



mongoDB®

# MongoDB: Select and Update

SENG 4640

Software Engineering for Web Apps

Winter 2023

Sina Keshvadi

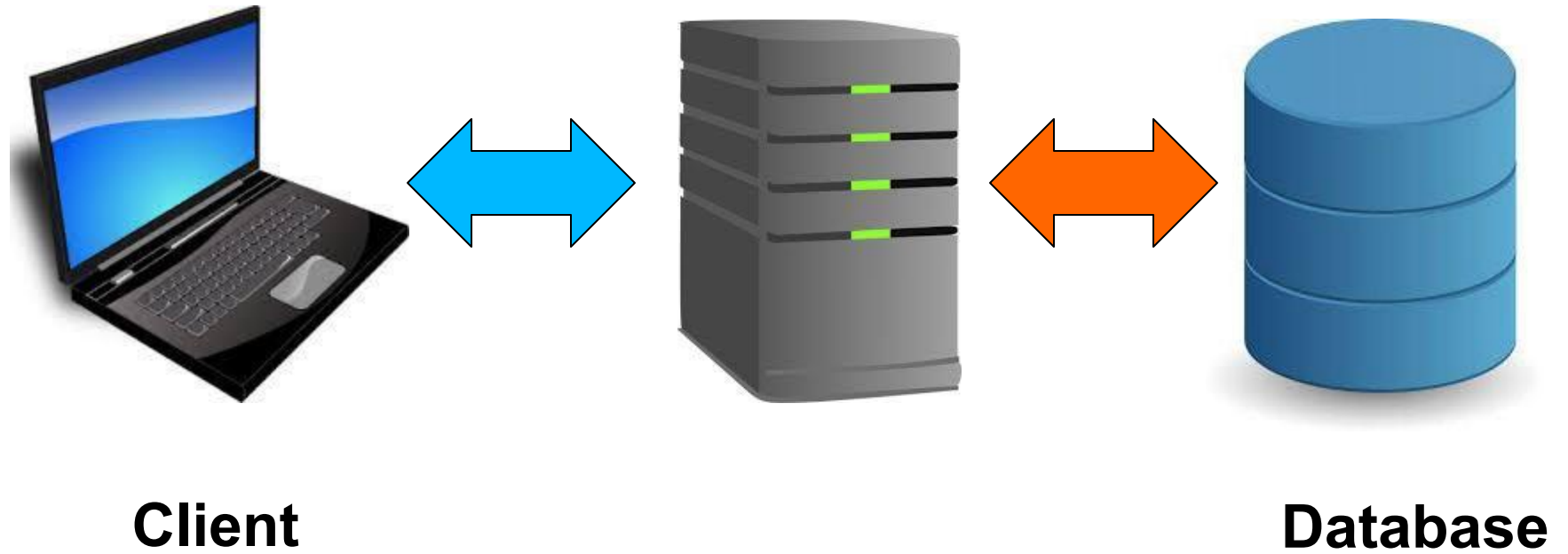
Thompson Rivers University

# Review

---

- **MongoDB** is a NoSQL Database that is designed for use with JavaScript apps
- MongoDB stores **collections** of **documents**
- We can access MongoDB from our Node/Express app using libraries such as Mongoose
- We define a **Schema** and then can create new documents using the **save** function

**Server**



Name:

Age:

Submit form!

Name: Alexis

Age: 17

Submit form!

Successfully created new person:

**Name:** Alexis

**Age:** 17

[Create New Person](#)

[Show All](#)

Successfully created new person:

**Name:** Alexis

**Age:** 17

[Create New Person](#)

[Show All](#)

Here are all the people:

- [Alexis](#): 17
- [Olivier](#): 12
- [Mesut](#): 11

[Create New Person](#)



Here are all the people:

- [Alexis](#): 17
- [Olivier](#): 12
- [Mesut](#): 11

[Create New Person](#)

```
/* This is index.js */
... // previous Example

app.use("/all", async (req, res) => {
  try {
    const allPeople = await Person.find();
    if (allPeople.length === 0) {
      res.status(200).send("There are no people");
    } else {
      res.render("showAll", { people: allPeople });
    }
  } catch (err) {
    res.status(500).send("Error: " + err);
  }
});

... // previous Example
```



```
/* This is index.js */
... // previous Example

app.use("/all", async (req, res) => {
  try {
    const allPeople = await Person.find();
    if (allPeople.length === 0) {
      res.status(200).send("There are no people");
    } else {
      res.render("showAll", { people: allPeople });
    }
  } catch (err) {
    res.status(500).send("Error: " + err);
  }
});

... // previous Example
```



Here are all the people:

- [Alexis](#): 17
- [Olivier](#): 12
- [Mesut](#): 11

[Create New Person](#)

```
<!-- This is views/showAll.ejs -->
```

Here are all the people:

```
<ul>
```

```
  <% people.forEach( (person)=> { %>
```

```
    <li><a href="/person?name=<%= person.name %>">
```

```
      <%= person.name %>
```

```
    </a>:
```

```
    <%= person.age %>
```

```
  </li>
```

```
  <% }); %>
```

```
</ul>
```

```
<br>
```

```
<a href='/public/personform.html'>Create New Person</a>
```

```
<!-- This is views/showAll.ejs -->
```

Here are all the people:

```
<ul>
```

```
  <% people.forEach( (person)=> { %>
```

```
    <li><a href="/person?name=<%= person.name %>">
```

```
      <%= person.name %>
```

```
    </a>:
```

```
    <%= person.age %>
```

```
  </li>
```

```
  <% }); %>
```

```
</ul>
```

```
<br>
```

```
<a href='/public/personform.html'>Create New Person</a>
```

Here are all the people:

- [Alexis](#): 17
- [Olivier](#): 12
- [Mesut](#): 11

[Create New Person](#)

Here are all the people:

- [Alexis](#): 17
- [Olivier](#): 12
- [Mesut](#): 11

[Create New Person](#)



Name: Alexis

Age:

[Create New Person](#)

[Show All](#)

Name: Alexis

Age:

[Create New Person](#)

[Show All](#)

Name: Alexis

Age:

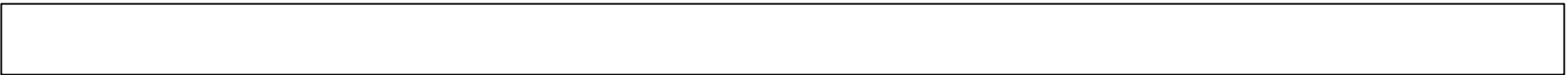
Update

[Create New Person](#)

[Show All](#)

```
/* This is index.js */
... // previous Example

app.use("/person", async (req, res) => {
  try {
    const searchName = req.query.name;
    const person = await Person.findOne({ name: searchName });
    if (!person) {
      res.type("html").status(200);
      res.send("No person named " + searchName);
    } else {
      res.render("personInfo", { person: person });
    }
  } catch (err) {
    res.type("html").status(500);
    res.send("Error: " + err);
  }
});
... // previous Example
```



Name: Alexis

Age:

[Create New Person](#)

[Show All](#)

```
<!-- This is views/personInfo.ejs -->
<form action='/update' method='post'>
  Name: <%= person.name %><br>
  <input name='username' value='<%= person.name %>' hidden>
  Age: <input name='age' value='<%= person.age %>'>
  <input type='submit' value='Update'>
</form>
<br><a href='/public/personform.html'>Create New Person</a>
<br><a href='/all'>Show All</a>
```



Name: Alexis

Age:

[Create New Person](#)

[Show All](#)

Name: Alexis

Age:

[Create New Person](#)  
[Show All](#)



Name: Alexis

Age: 18

Update

[Create New Person](#)

[Show All](#)

Updated Alexis's age to 18

[Create New Person](#)

[Show All](#)

Updated Alexis's age to 18

[Create New Person](#)

[Show All](#)

```
/* This is index.js */
... // previous Example
app.use("/update", async (req, res) => {
  const updateName = req.body.username;
  try {
    const person = await Person.findOne({ name: updateName });
    if (!person) {
      res.type("html").status(200);
      res.send("No person named " + updateName);
    } else {
      person.age = req.body.age;
      await person.save();
      res.render("updated", { person: person });
    }
  } catch (err) {
    res.type("html").status(500);
    res.send("Error: " + err);
  }
});
... // previous Example
```

Updated Alexis's age to 18

[Create New Person](#)

[Show All](#)

```
<!-- This is views/updated.ejs -->
```

```
Updated <%= person.name %>'s age to <%= person.age %>
```

```
<br><a href='/public/personform.html'>Create New Person</a>
```

```
<br><a href='/all'>Show All</a>
```

# Summary

---

- We can access MongoDB from our Node/Express app using libraries such as Mongoose
- We can use the **find** function to select all documents in a collection, or pass a query object to select only certain ones
- Once we have a document, we can update it using the **save** function