

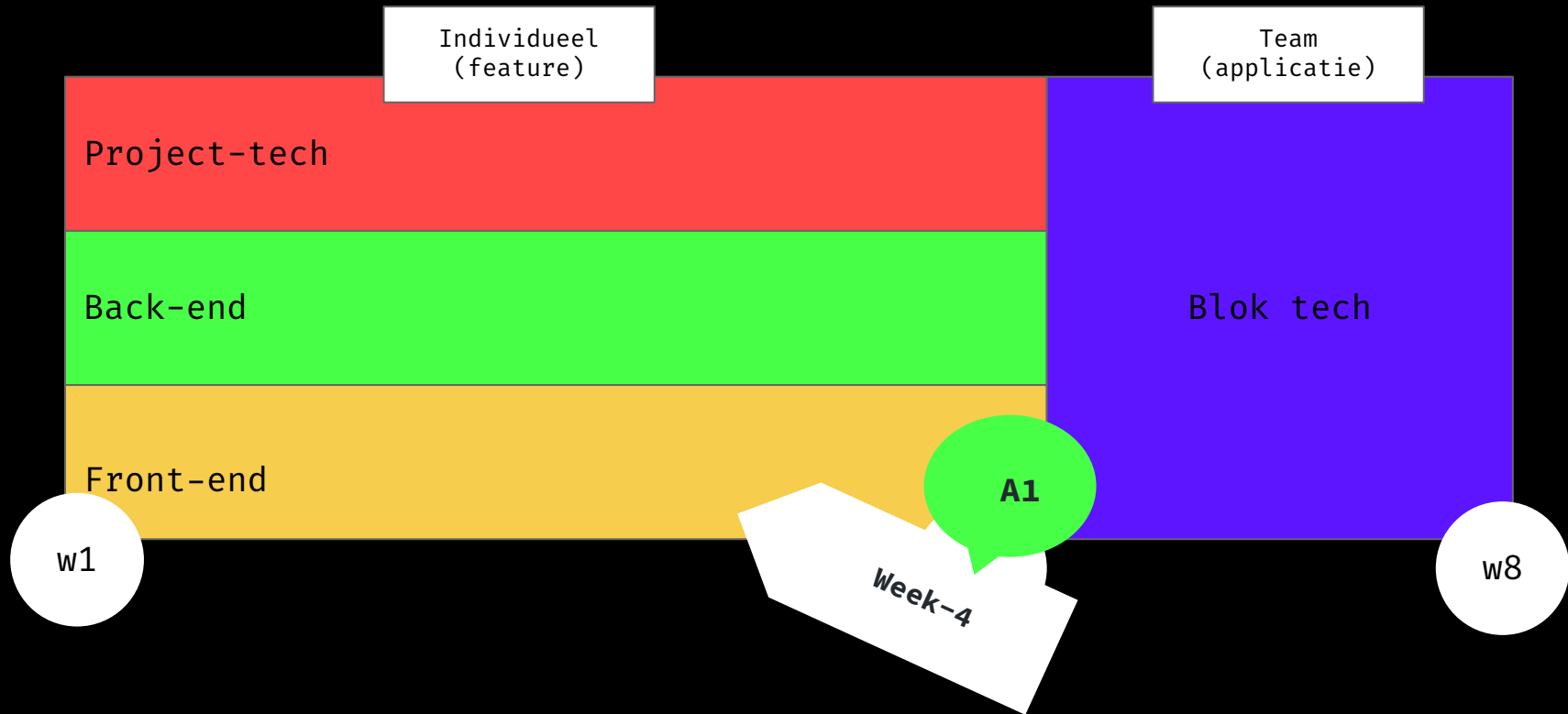
back-end

Database

lab 4/8

Stand-up!

Show what
you did



A1

Praktische info

- Lees goed de info over 'grading' op github
 - Daar staat ook de rubric voor de beoordeling
 - En hoe je je voorbereid op het mondeling
- Lever uiterlijk **<datum en tijd>** een link naar je github repo + een .zip van je code in op DLO
- Schrijf je in op een tijdslot en zorg zelf dat er geen overlap is met het mondeling van project tech en frontend 2

today

~~I. Stand-up~~

~~II. Info about A1~~

III. MongoDB

IV. Peer review

V. Crud

MongoDB

Storing data in db

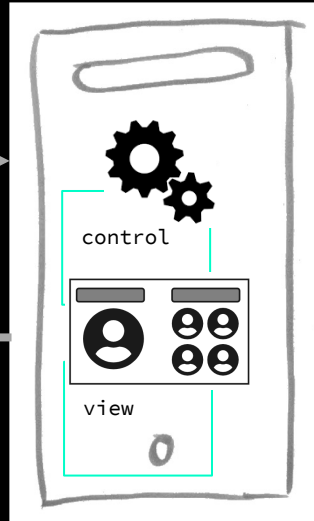
Client
(browser)

Server
(Webserver met Node.js)

Database
(MongoDB)



1.



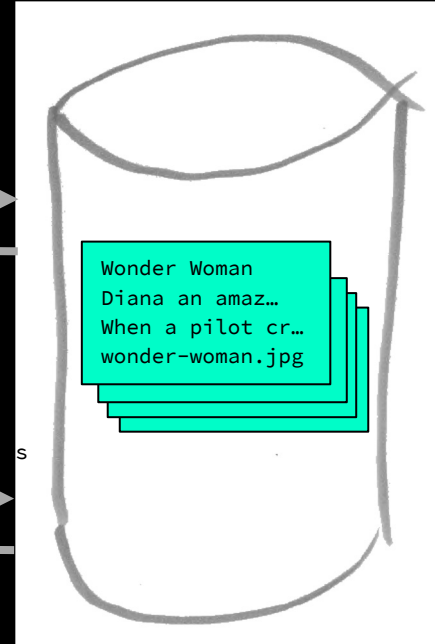
2.

3.

4.

5.

6.



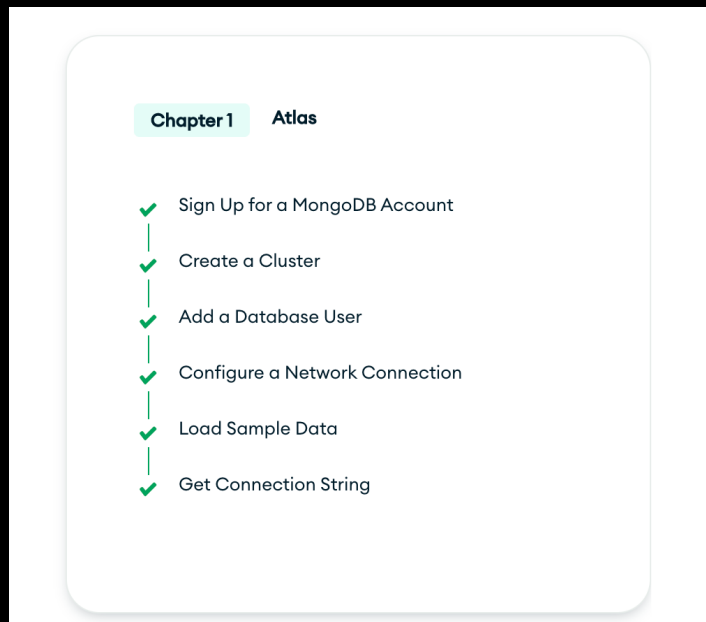
connect

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. [...]

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

connect



Steps in the “MongoDB start with guide”

connect

Chapter 1 Atlas

- ✓ Sign Up for a MongoDB Account
- ✓ Create a Cluster
- ✓ Add a Database User
- ✓ Configure a Network Connection
- ✓ Load Sample Data
- ✓ Get Connection String

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

index.js

...

```
require('dotenv').config()
```

```
const { MongoClient, ServerApiVersion, ObjectId } = require('mongodb')
```

```
const uri = 'mongodb+srv://' + process.env.DB_USERNAME + ':' +  
  process.env.DB_PASS + '@' +  
  process.env.DB_HOST + '/' + process.env.DB_NAME +  
  '?retryWrites=true&w=majority'
```

```
const client = new MongoClient(uri, { useNewUrlParser: true,  
  useUnifiedTopology: true, serverApi: ServerApiVersion.v1 })
```

```
client.connect(err => {  
  if (err) { throw err }  
})
```

...

live demo `.env` en connection



bash

```
$ npm install mongodb d
```

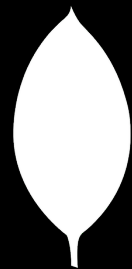
**mongodb wraps MongoDB
for Node**

```
+ mongodb@2.2.33
```

```
+ dotenv@4.0.0
```

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


```
$
```



connect

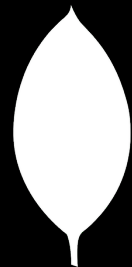
mongodb

```
// 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
```



...

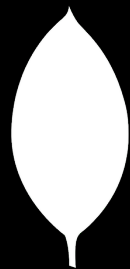
```
require('dotenv').config()
```

```
const { MongoClient, ServerApiVersion } = require('mongodb')
const uri = 'mongodb+srv://' + process.env.DB_USERNAME + ':' +
  process.env.DB_PASS + '@' +
  process.env.DB_HOST + '/' + process.env.DB_NAME +
  '?retryWrites=true&w=majority'
```

```
const client = new MongoClient(uri, { useNewUrlParser: true,
  useUnifiedTopology: true, serverApi: ServerApiVersion.v1 })
```

```
client.connect(err => {
  if (err) { throw err }
})
```

...



index.js

...

```
function add(req, res) {  
  var id = slug(req.body.title).toLowerCase()
```

```
  data.push({  
    id: id,  
    title: req.body.title,  
    plot: req.body.plot,  
    description: req.body.description  
  })
```

```
  res.redirect('/') + id  
}
```

...

title, plot, and description come
from **name** attributes on form inputs

Express

Wonder Woman

When a pilot crashes and tells of conflict in the outside world, Diana, an Amazonian warrior in training, leaves home to fight a war, discovering her full powers and true destiny.

...and we can submit the form!

localhost:8000/wonder-woman

Wonder Woman

When a pilot crashes and tells of conflict in the outside world, Diana, an Amazonian warrior in training, leaves home to fight a war, discovering her full powers and true destiny.

remove

...but we cannot **remove** movies? And we can't **update** them.

Express

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 white 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 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'.

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">N. Virginia (us-east-1) FREE TIER AVAILABLEOhio (us-west-1)N. California (us-west-1)Oregon (us-west-2)Montreal (ca-central-1)	<ul style="list-style-type: none">Ireland (eu-west-1)London (eu-west-2)Frankfurt (eu-central-1) FREE TIER AVAILABLEStockholm (eu-north-1)São Paulo (sa-east-1)	<ul style="list-style-type: none">Tokyo (ap-northeast-1)Bangkok (ap-southeast-1)Singapore (ap-southeast-1)Mumbai (ap-south-1)

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>



Break!

Back-end Peer Review


peer review for assessment 1

Reviewer	
Reviewee	
Class	
Teacher	

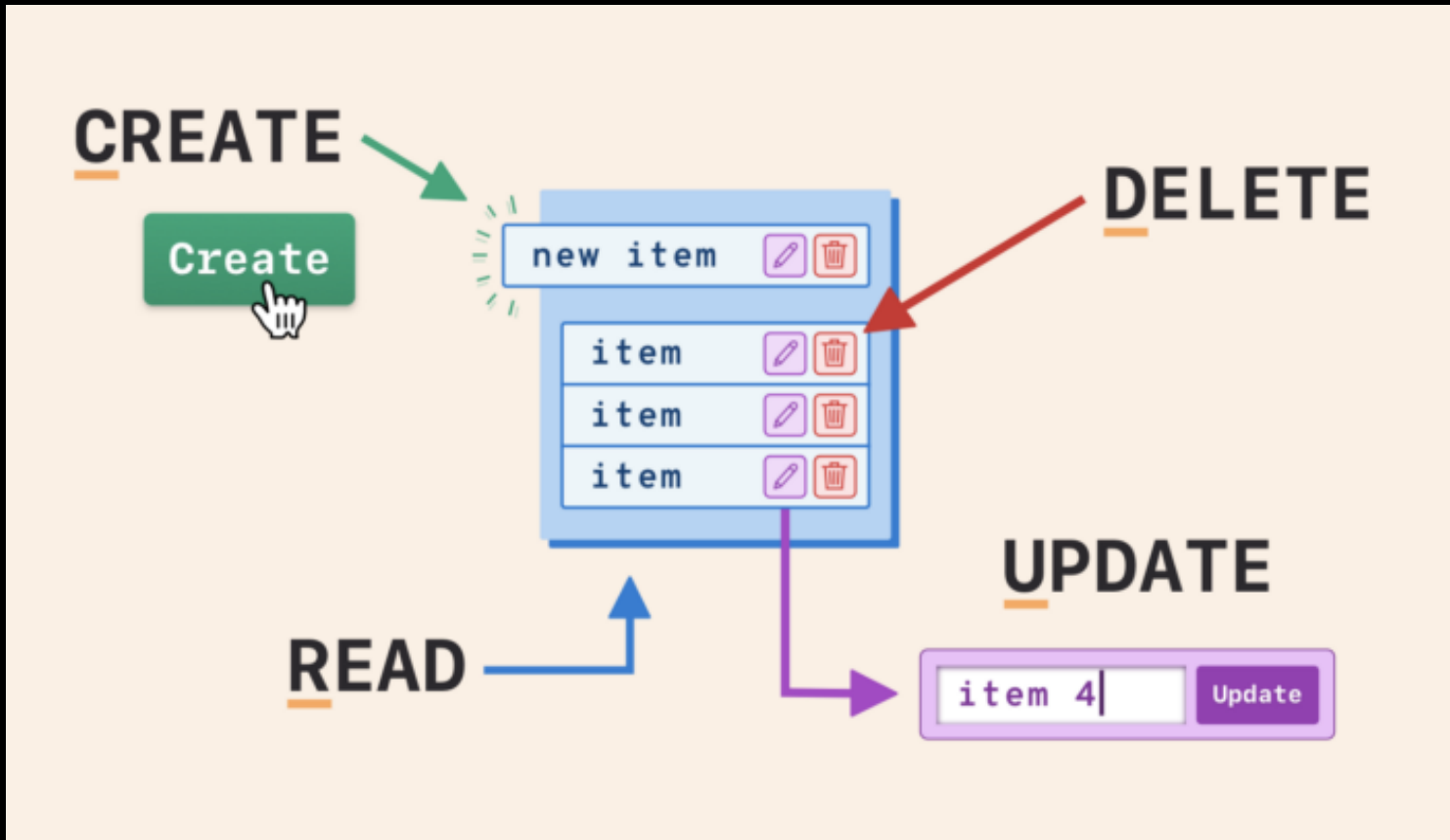
Checklist

Project is documented and has a readme.md	

a1-peer-review_student.docx

The background is a solid black field populated with a variety of small, light green geometric shapes. These shapes include triangles of different sizes and orientations, wavy lines, and spiral patterns. They are scattered across the entire frame, creating a textured, abstract effect.

Crud

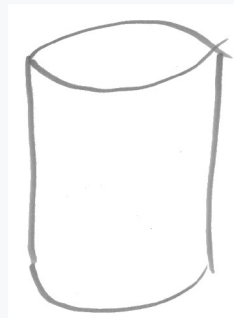
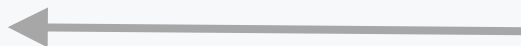
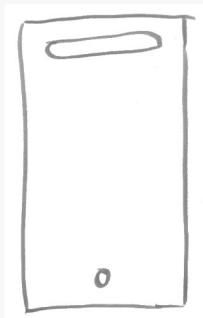


Source: [CRUD Operations Explained](#) by Avelon Pang, June 14, 2021

index.js

```
...  
const db = client.db(process.env.DB_NAME)  
  
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})  
    }  
  }  
}  
}  
  
...
```

find



index.js

...

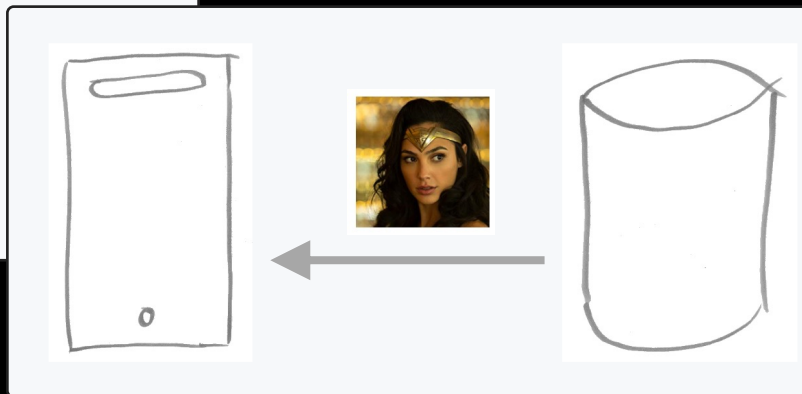
```
const db = client.db(process.env.DB_NAME)
```

```
function movie(req, res, next) {  
  db.collection('movie').findOne({  
    title: req.params.title  
  }, done)
```

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

...

findOne



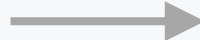
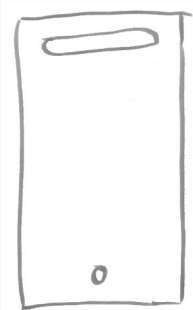
index.js

```
const { MongoClient, ObjectId } = require('mongodb')
const db = client.db(process.env.DB_NAME)

function update(req, res, next) {
  db.collection('movie').updateOne({
    _id: ObjectId(req.body.id),
    {$set: {plot: req.body.plot}}
  }, done)

  function done(err, data) {
    if (err) {
      next(err)
    } else {
      res.render('movie_updated.ejs', {title: req.body.title})
    }
  }
}
```

update



index.js

```
const { MongoClient, ObjectId } = require('mongodb')
const db = client.db(process.env.DB_NAME)
```

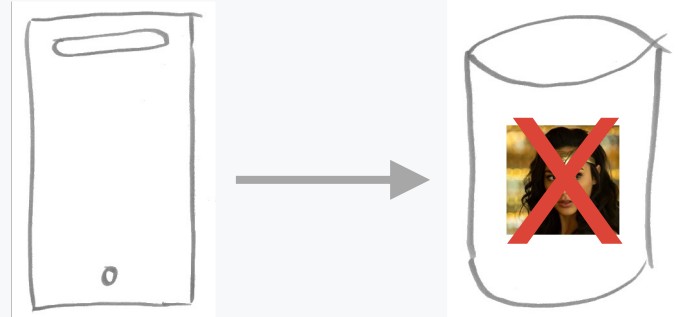
...

```
function remove(req, res, next) {
  db.collection('movie').deleteOne({
    _id: ObjectId(req.body.id)
  }, done)
```

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

...

delete

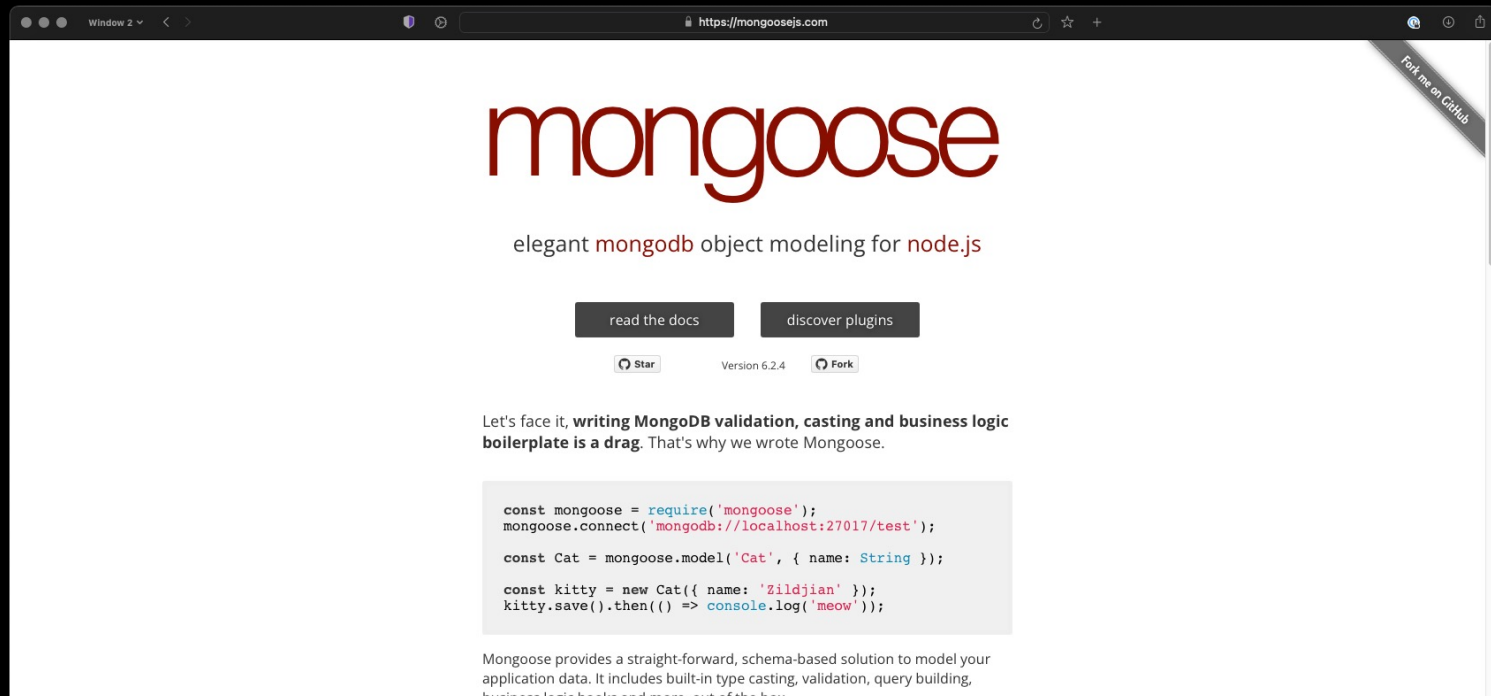


delete

```
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: don't just use find - to pass you'll also need
Update or insert or delete



Pick mongoDB (default driver) over Mongoose.

MongoDB.

Products

Solutions

Resources

Company

Pricing

Sign In

University Home » Browse Courses » M001

M001

MongoDB Basics

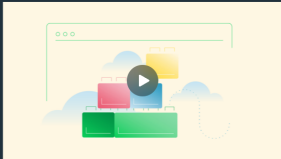
INTRODUCTORY

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!