

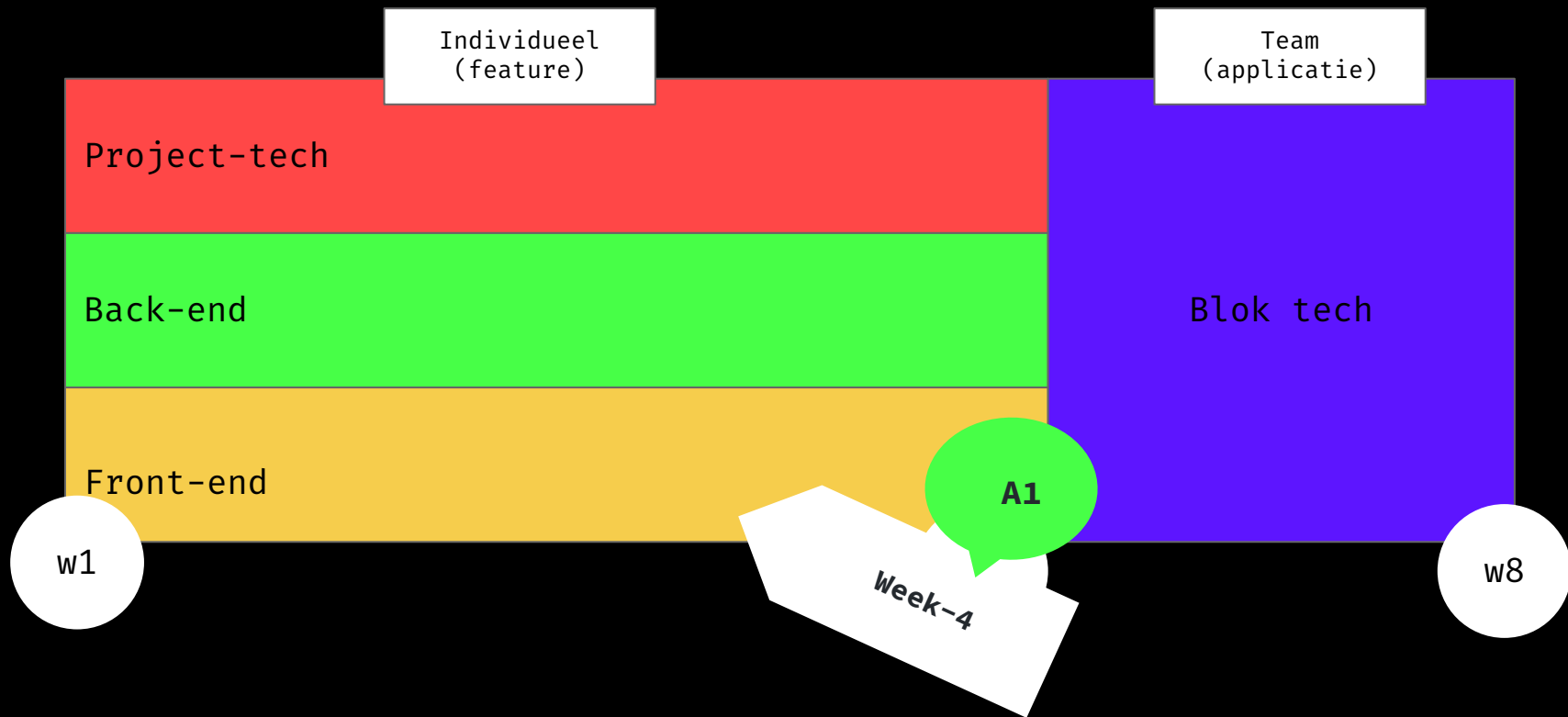
# back-end

## Database

lab 4/8

Stand-up!

Show what  
you did



# today

~~I. Stand-up~~

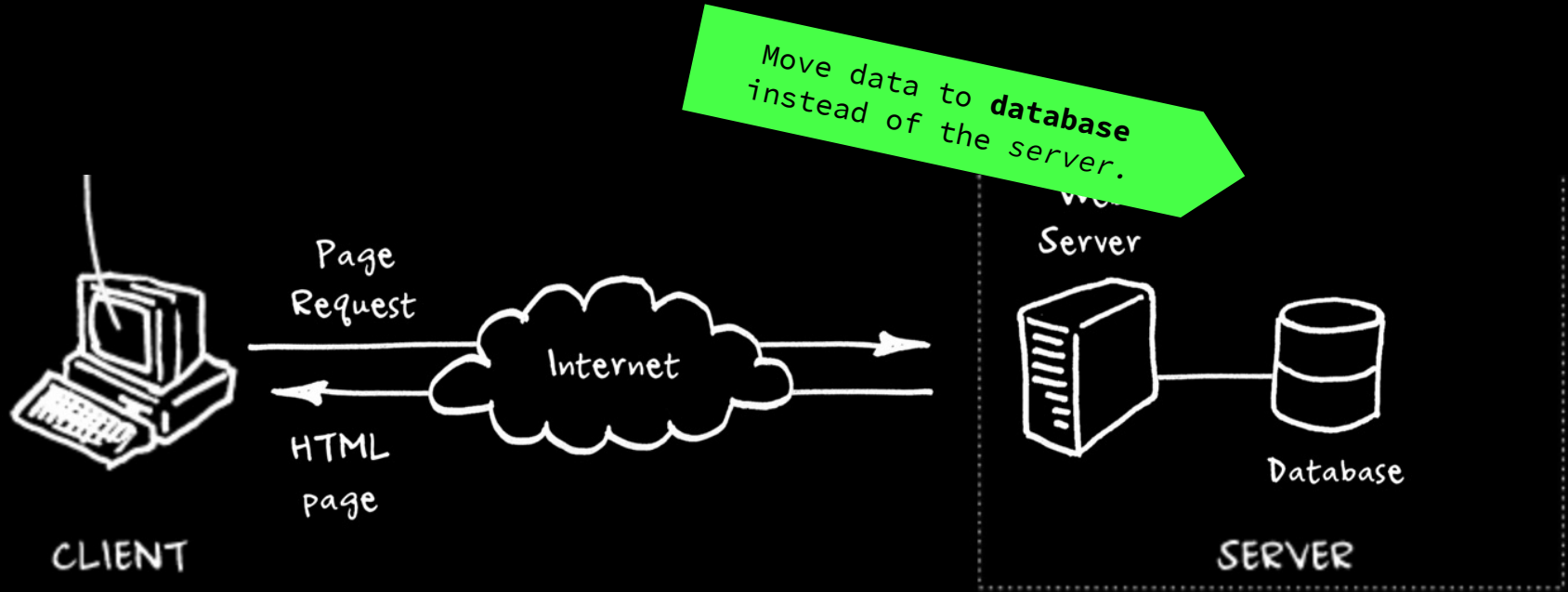
II. Connect (recap)

III. SQL vs NoSQL

IV. Crud

The background is a solid black field populated with numerous small, light green geometric shapes. These shapes include triangles of various sizes and orientations, as well as wavy, squiggly lines. The distribution of these shapes is random and sparse, creating a textured, abstract effect.

# Connect



# connect

mongodb

MongoDB (from hum**ong**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](https://www.wikipedia.org)

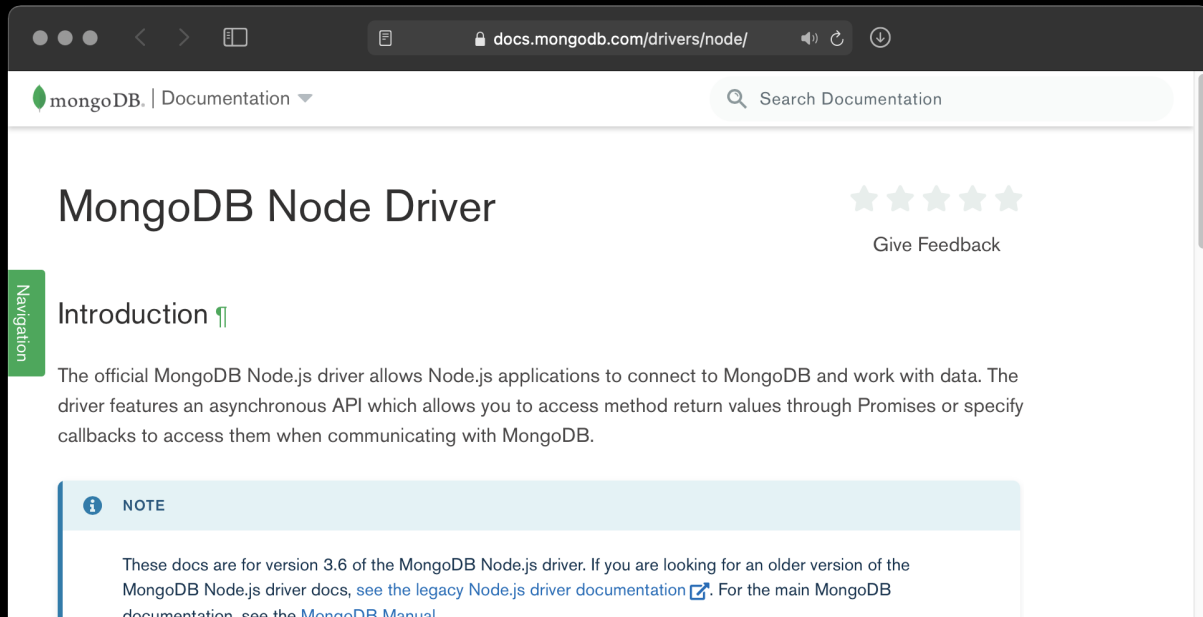
# connect

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 wraps MongoDB  
for Node**

```
+ mongodb@2.2.33
```

```
+ dotenv@4.0.0
```

```
added 11 packages in 4.022s
```


```
$
```



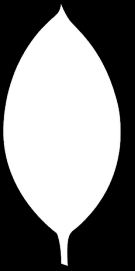
# connect

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
DB_USERNAME=dandevri
```



# 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
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
```



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)  
})  
  
...
```





```
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  
  }  
}
```

live demo [.env en connection](#)

# SQL & NoSQL



# sql

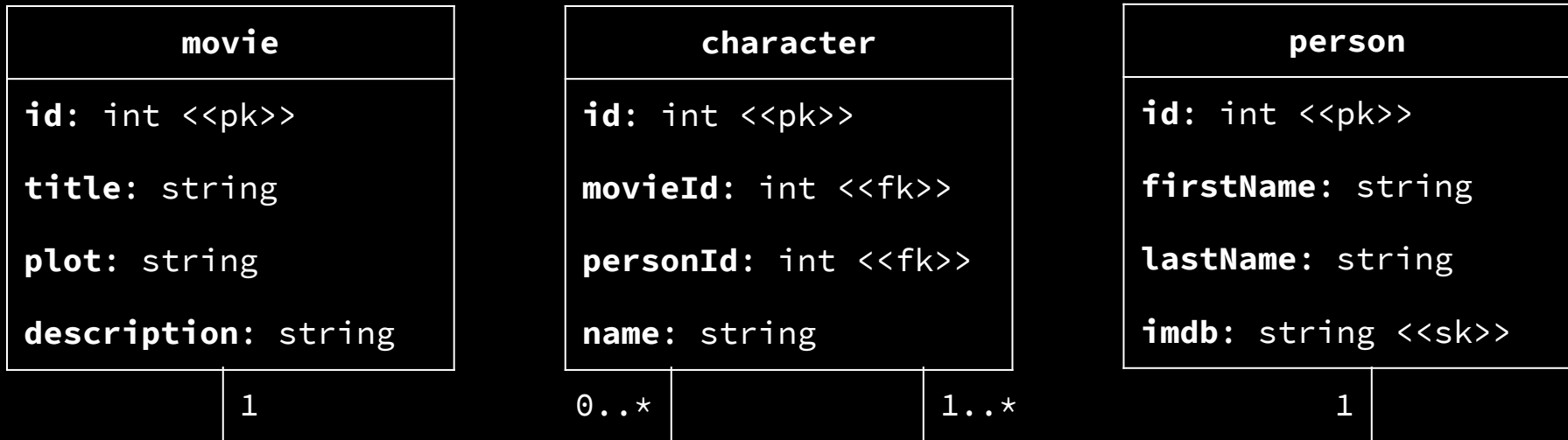
?

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

[wikipedia.org](https://en.wikipedia.org)

# 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. [...]

[wikipedia.org](https://en.wikipedia.org)

# mongodb

?

MongoDB (from hum**ong**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](https://www.wikipedia.org)

# 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. [...]

JSON like documents (no relations)

[wikipedia.org](https://www.wikipedia.org)

DATABASES: 5 COLLECTIONS: 17

REFRESH

+ Create Database

Q NAMESPACES

▶ sample\_airbnb

▶ sample\_geospatial

▼ sample\_mflix

comments

movies

sessions

theaters

users

▶ sample\_training

▶ sample\_weatherdata

## sample\_mflix.movies

COLLECTION SIZE: 61.82MB

TOTAL DOCUMENTS: 45993

INDEXES TOTAL SIZE: 37.95MB

Find

Indexes

INSERT DOCUMENT

FILTER {"filter":"example"}

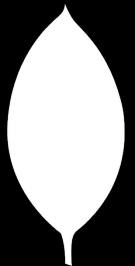
Find

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
  fullplot: "Performing on what looks like a
  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

REFRESH

+ Create Database

Q NAMESPACES

- ▶ sample\_airbnb
- ▶ sample\_geospatial
- ▼ sample\_mflix
  - comments
  - movies
  - sessions
  - theaters
  - users
- ▶ sample\_training
- ▶ sample\_weatherdata

## 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
fullplot: "Performing on what looks like a
lastupdated: "2015-08-26 00:03:45.040000000"
type: "movie"
> directors: Array
```

Keys & values

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



# mongodb

# atlas

The screenshot shows the MongoDB Atlas website. The header includes the MongoDB logo, navigation links (Cloud, Software, Learn, Solutions, Docs), a search icon, and links for Contact, Sign In, and a green 'Try Free' button. The main content area has a green background with the text 'MongoDB Atlas' and a description: '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.' Below this is a 'Start free' button and a link for existing users. A modal window titled 'Cloud Provider & Region' is open, showing options for AWS, Google Cloud, and Azure. It lists recommended regions for each provider, such as N. Virginia for AWS and Ireland for Google Cloud. The footer contains links for Pricing, Getting started, Migrate to MongoDB Atlas, and Frequently Asked Questions. At the bottom, a text block states: 'MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class'.

mongodb Cloud Software Learn Solutions Docs

Search Contact Sign In Try Free

## MongoDB Atlas

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.

Start free

Already have an account? [Log in here](#) →

### Cloud Provider & Region

Choose your preferred cloud provider and the region nearest to clients

Select a cloud provider to see its region availability

aws google cloud azure

Configure a free tier cluster by first selecting a region labeled with [FREE TIER AVAILABLE](#). Then choose the M0 option in the Cluster Tier below.

recommended region (3)

North America	Europe	Asia
<ul style="list-style-type: none"><li>N. Virginia (us-east-1) <b>FREE TIER AVAILABLE</b></li><li>Ohio (us-west-1)</li><li>N. California (us-west-1)</li><li>Oregon (us-west-2)</li><li>Montreal (ca-central-1)</li></ul>	<ul style="list-style-type: none"><li>Ireland (eu-west-1)</li><li>London (eu-west-2)</li><li>Frankfurt (eu-central-1) <b>FREE TIER AVAILABLE</b></li><li>São Paulo (sa-east-1)</li></ul>	<ul style="list-style-type: none"><li>Tokyo (ap-northeast-1)</li><li>Seoul (ap-northeast-2)</li><li>Singapore (ap-southeast-1)</li><li>Mumbai (ap-south-1)</li></ul>

Pricing Getting started Migrate to MongoDB Atlas Frequently Asked Questions

MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or GCP. Best-in-class

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

The background is a solid black field filled with a repeating pattern of small, light green geometric shapes. These shapes include triangles, squares, and circles, some of which are further divided into smaller sub-shapes, creating a complex, fractal-like texture. The word "Crud" is centered in the image in a bright green, sans-serif font.

Crud



bash

```
$ npm install mongodb dotenv
```

**mongodb wraps MongoDB  
for Node**

```
+ mongodb@2.2.33
```

```
+ dotenv@4.0.0
```

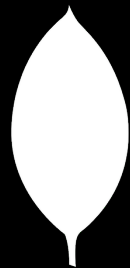
```
added 11 packages in 4.022s
```

```
$
```



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)  
})  
  
...
```

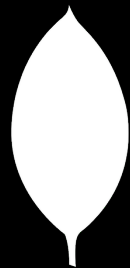


index.js

...

```
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)

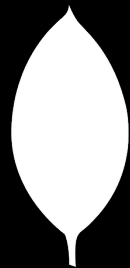
  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.render('detail.ejs', {data: data})
    }
  }
}
}
...
```



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  
    }  
  }  
}
```

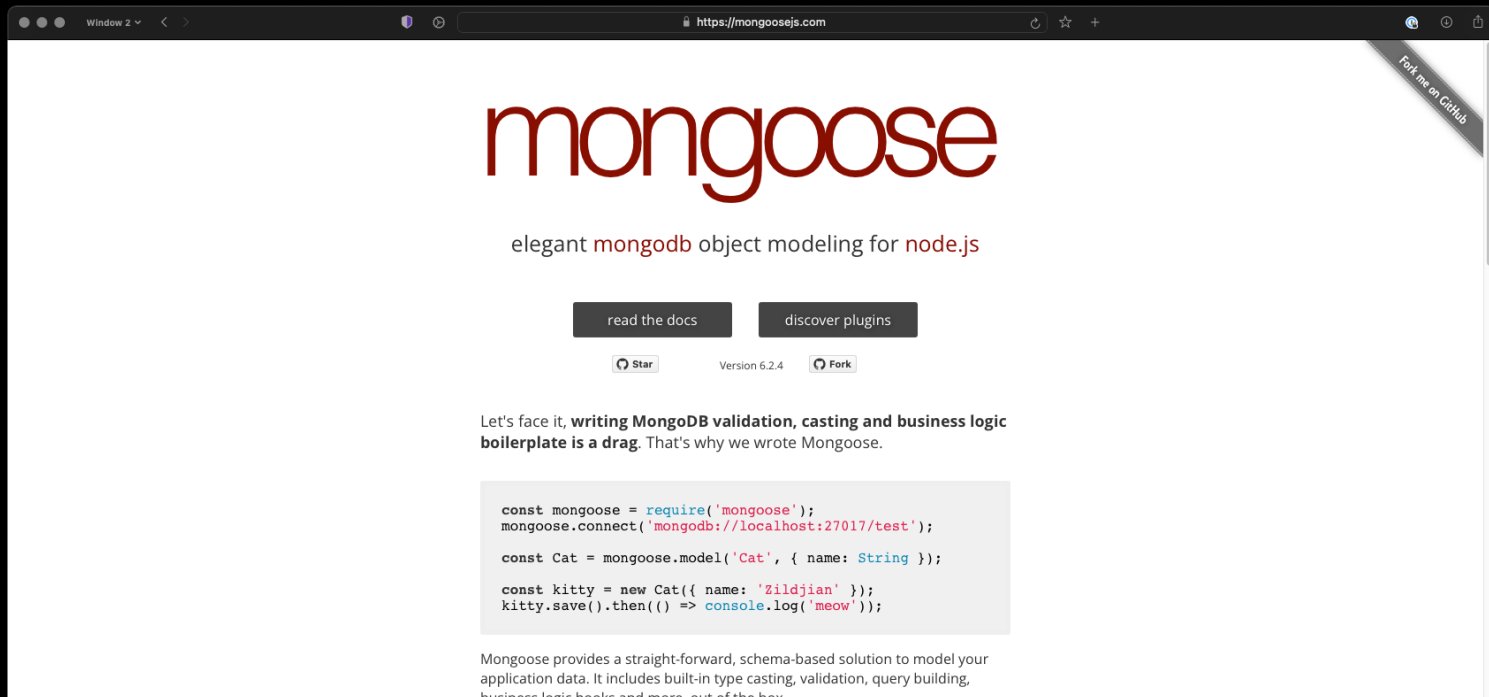


index.js

```
...  
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.

MongoDB

Products

Solutions

Resources

Company

Pricing

Sign In

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 with hands-on learning, and skill-up to mastering 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}}  
  }, done)  
  
  function done(err, data) {  
    if (err) {  
      next(err)  
    } else {  
      res.redirect('/') + data.insertedId  
    }  
  }  
}
```

live demo crud operations



# exit;

see you in lab-5!