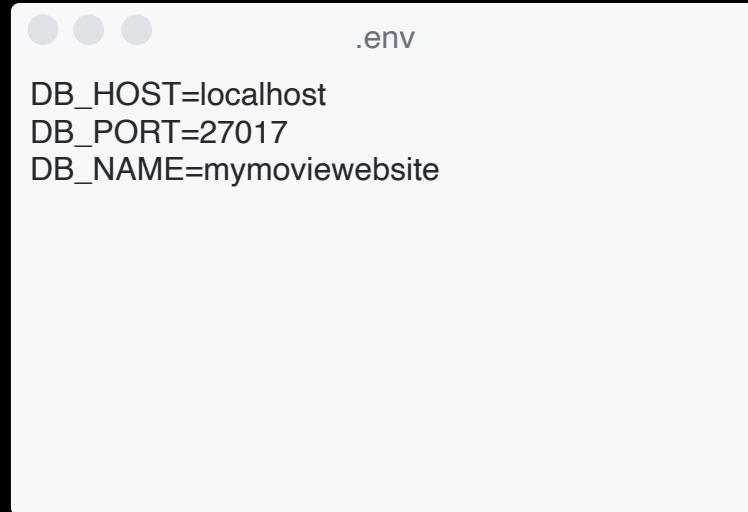
```
$ npm install mongodb dotenv  
+ mongodb@2.2.33  
+ dotenv@4.0.0  
added 11 packages in 4.022s  
$
```

mongodb wraps MongoDB
for Node

connect

```
// 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
```

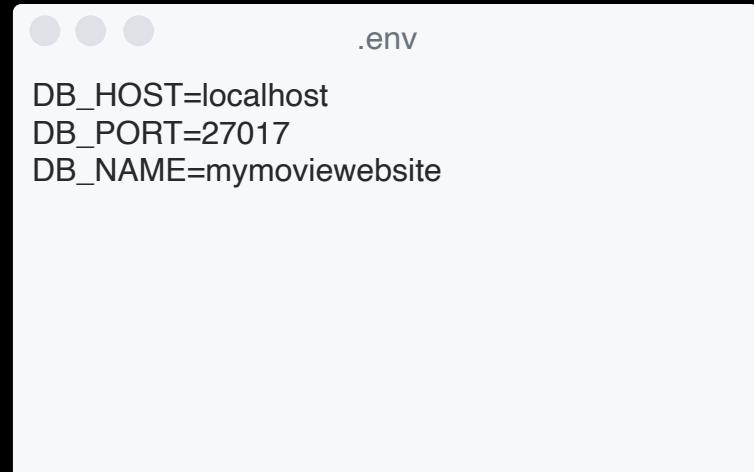
mongodb



connect

mongodb

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

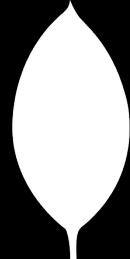


Note: Never ever put your **host and password in code or on GitHub!** People will be able to access your database!

connect

```
// 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
```

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)
})

...
...
```

Crud



index.js

```
...  
  
function movies(req, res, next) {  
  db.collection('movie').find().toArray(done)  
  
  function done(err, data) {  
    if (err) {  
      next(err)  
    } else {  
      res.render('list.ejs', {data: data})  
    }  
  }  
}  
  
...
```

index.js

```
...
function movie(req, res, next) {
  var id = req.params.id
  db.collection('movie').findOne({
    _id: mongo.ObjectID(id)
  }, done)

  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.render('detail.ejs', {data: data})
    }
  }
}

...

```

index.js

```
function add(req, res, next) {
  db.collection('movie').updateOne({
    _id: ObjectId(req.body._id),
    {$set: {textProfile: req.body.description}}
  }, done)

  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.redirect('/' + data.insertedId)
    }
  }
}
```



index.js

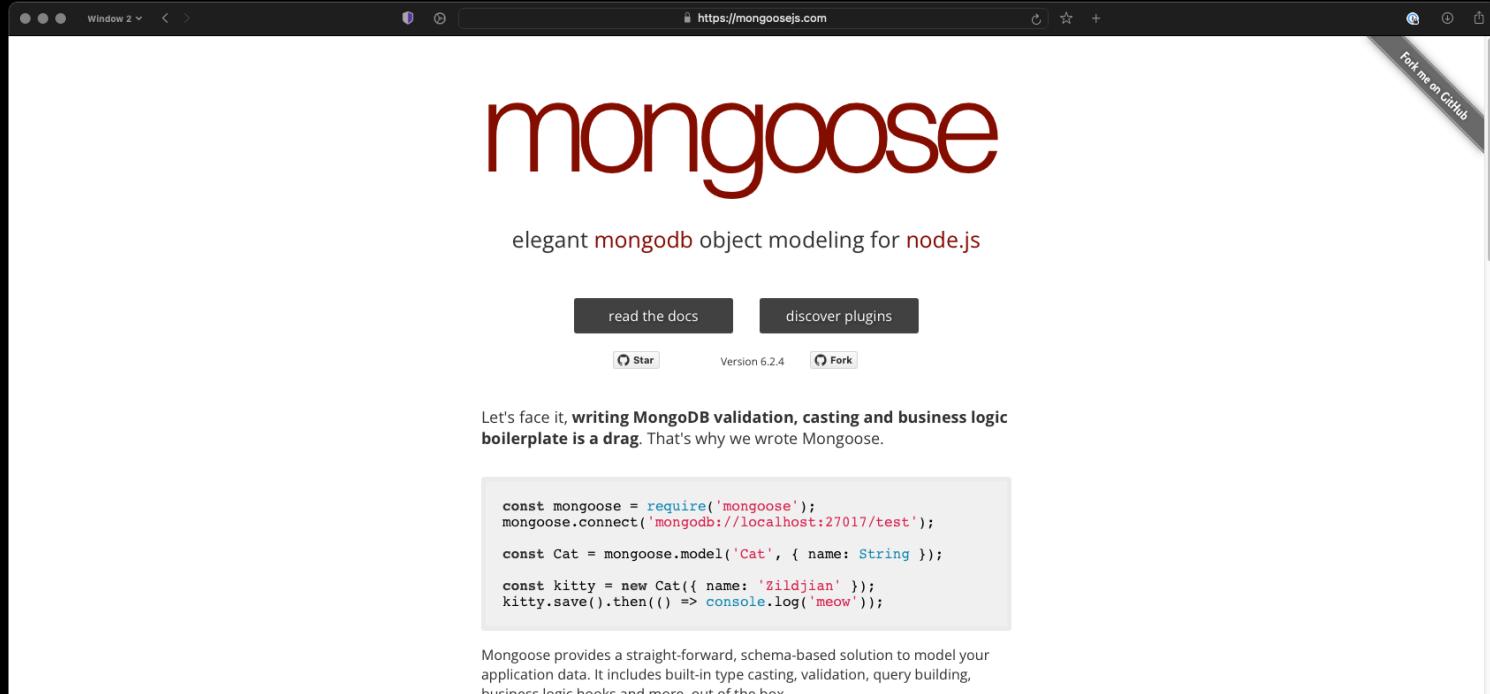
```
...
function remove(req, res, next) {
  var id = req.params.id
  db.collection('movie').deleteOne({
    _id: mongo.ObjectID(id)
  }, done)

  function done(err) {
    if (err) {
      next(err)
    } else {
      res.json({status: 'ok'})
    }
  }
}

...

```

Note: for A1 to pass you'll need
to **find and update**



Pick **mongoDB (default driver)** over Mongoose.

Windows 2

MongoDB. Products Solutions Resources Company Pricing

University Home > Browse Courses > M001

M001 MongoDB Basics

INTRODUCTORY

Learn the fundamentals of MongoDB.

Estimated Total Effort
8.5 Hours

Start Course

All courses at MongoDB University are free! Get started now by learning MongoDB directly from the source.

What You'll Learn

In this course you will learn how to set up your database and start exploring different ways to search, create, and analyze your data with MongoDB. We will cover database performance basics, and discover how to get started with creating applications and visualizing your data.

We'll start together with the ultimate basics, learning what a database is and recognizing what makes MongoDB different in the database space. Then you'll move on to working with data as you grasp the difference between BSON and JSON and start to import, export and query. Next you'll absorb how to create and manipulate documents and learn how to use the advanced Create Read Update Delete (CRUD) operations. By this time, you'll be ready to work on indexing, Data Modeling, and creating an Aggregation Pipeline. Lastly, you'll have the opportunity to explore the Atlas UI in more detail, investigate the Charts functionality and Realm, as well as explore the use of Compass.

This course is rich in hands-on learning and additional resources to support your educational experience with MongoDB University. It has been developed and taught by a MongoDB Curriculum Engineer at MongoDB University, where we strive to free the genius within everyone by making data stunningly easy to work with.

In this course you'll get your hands on all the basics, including querying, computing, connecting to, storing, indexing and analyzing your data.

PREREQUISITES

A basic knowledge of programming concepts such as command line and

SYSTEM REQUIREMENTS

Web Browser: Firefox 60.0+ or Chrome 70+

Operating System: Mac OS X 10.7+ 64-bit, Ubuntu 14.04+ 64-bit, or Windows 8+ (64-bit)

YOUR INSTRUCTOR



Yulia Genkina
MongoDB Course Instructor

Yulia is a Senior Curriculum Engineer at MongoDB. Prior to MongoDB Yulia worked at Stuyvesant High School where she taught Computer Science to hundreds of unsuspecting students.

Already have a MongoDB account? Sign in.

Get Started

MongoDB University

```
function add(req, res, next) {  
  db.collection('movie').updateOne({  
    _id: ObjectId(req.body._id),  
    {$set: {textProfile: req.body.description}}  
  
function done(err, data) {  
  if (err) {  
    next(err)  
  } else {  
    res.redirect('/' + data.insertedId)  
  }  
}
```

live demo **crud operations**

exit;

see you in **lab-5**!