

# Generic Data Project

Lester (Dishu) Lyu



# Stack



**M**

MongoDB



**E**

Express



**R**

React

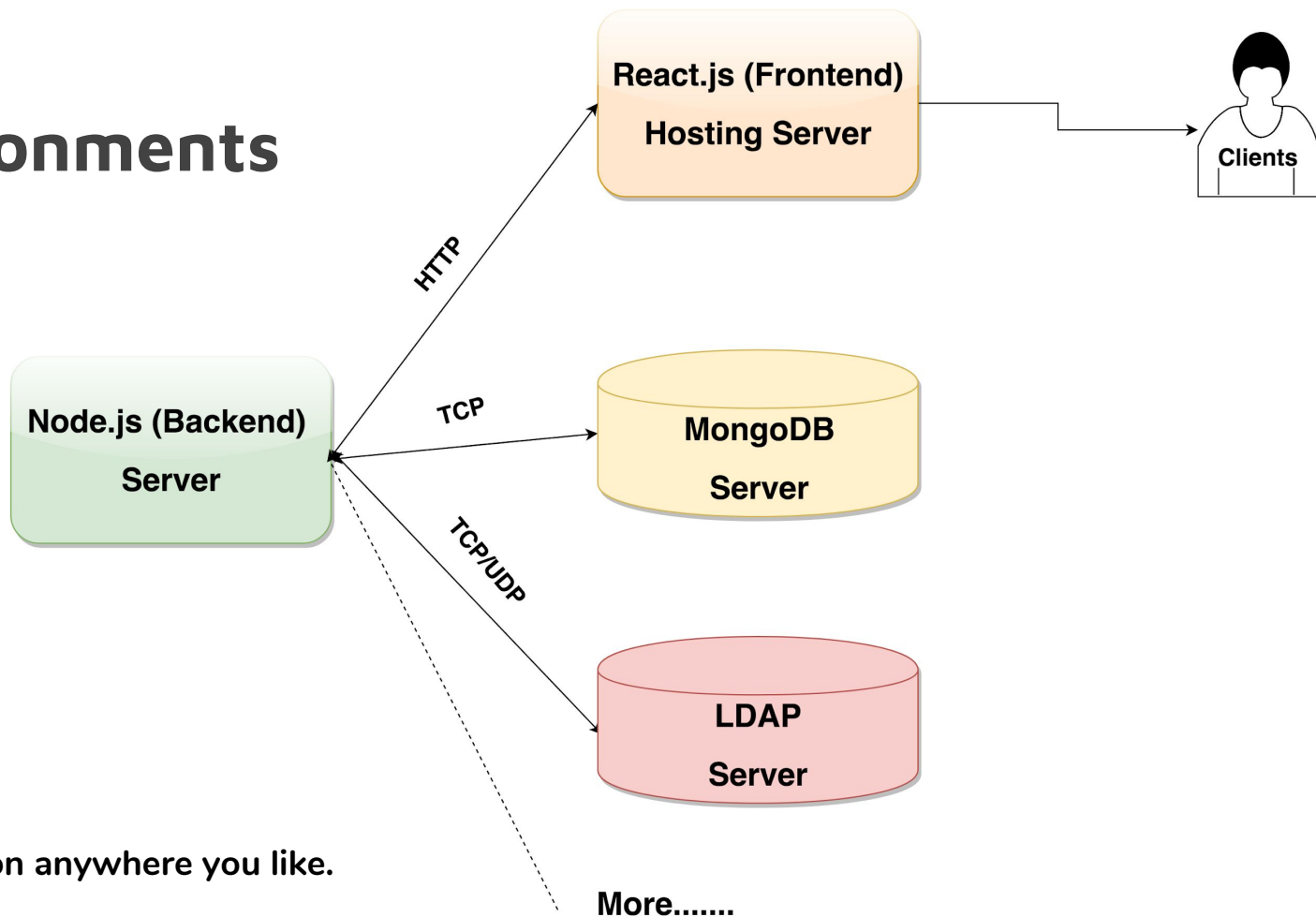


**N**

Node



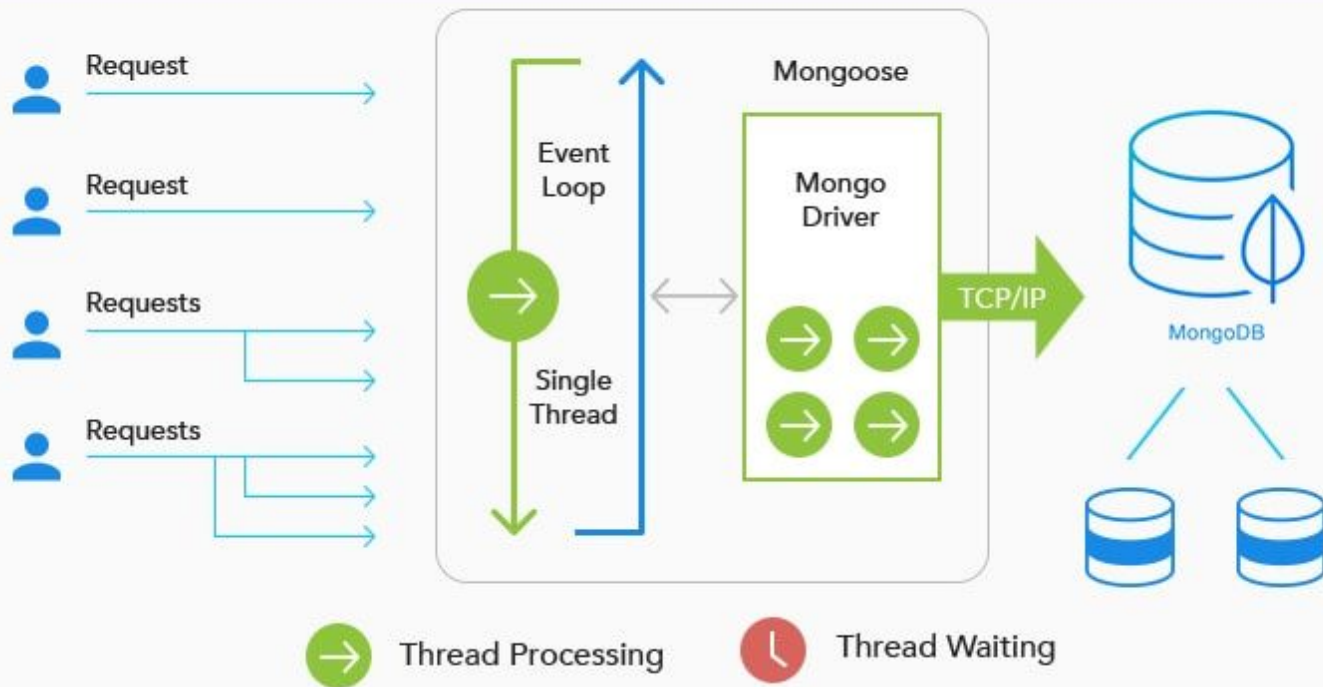
# Environments



Servers can be hosted on anywhere you like.

# Backend

Express.js





# Backend: Express Framework

```
var express = require('express')
```

```
var app = express()
```

```
app.get('/', function (req, res) {
```

```
  res.send('Hello World!')
```

```
})
```

```
app.listen(3000)
```

HTTP method (Get here)

Path (route)

The middleware function.

HTTP [response](#) argument to the middleware function, called "res" by convention.

HTTP [request](#) argument to the middleware function, called "req" by convention.



## Backend: Mongoose (MongoDB Driver)

```
const mongoose = require('mongoose');
```

Connect to MongoDB

```
mongoose.connect('mongodb://localhost:27017/test', {useNewUrlParser: true});
```

```
const Cat = mongoose.model('Cat', { name: String });
```

Define a model that contains one string property name.

```
const kitty = new Cat({ name: 'Zildjian' });
```

Create a new cat called Zildjian

```
kitty.save()
```

Save the new cat to MongoDB



# How do we save/load workbooks

- Save workbook template as a file in MongoDB. The template is used to store styles.
- Create a dictionary for each worksheet that maps category/attribute ID to row/column number. Store this dictionary in MongoDB.
- When query data from a workbook, you can simply give the workbook name, worksheet name, category ID and attribute ID, then we are able to do a fast query in MongoDB.
- When front-end loads a workbook, it loads the full workbook template file and the data previously filled by the user. The front-end will populate the data previously filled to the workbook and display the workbook.



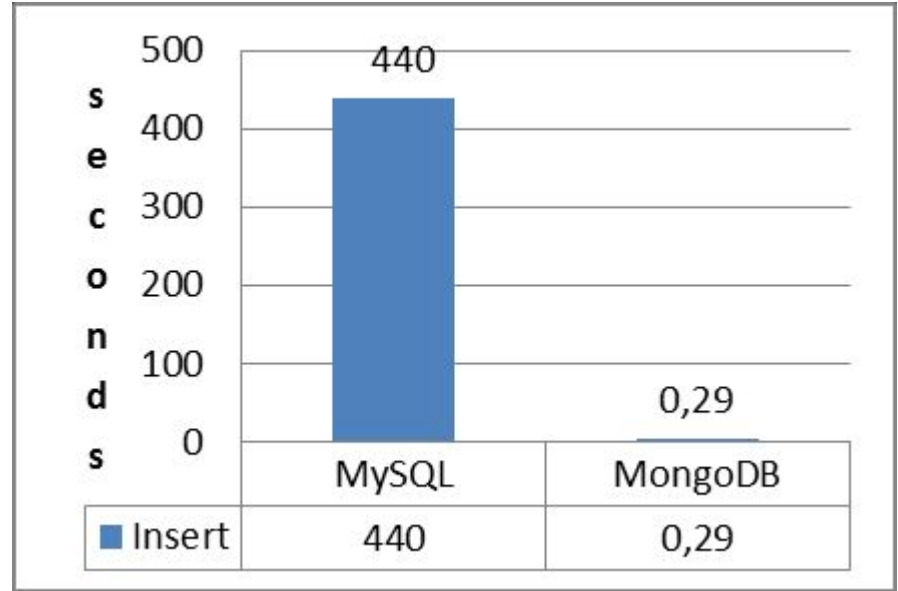
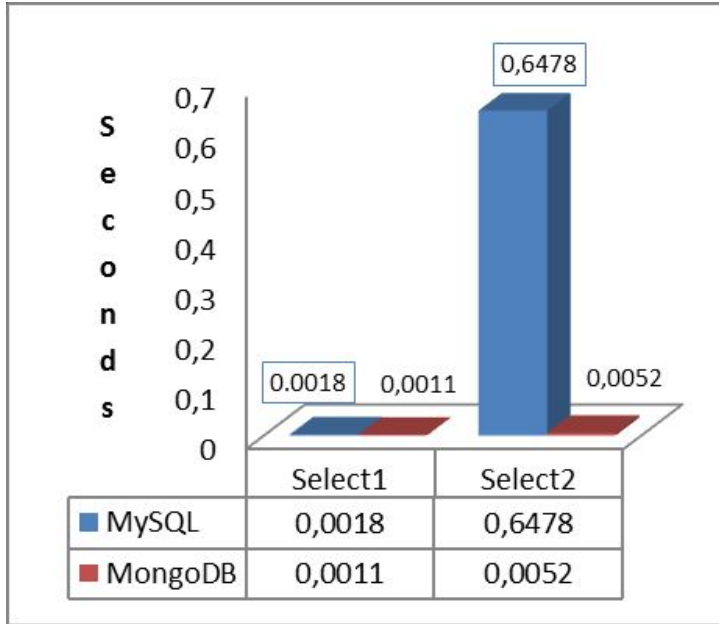
# Why MongoDB (nosql)?

- There are not much relations in this project.
- High performance when processing big data.
- Has much faster speed on handling large unstructured data.
- Easy to implement.
- Friendly to node.js .



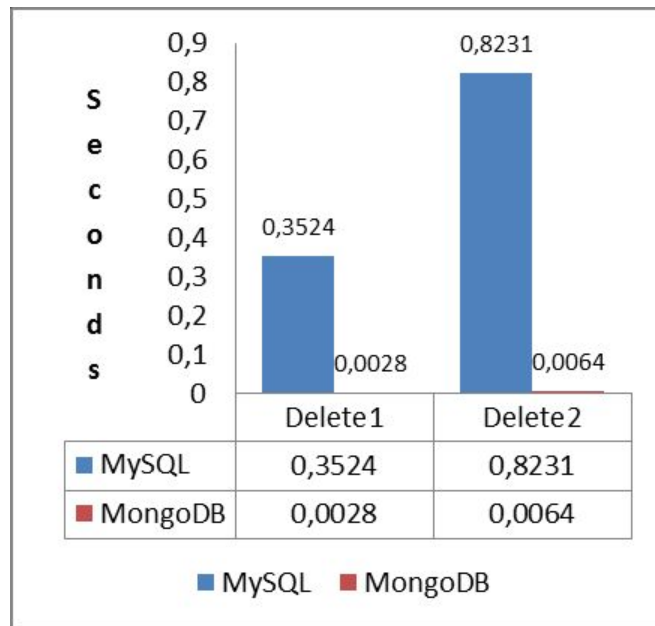
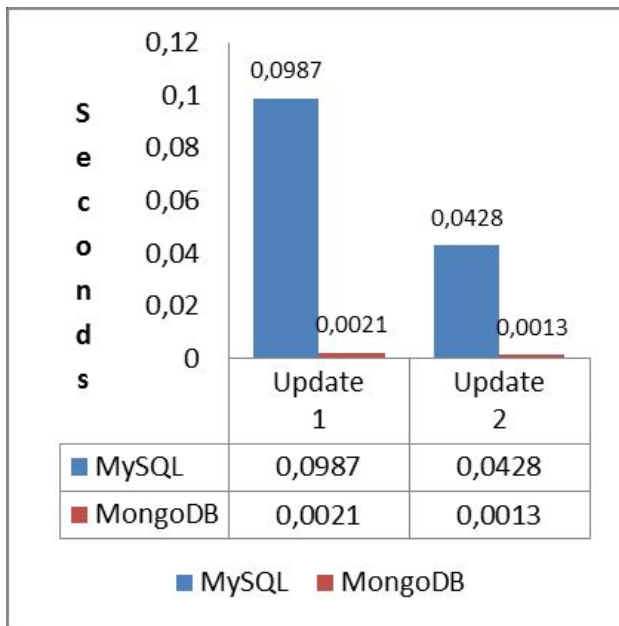


# MySQL vs MongoDB





# MySQL vs MongoDB





# Github Branches





# Github with Travis-CI



Commits on Mar 12, 2019

Merge branch 'richtext' into beta

 LesterLyu committed 22 hours ago ✓



a86ba97



merge sheetview-pane branch and latest change from parent/master

 LesterLyu committed 22 hours ago ✓



1ed861f



remove default value in docs.

 LesterLyu committed a day ago



6e386bf



support freezes panes and splits panes. (tests & docs added)

 LesterLyu committed a day ago ✓



05271e9



Commits on Mar 11, 2019

Updates docs.

 LesterLyu committed 2 days ago ✗



666f4ba



Eliminates cell reference when initiating RichText; ...

 LesterLyu committed 2 days ago ✓



243b4c1





# Github with Travis-CI

- Every time you push your changes to Github, Travis-CI will be invoked to run all tests.

## How does CI work?

1. Read your configuration file from Github. i.e. [.travis.yml](#).
2. Creates a virtual machine and boot it up.
3. Clone your Github repository.
4. Run the scripts defined in [.travis.yml](#).
5. If the scripts have exit code = 0, which means success, Travis-CI tells Github the build pass.

# What is in .travis.yml.

```
language: node_js
```

← The programming language.

```
node_js:
```

- "stable"
- "10"
- "9"
- "8"
- "7"
- "6"

} The node js versions you want to use.

```
script:
```

- npm test
- npm run e2e-parse

} What bash commands you want the CI execute?

