



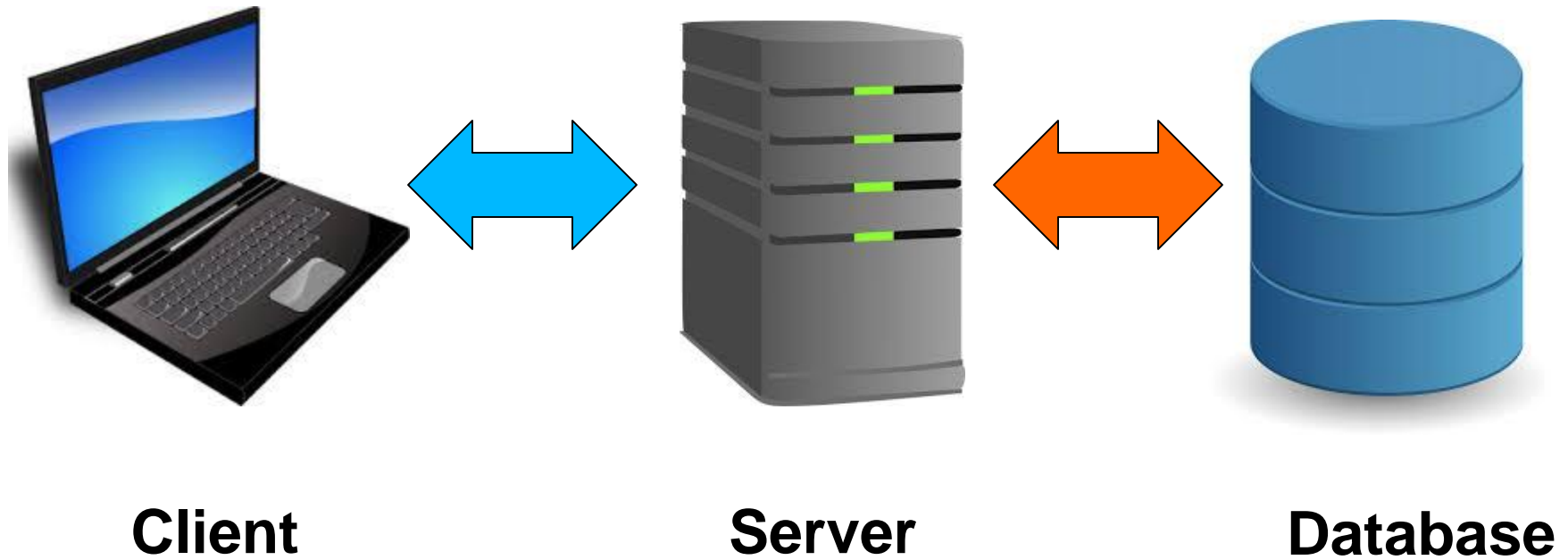
Video 4.7

MongoDB: Select and Update

Chris Murphy

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





Name:

Age:

Submit form!

Name:

Age:

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](#)

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  } );
});
```

```
/* This is Person.js */

var mongoose = require('mongoose');

// note: your host/port number may be different!
mongoose.connect('mongodb://localhost:27017/myDatabase');

var Schema = mongoose.Schema;

var personSchema = new Schema( {
  name: {type: String, required: true, unique: true},
  age: Number
} );

module.exports = mongoose.model('Person', personSchema);
```

```
/* This is Person.js */

var mongoose = require('mongoose');

// note: your host/port number may be different!
mongoose.connect('mongodb://localhost:27017/myDatabase');

var Schema = mongoose.Schema;

var personSchema = new Schema( {
  name: {type: String, required: true, unique: true},
  age: Number
} );

module.exports = mongoose.model('Person', personSchema);
```

```
/* This is Person.js */  
  
var mongoose = require('mongoose');  
  
// note: your host/port number may be different!  
mongoose.connect('mongodb://localhost:27017/myDatabase');  
  
var Schema = mongoose.Schema;  
  
var personSchema = new Schema( {  
    name: {type: String, required: true, unique: true},  
    age: Number  
} );  
  
module.exports = mongoose.model('Person', personSchema);
```

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  } );
});
```



```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  }));
});
```

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```

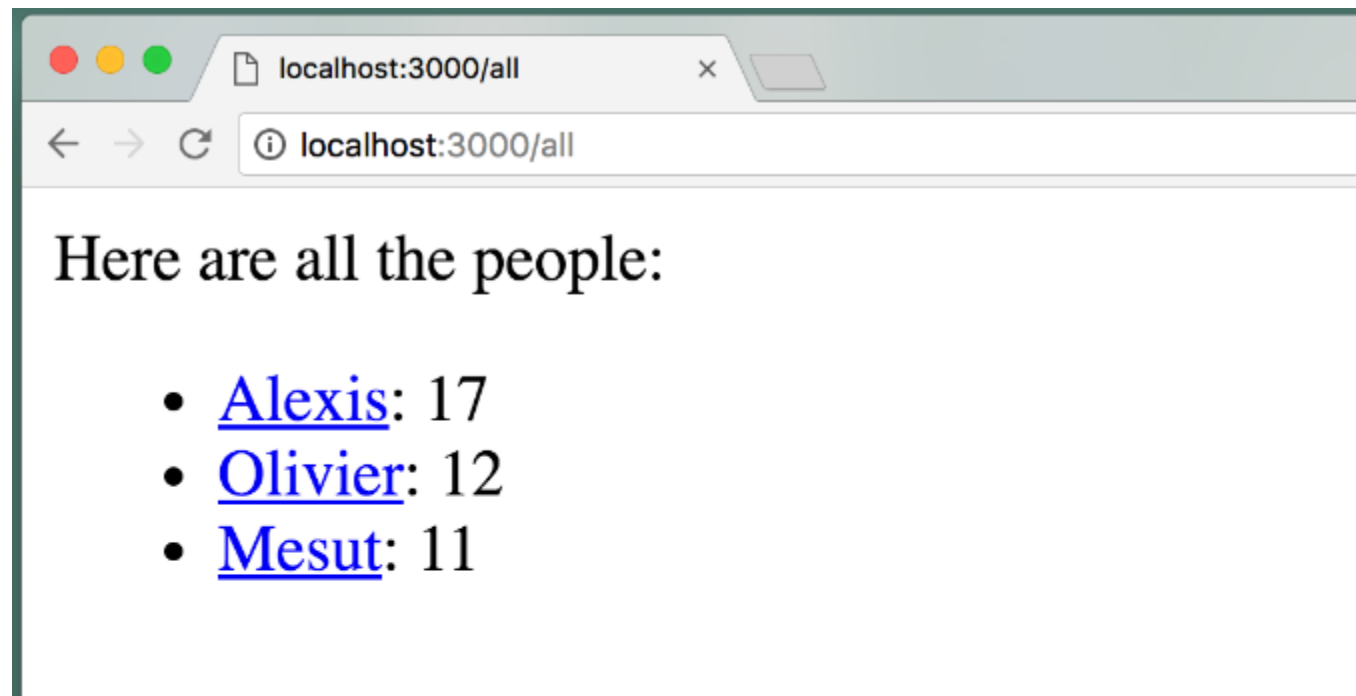
```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```

```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```

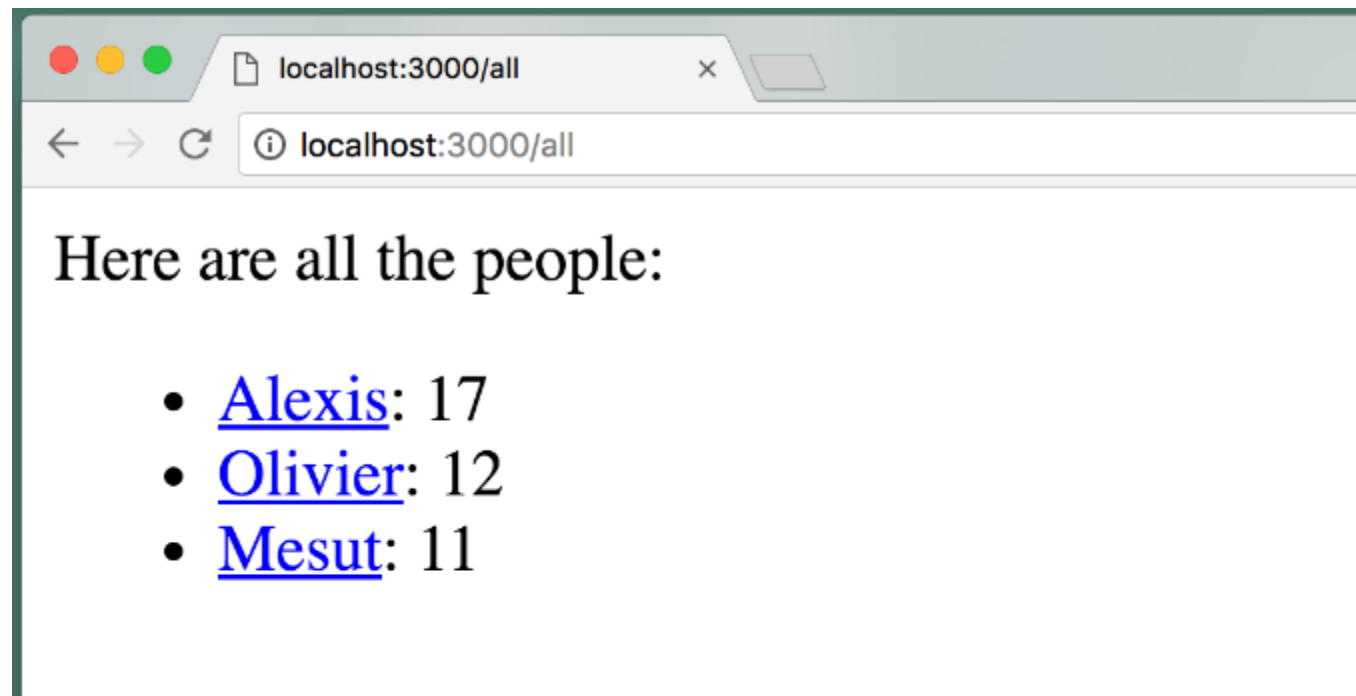
```
app.use('/all', (req, res) => {
  Person.find( (err, allPeople) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (allPeople.length == 0) {
      res.type('html').status(200);
      res.send('There are no people');
    }
    else {
      res.render('showAll', { people: allPeople });
    }
  });
});
```



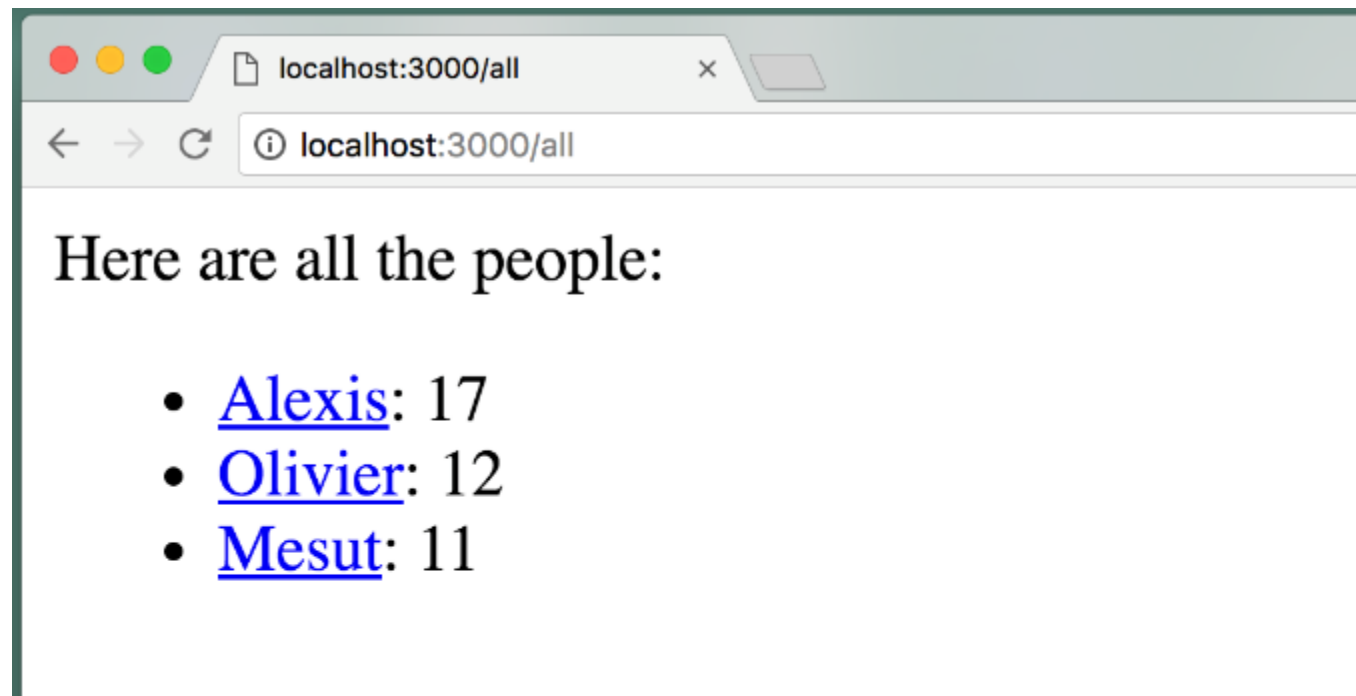
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



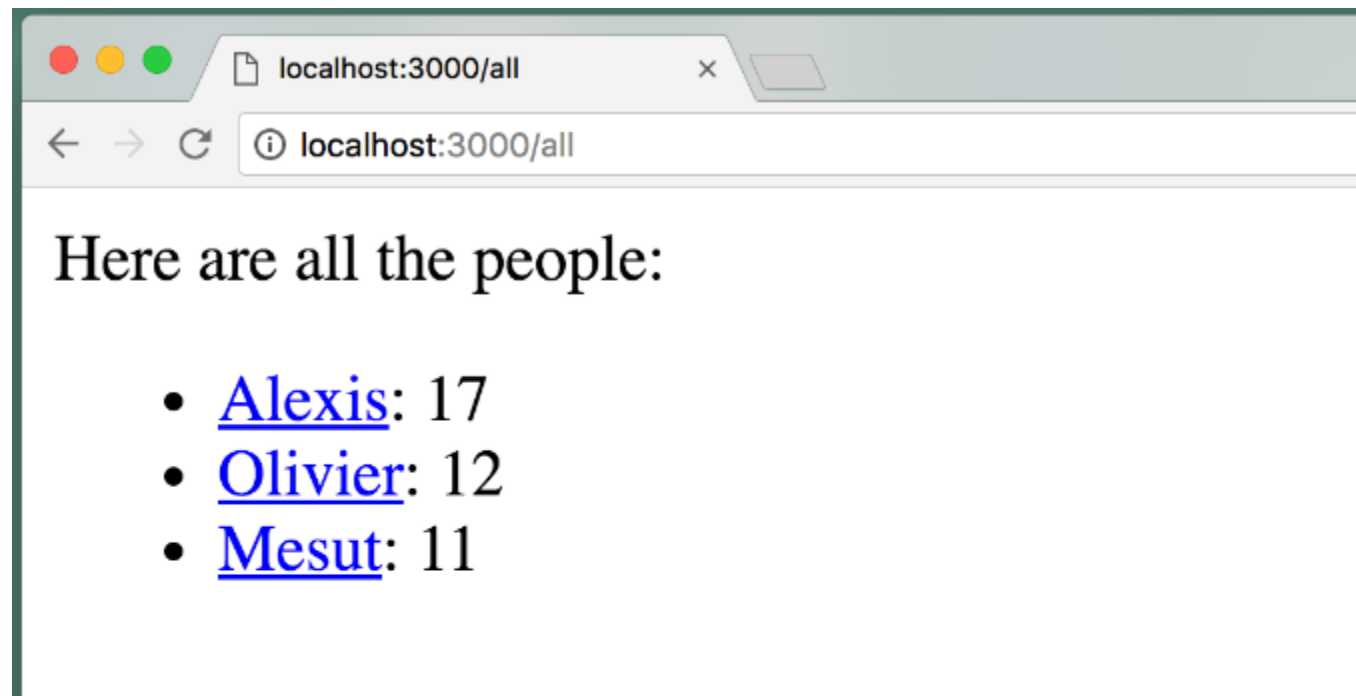
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



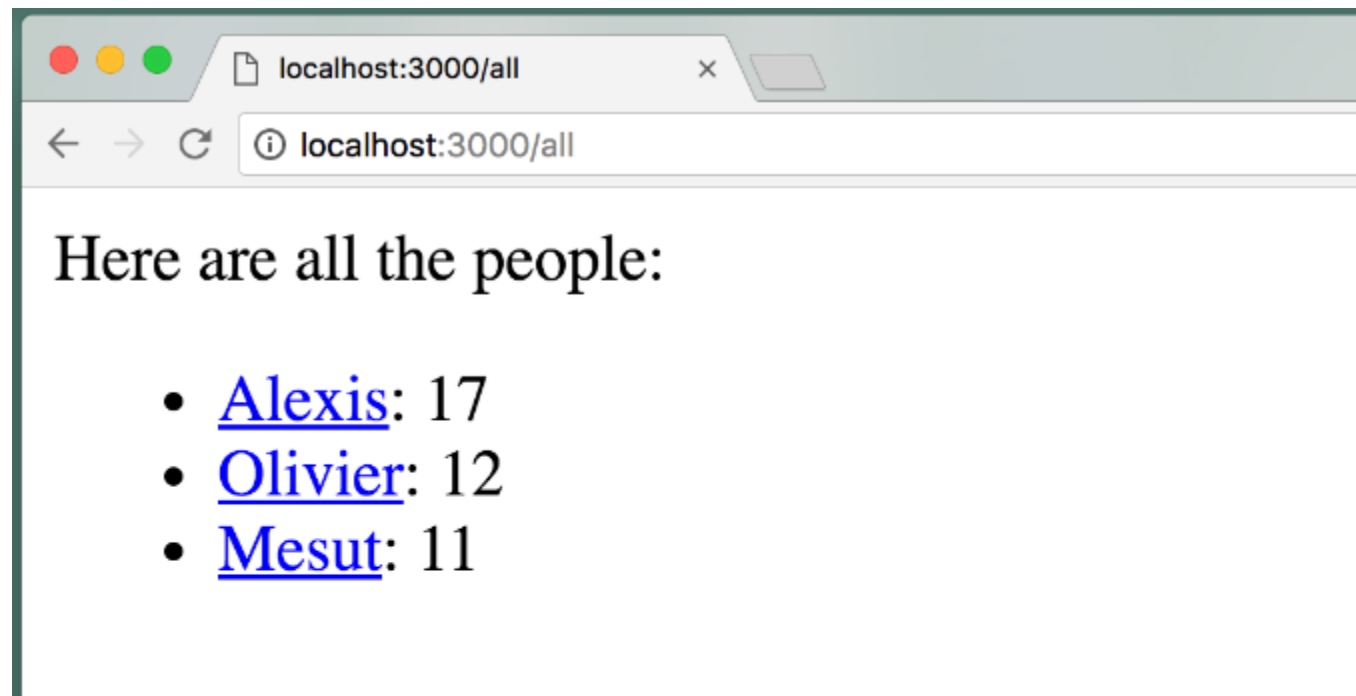
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



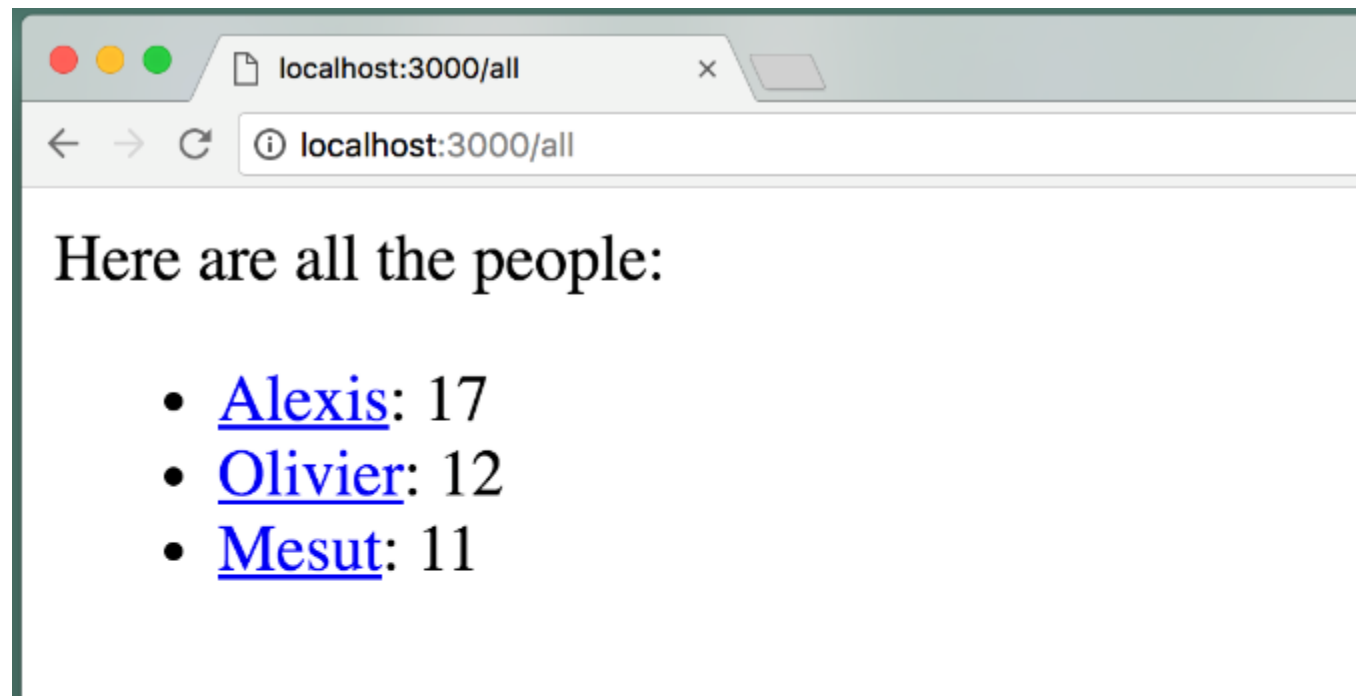
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



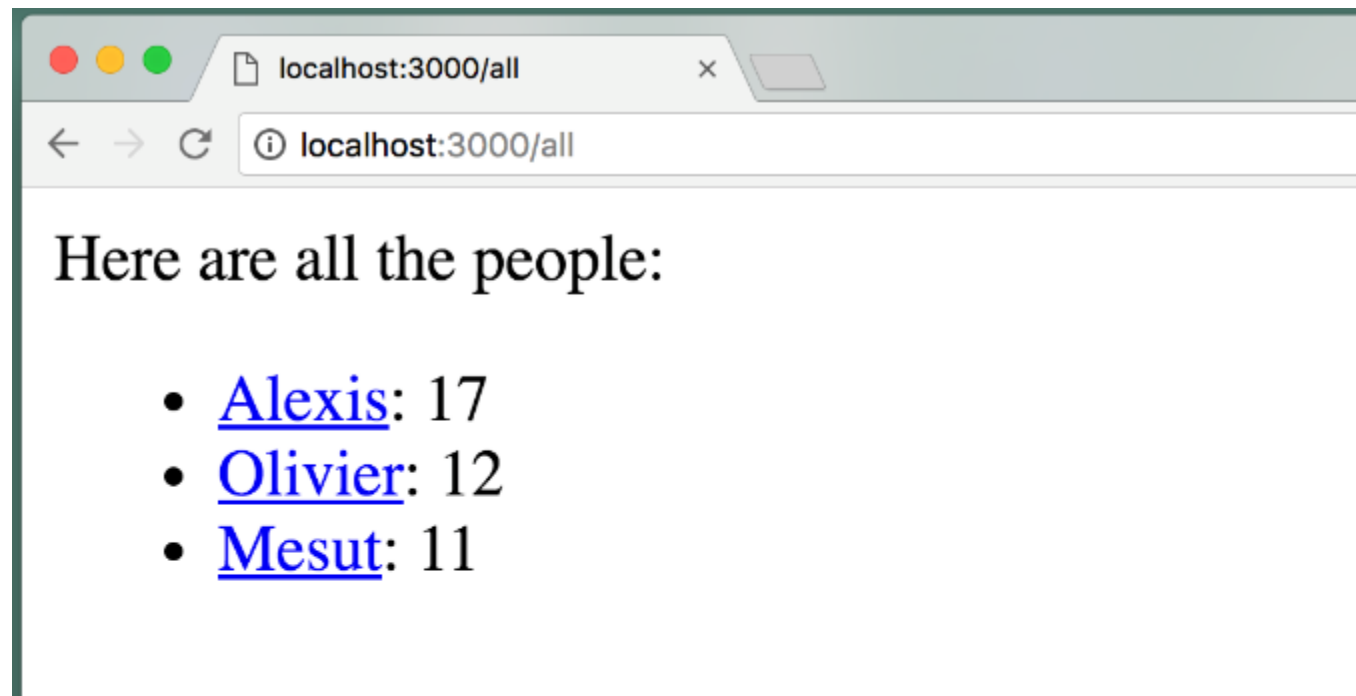
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



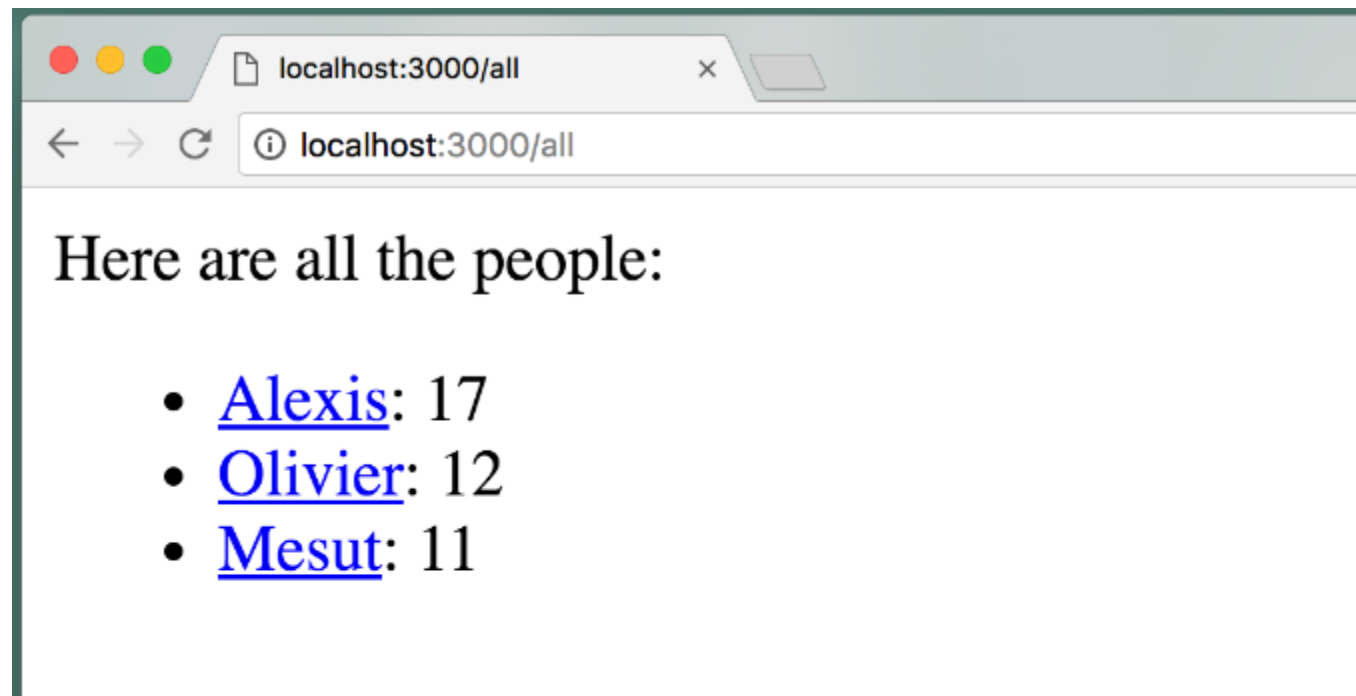
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
    <li><a href="/person?name=<%= person.name %>">
        <%= person.name %></a>:
        <%= person.age %>
    </li>
<% } ) ; %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



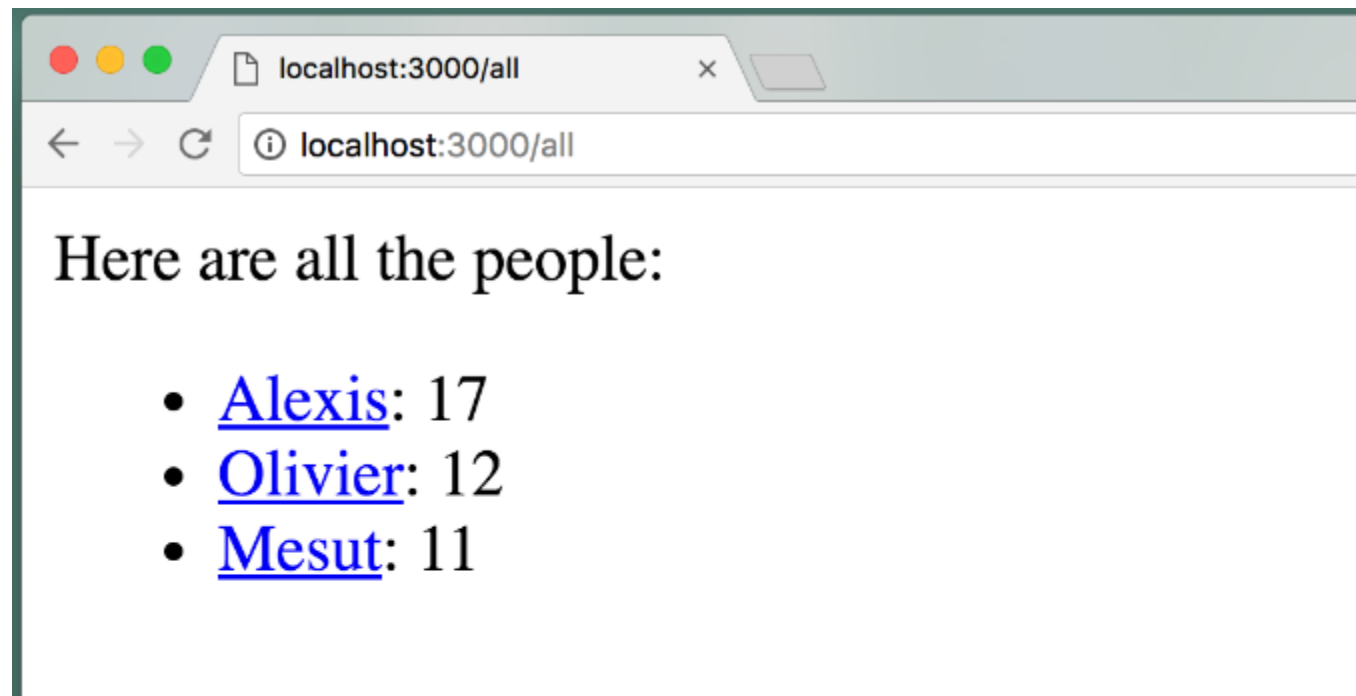
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
    <li><a href="/person?name=<%= person.name %>">
        <%= person.name %></a>:
        <%= person.age %>
    </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



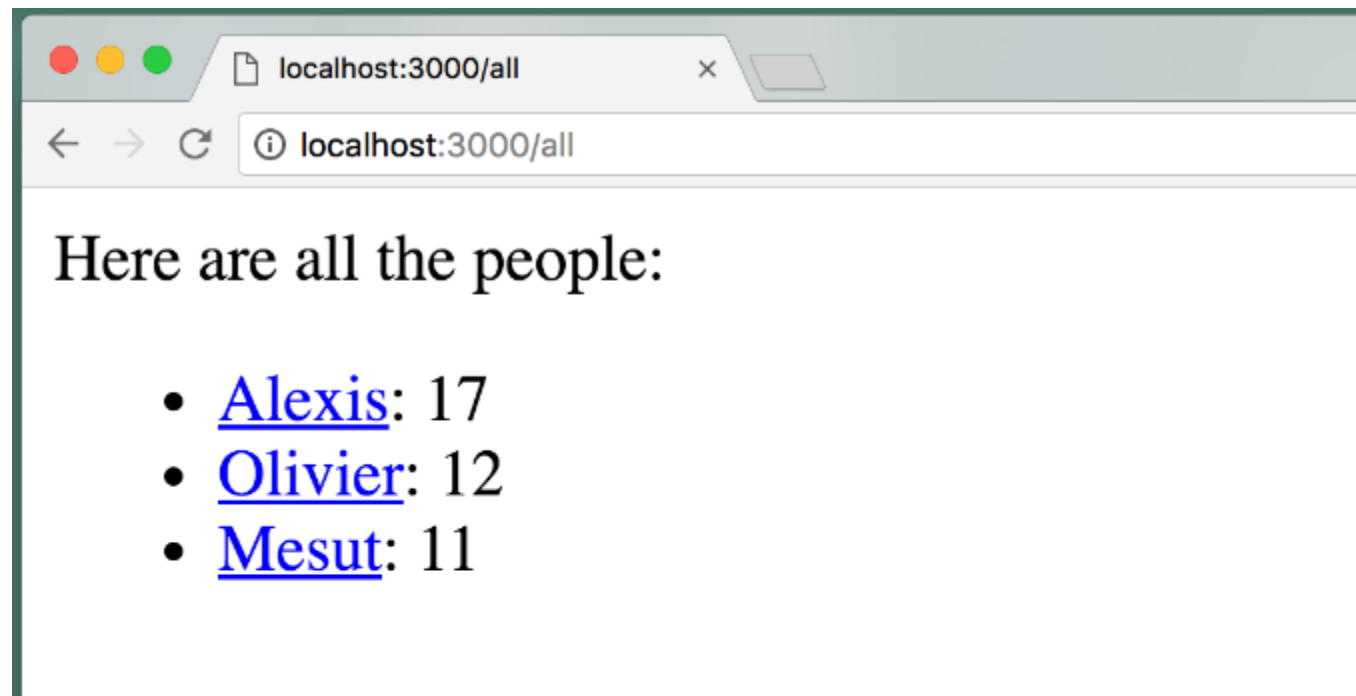
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



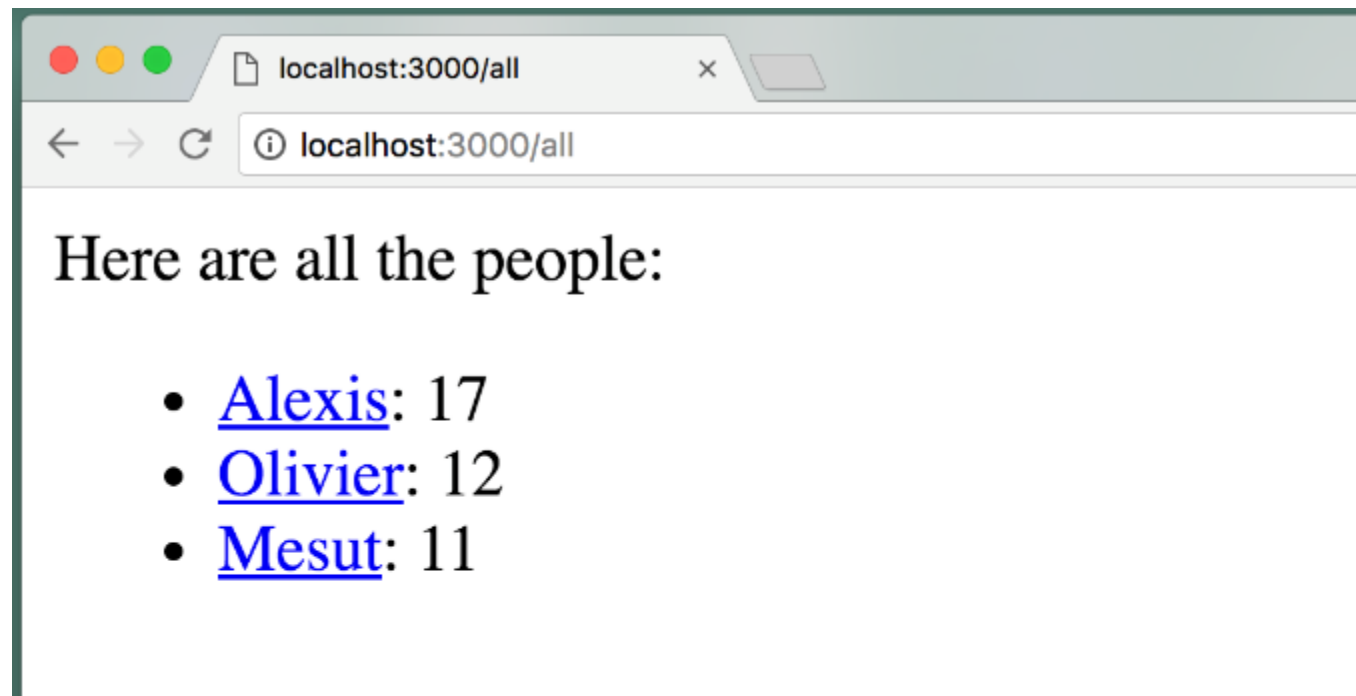

```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



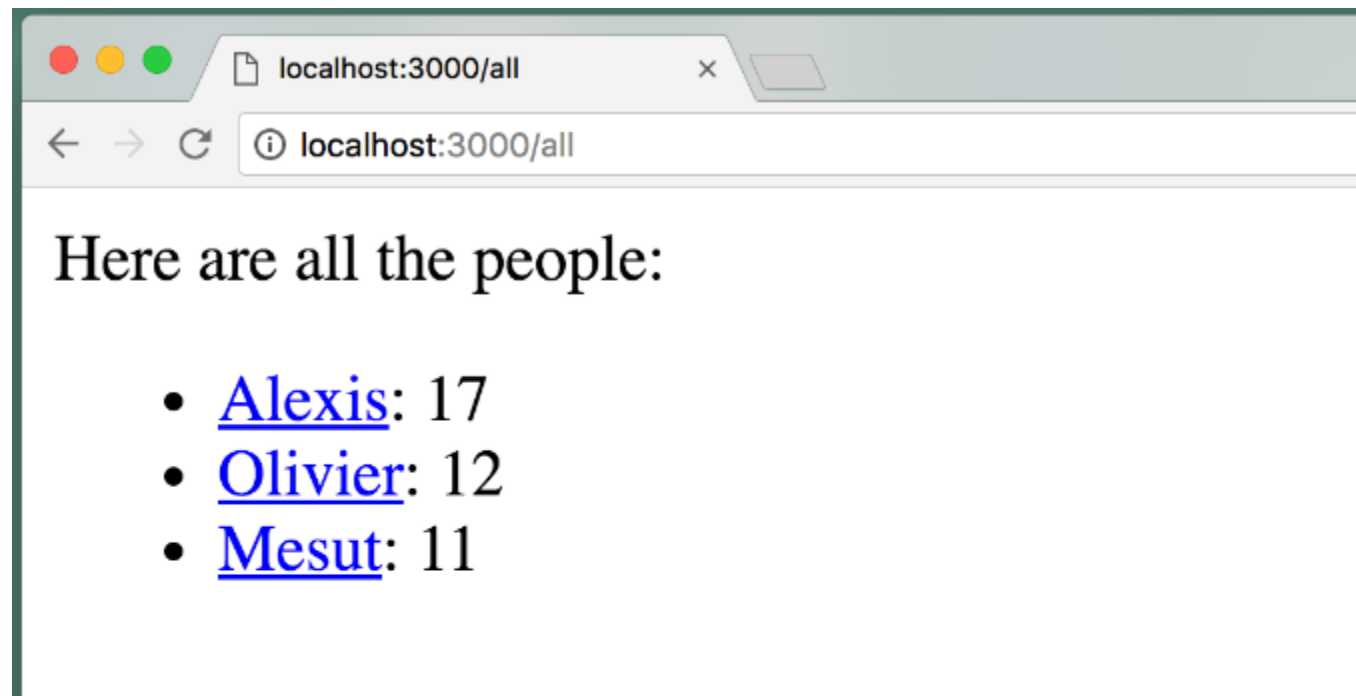
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href="/public/personform.html">Create New Person</a>
```



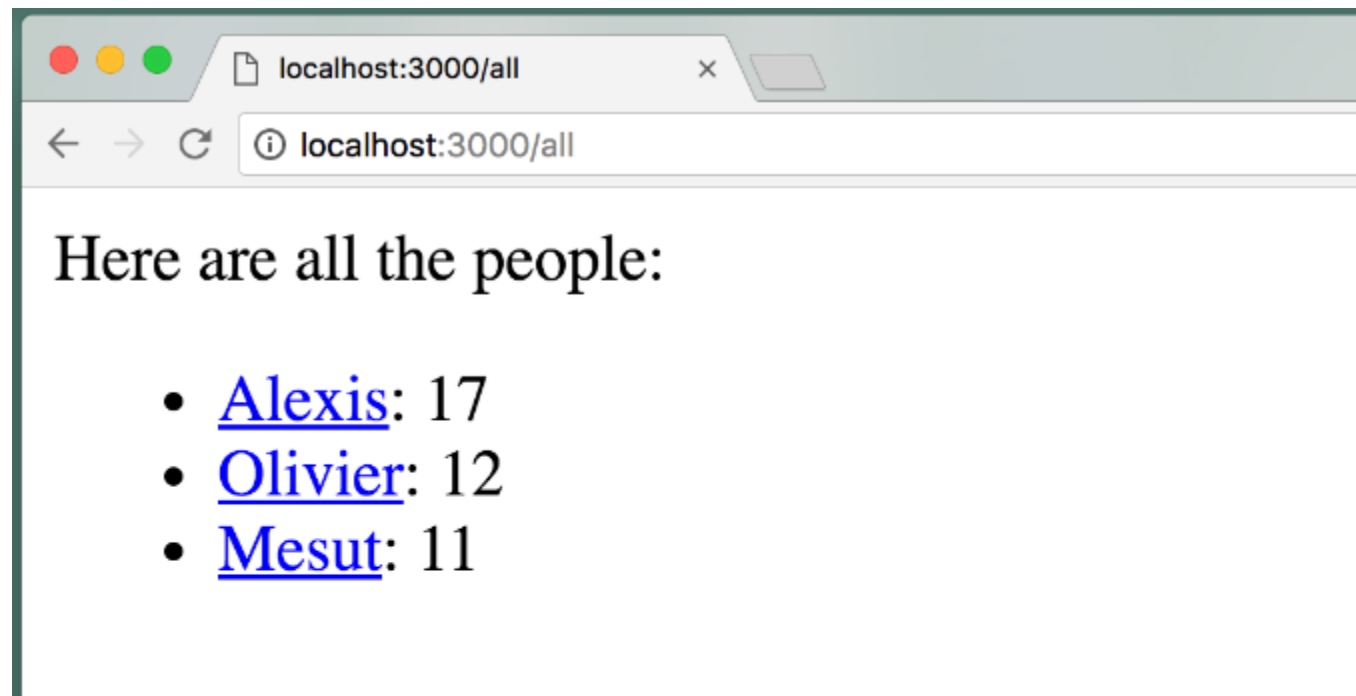
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href="/public/personform.html">Create New Person</a>
```



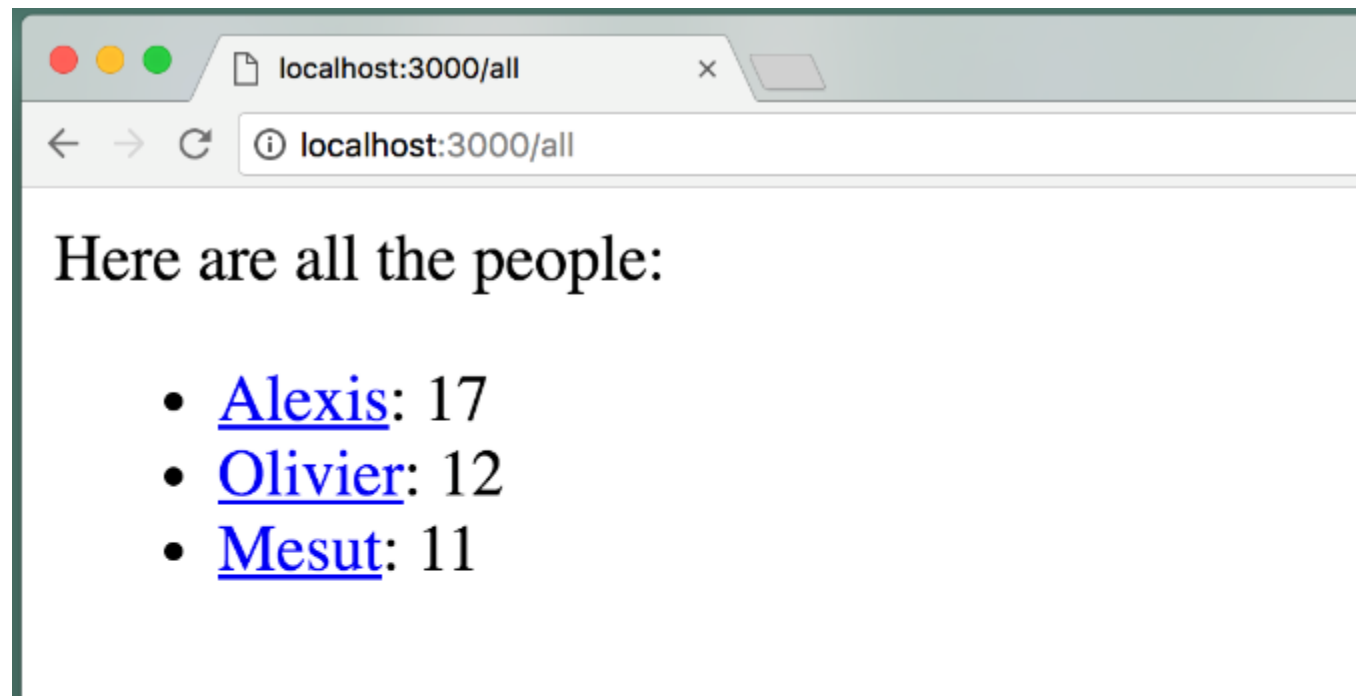
```
<!-- This is views/showAll.ejs -->
Here are all the people:
<ul>
<% people.forEach( (person) => { %>
  <li><a href="/person?name=<%= person.name %>">
    <%= person.name %></a>:
    <%= person.age %>
  </li>
<% }); %>
</ul>
<br><a href='/public/personform.html'>Create New Person</a>
```



```
<!-- 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](#)



localhost:3000/person?name= Alexis

Guest



localhost:3000/person?name=Alexis



Name: Alexis

Age: 17

Update

[Create New Person](#)

[Show All](#)

Name: Alexis

Age:

[Create New Person](#)

[Show All](#)

Name: Alexis

Age:

[Create New Person](#)

[Show All](#)

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {  
  var searchName = req.query.name;  
  Person.findOne( { name: searchName }, (err, person) => {  
    if (err) {  
      res.type('html').status(500);  
      res.send('Error: ' + err);  
    }  
    else if (!person) {  
      res.type('html').status(200);  
      res.send('No person named ' + searchName);  
    }  
    else {  
      res.render('personInfo', { person : person });  
    }  
  });  
});
```

```
app.use('/person', (req, res) => {  
  var searchName = req.query.name;  
  Person.findOne( { name: searchName }, (err, person) => {  
    if (err) {  
      res.type('html').status(500);  
      res.send('Error: ' + err);  
    }  
    else if (!person) {  
      res.type('html').status(200);  
      res.send('No person named ' + searchName);  
    }  
    else {  
      res.render('personInfo', { person : person });  
    }  
  });  
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  } );
});
```



```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {  
  var searchName = req.query.name;  
  Person.findOne( { name: searchName }, (err, person) => {  
    if (err) {  
      res.type('html').status(500);  
      res.send('Error: ' + err);  
    }  
    else if (!person) {  
      res.type('html').status(200);  
      res.send('No person named ' + searchName);  
    }  
    else {  
      res.render('personInfo', { person : person });  
    }  
  });  
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

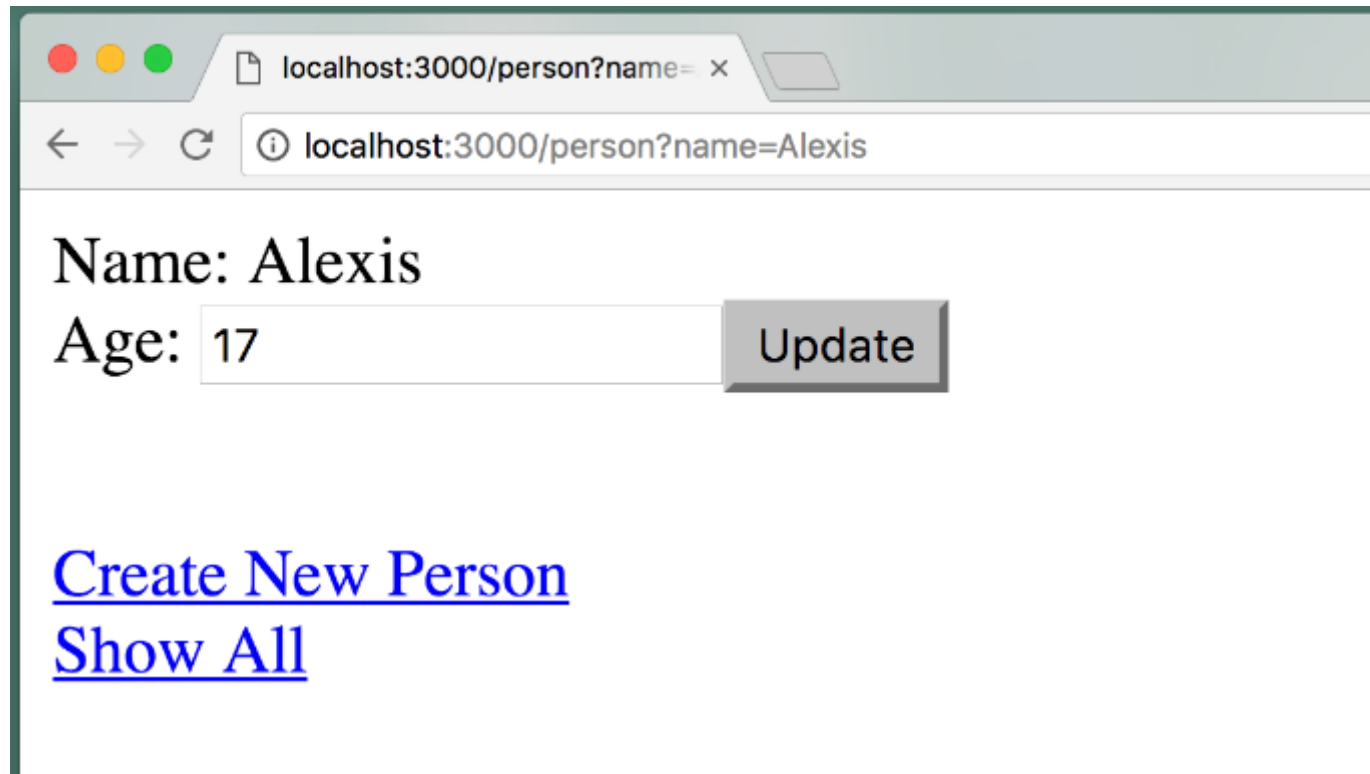


```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
app.use('/person', (req, res) => {
  var searchName = req.query.name;
  Person.findOne( { name: searchName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + searchName);
    }
    else {
      res.render('personInfo', { person : person });
    }
  });
});
```

```
<!-- 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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' in a text input field. To the right of the input field is a grey 'Update' button. Below the form, there are two blue, underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

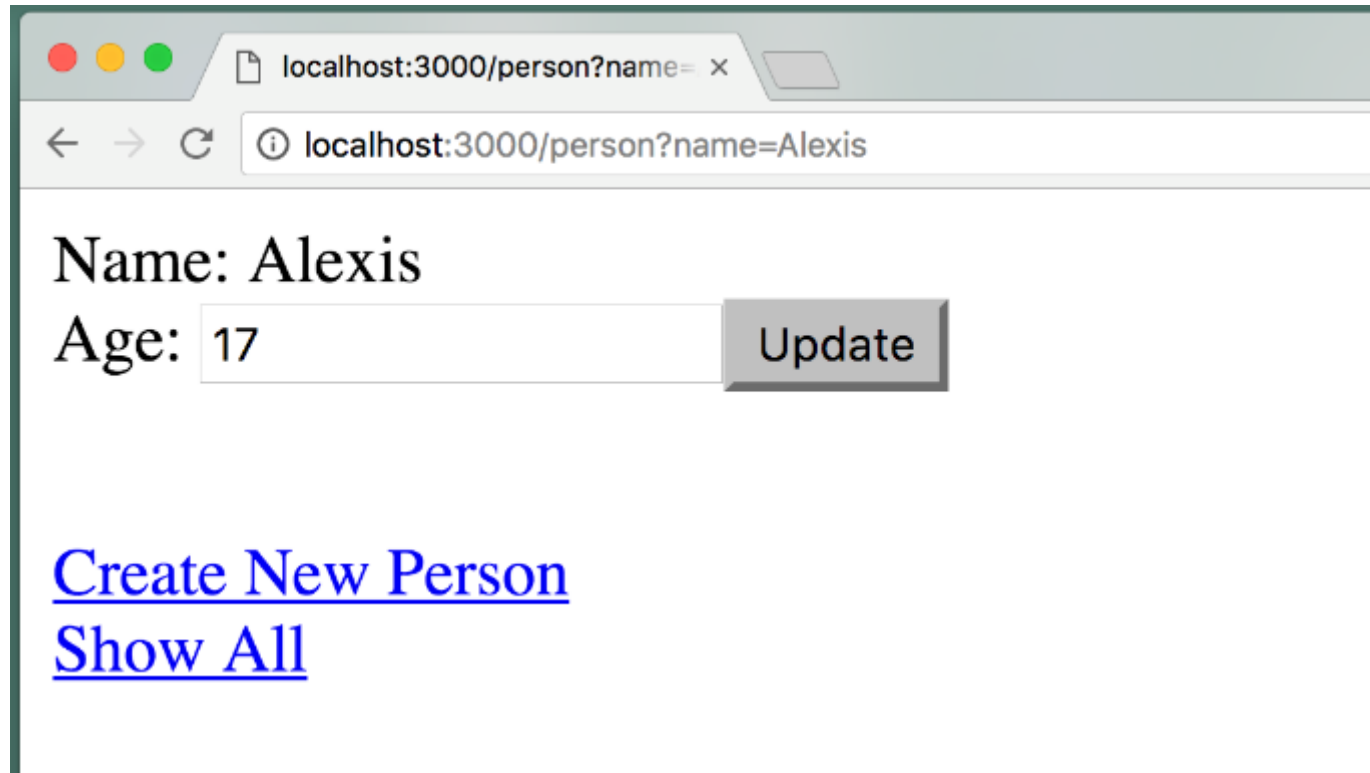
Name: Alexis

Age: 17

[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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' in a text input field. To the right of the input field is a grey 'Update' button. Below the form, there are two blue, underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

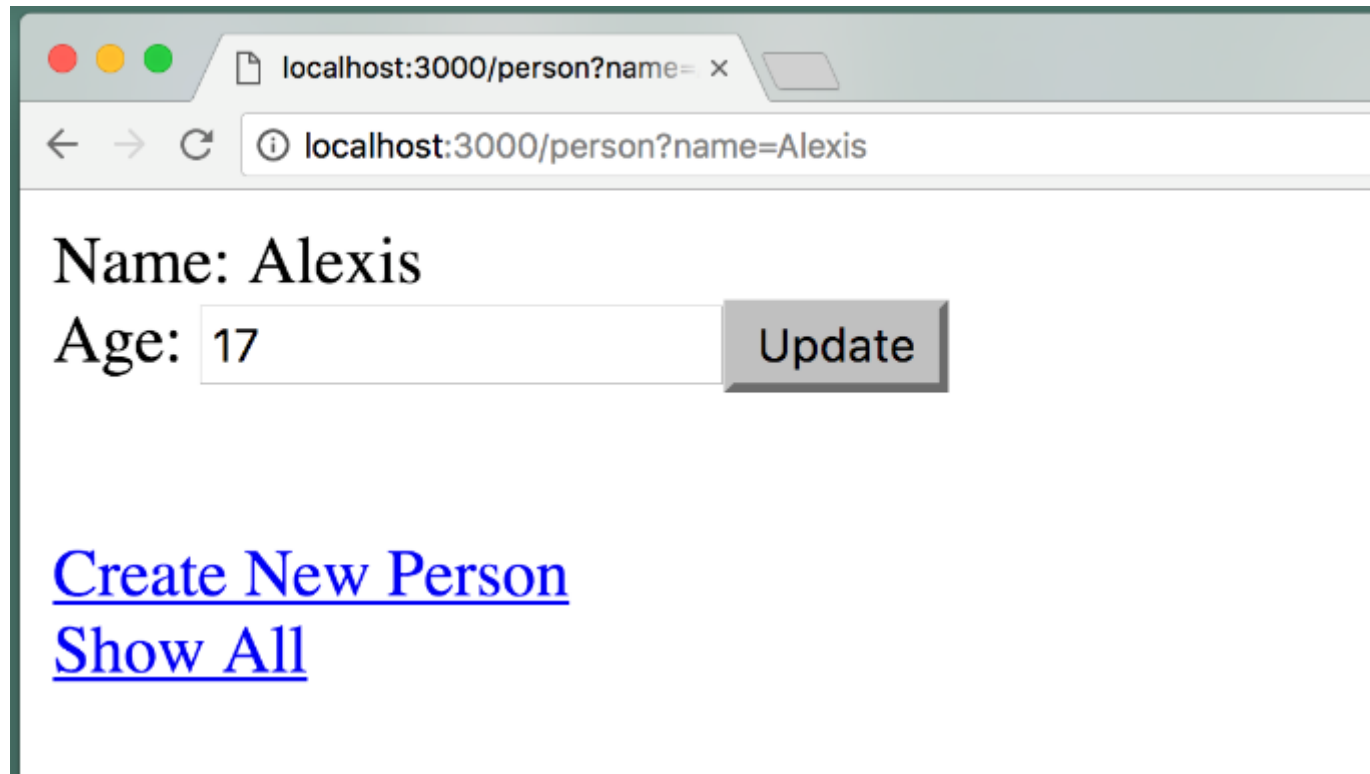
Name: Alexis

Age: 17

[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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' with an 'Update' button. Below the form are two blue underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

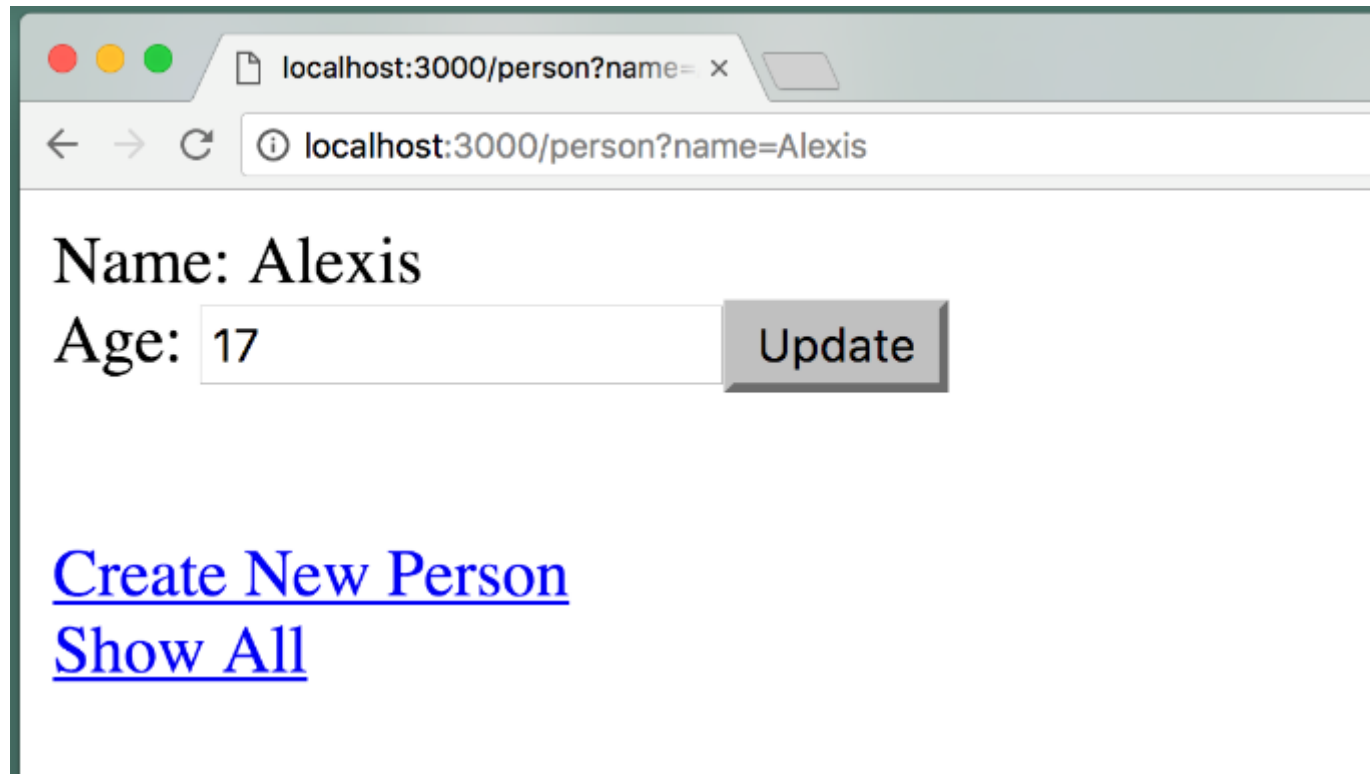
Name: Alexis

Age: 17

[Create New Person](#)

[Show All](#)

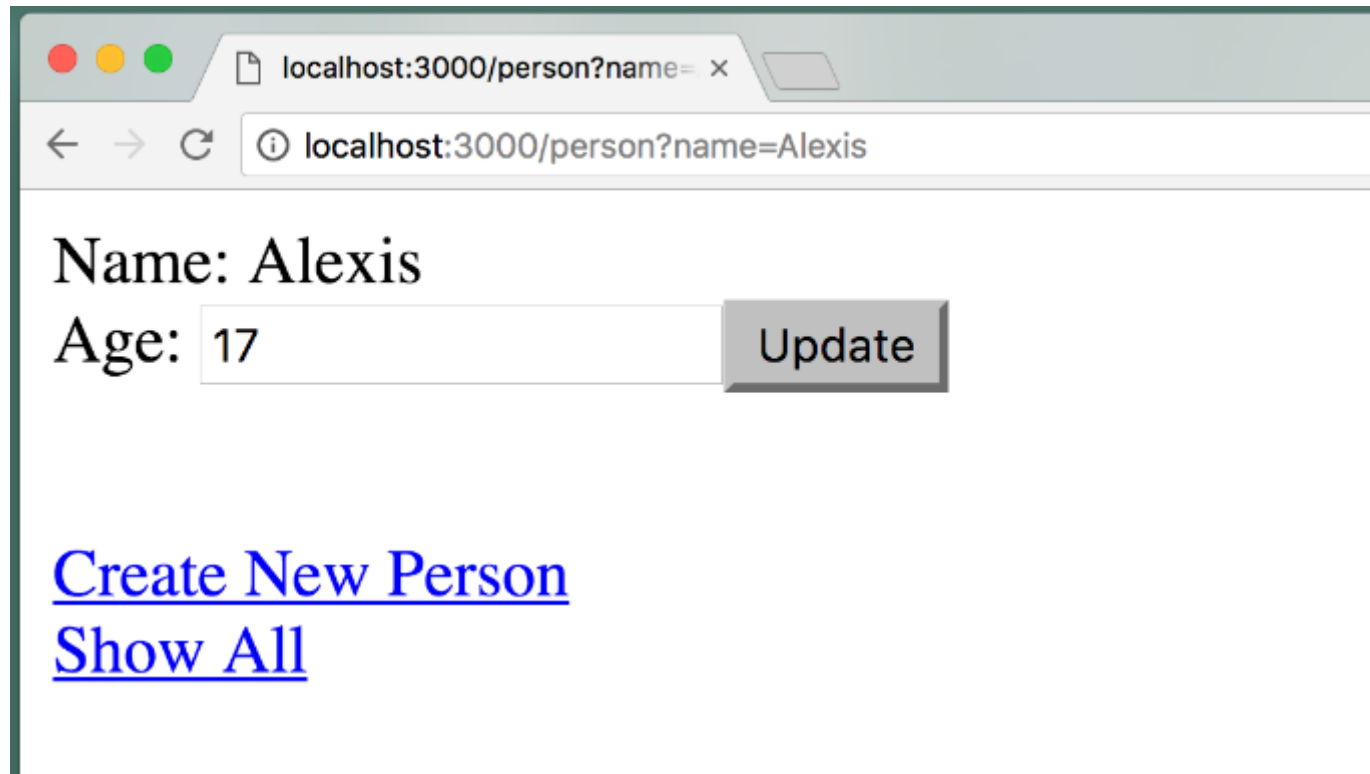
```
<!-- This is views/personInfo.ejs -->
<form action='/update' method='post'>
Name: <%= person.name %><br>
<input name='username' value='<%= person.name %>' hidden>
Age: <input name='age' value='<%= person.age %>'>
<input type='submit' value='Update'>
</form>
<br><a href='/public/personform.html'>Create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



A screenshot of a web browser window. The address bar shows the URL `localhost:3000/person?name= Alexis`. The page content displays a form with the following elements:

- The text "Name: Alexis" is shown.
- Below it, the text "Age:" is followed by a text input field containing the value "17".
- To the right of the age input field is a grey button labeled "Update".
- Below the form, there are two blue, underlined links: "Create New Person" and "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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' with an 'Update' button. Below this are two blue underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

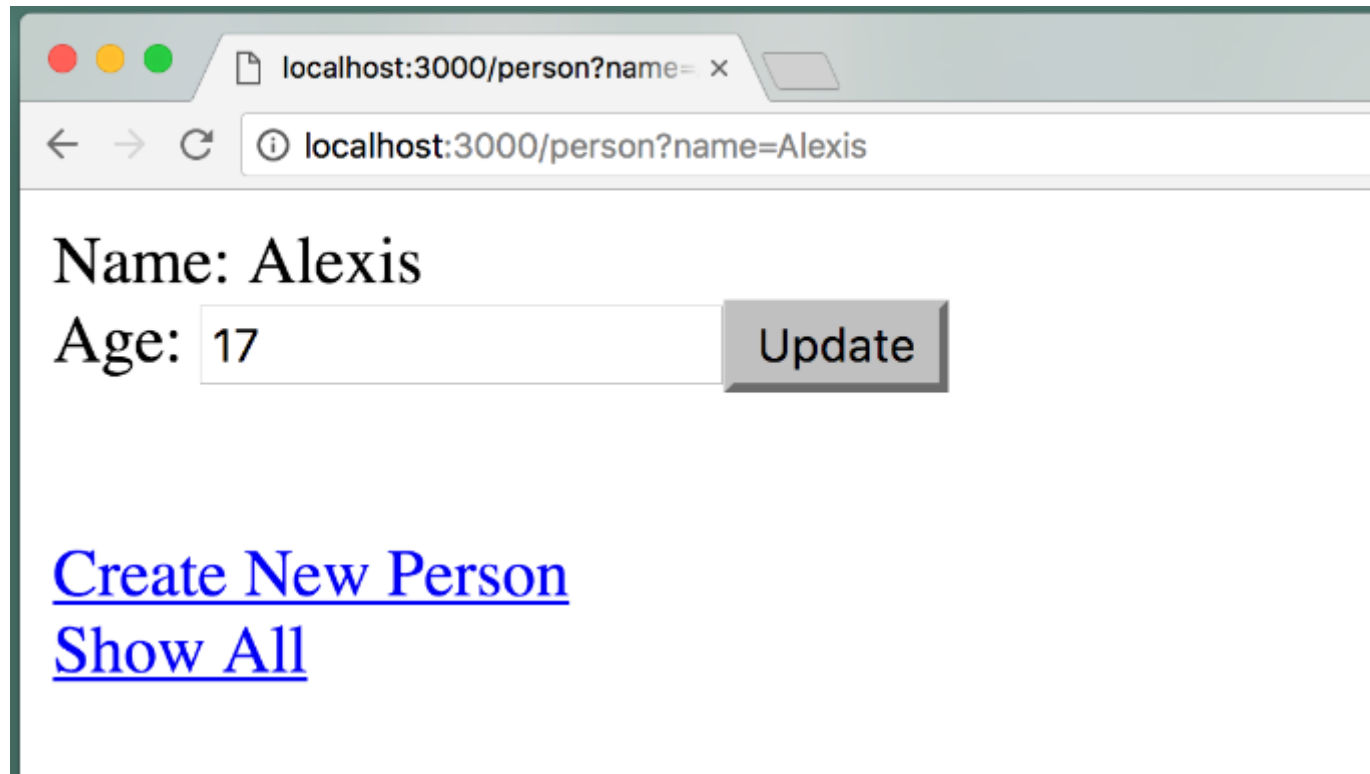
Name: Alexis

Age: 17

[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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' with an 'Update' button. Below the form are two blue underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

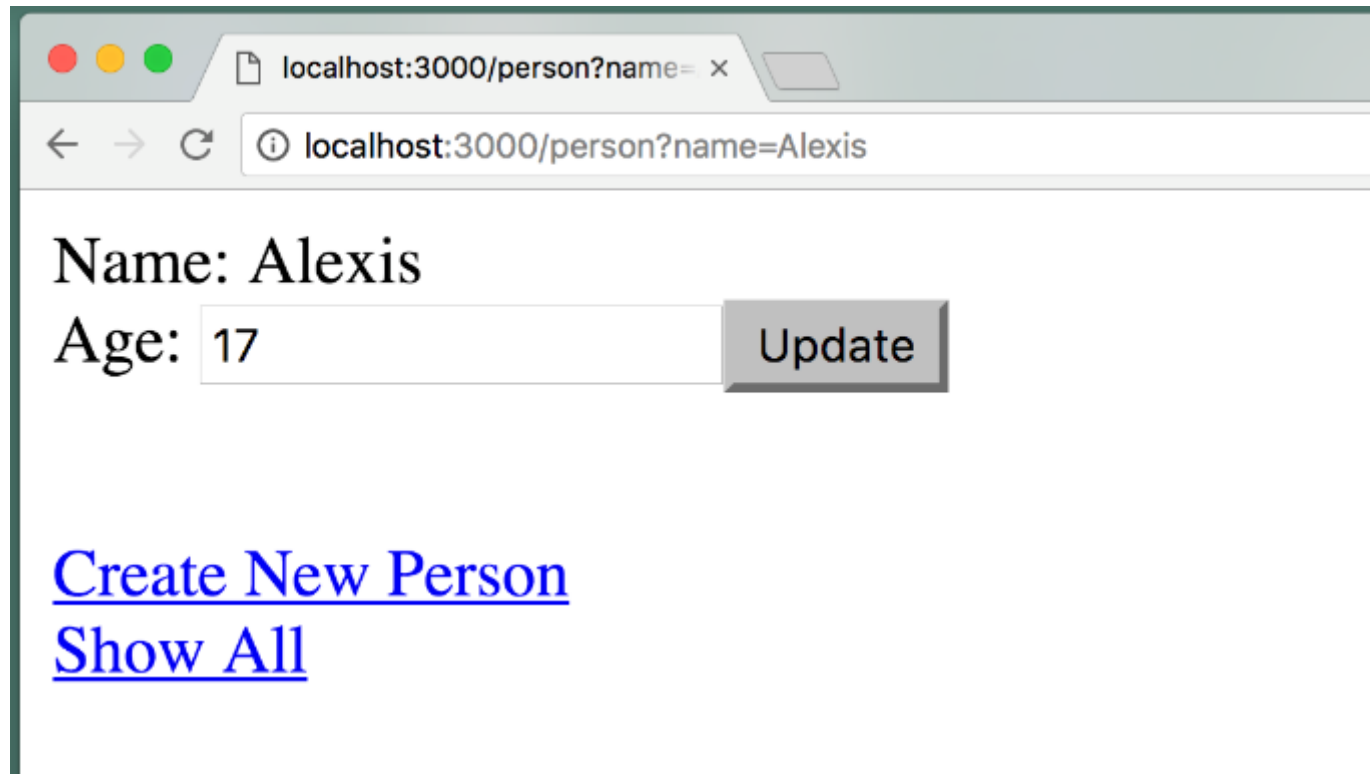
Name: Alexis

Age: 17

[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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' in a text input field. To the right of the input field is a grey 'Update' button. Below the form, there are two blue, underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

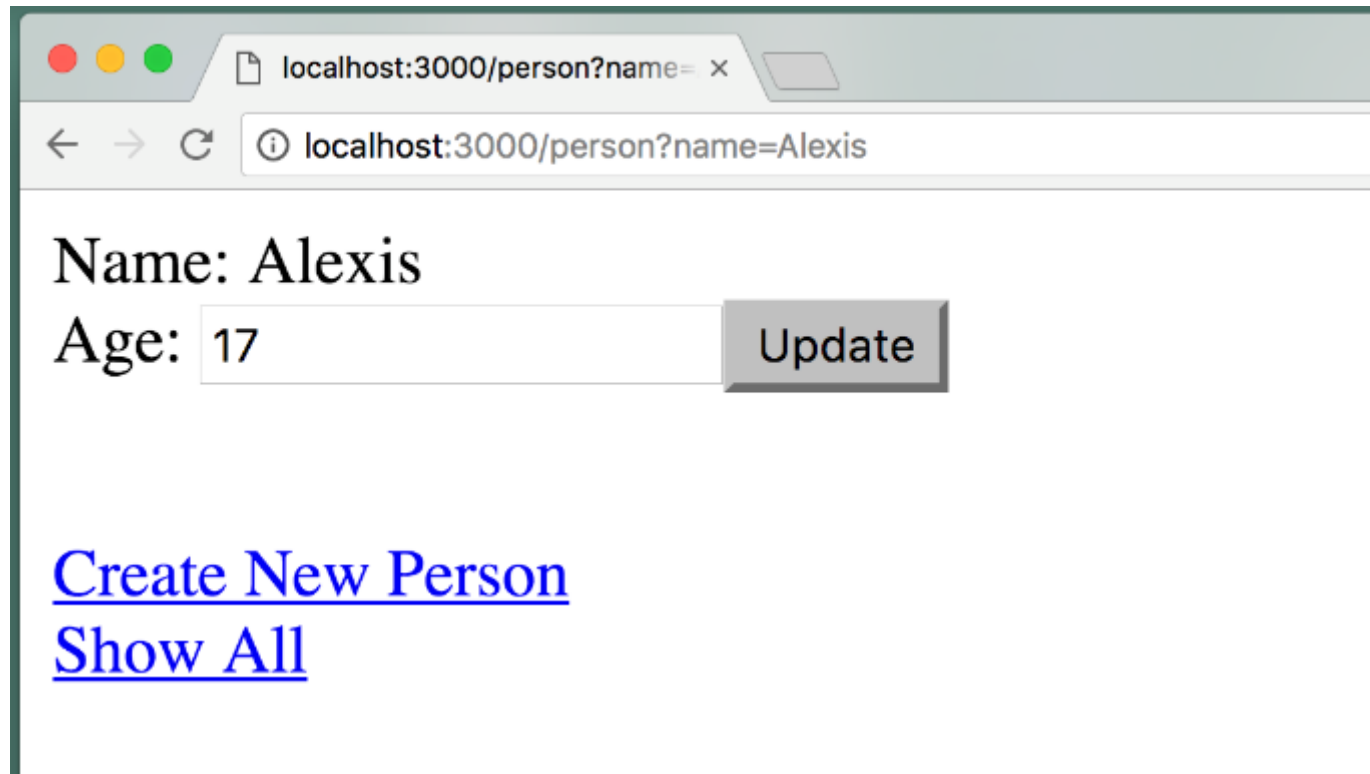
Name: Alexis

Age: 17

[Create New Person](#)

[Show All](#)

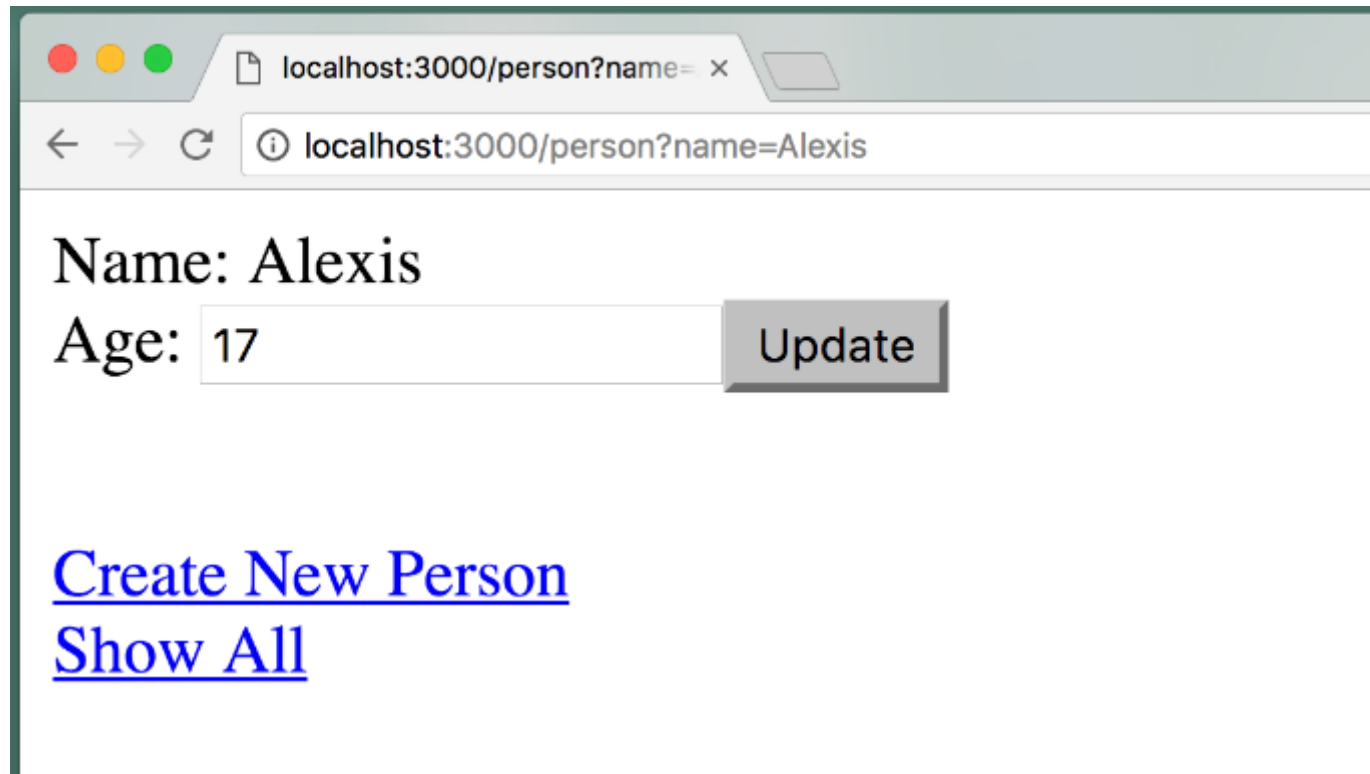
```
<!-- This is views/personInfo.ejs -->
<form action='/update' method='post'>
Name: <%= person.name %><br>
<input name='username' value='<%= person.name %>' hidden>
Age: <input name='age' value='<%= person.age %>'>
<input type='submit' value='Update'>
</form>
<br><a href='/public/personform.html'>Create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



A screenshot of a web browser window. The address bar shows the URL `localhost:3000/person?name=Alexis`. The page content displays a form with the following elements:

- The text "Name: Alexis" is shown.
- Below it, the text "Age:" is followed by a text input field containing the value "17".
- To the right of the age input field is a grey button labeled "Update".
- Below the form, there are two blue, underlined links: "Create New Person" and "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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' in a text input field. To the right of the input field is a grey 'Update' button. Below the form, there are two blue, underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

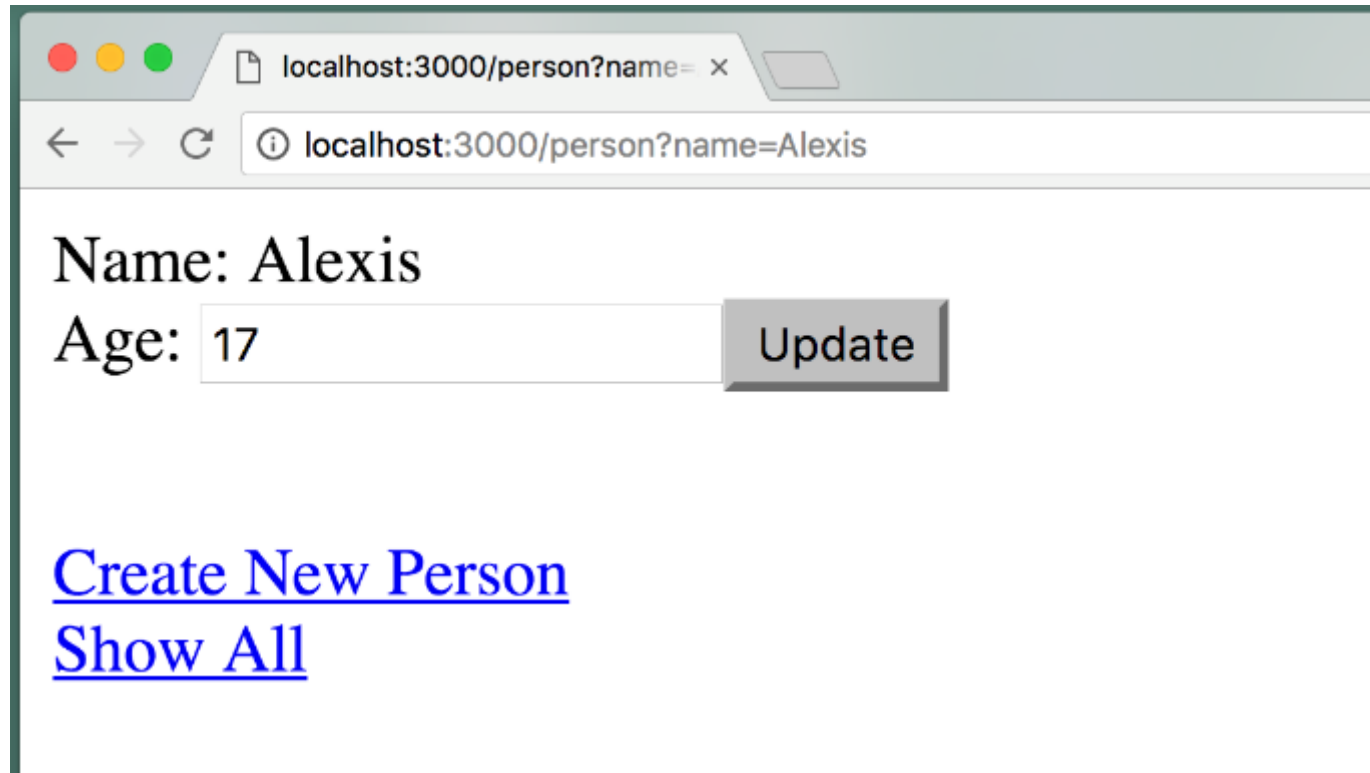
Name: Alexis

Age: 17

[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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' with an 'Update' button. Below the form are two blue underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

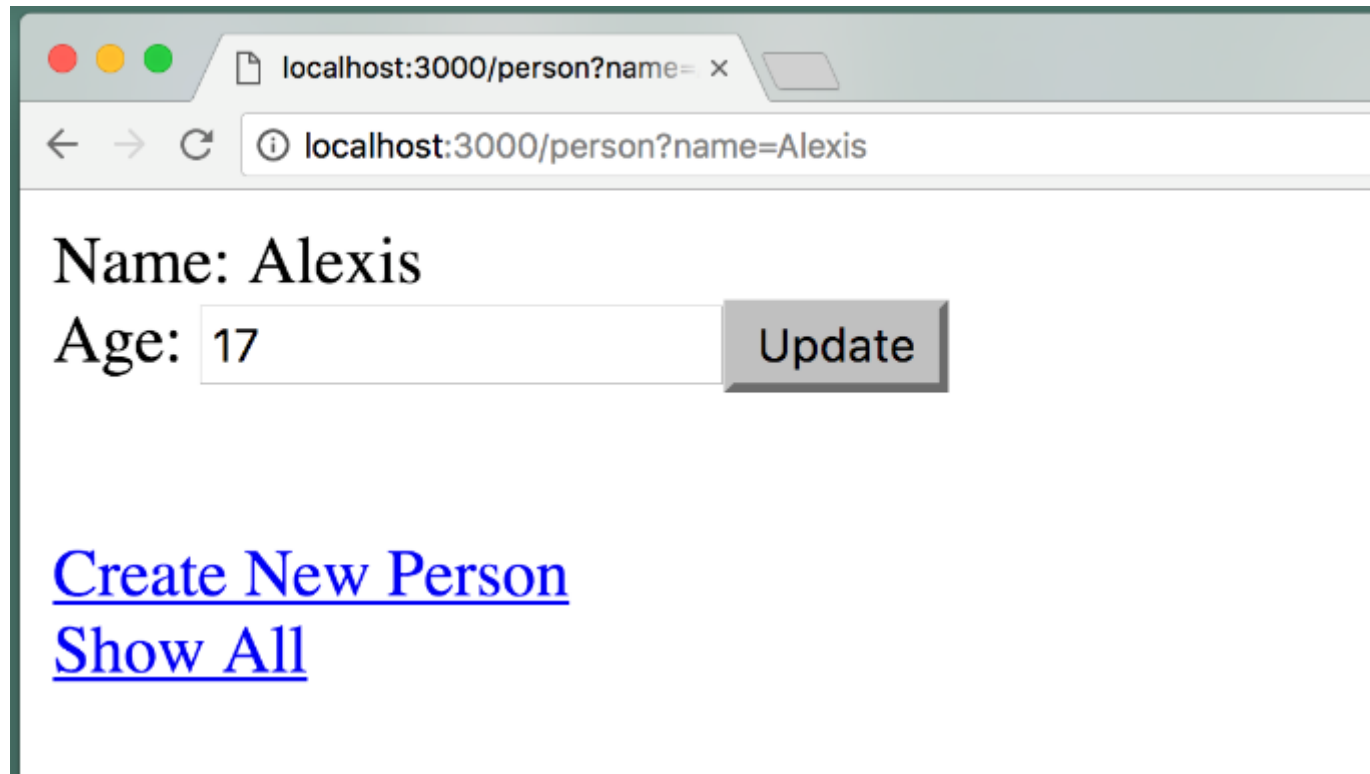
Name: Alexis

Age: 17

[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>
```



A screenshot of a web browser window. The address bar shows 'localhost:3000/person?name= Alexis'. The page content displays 'Name: Alexis' and 'Age: 17' in a text input field. To the right of the input field is a grey 'Update' button. Below the form, there are two blue, underlined links: 'Create New Person' and 'Show All'.

localhost:3000/person?name= Alexis

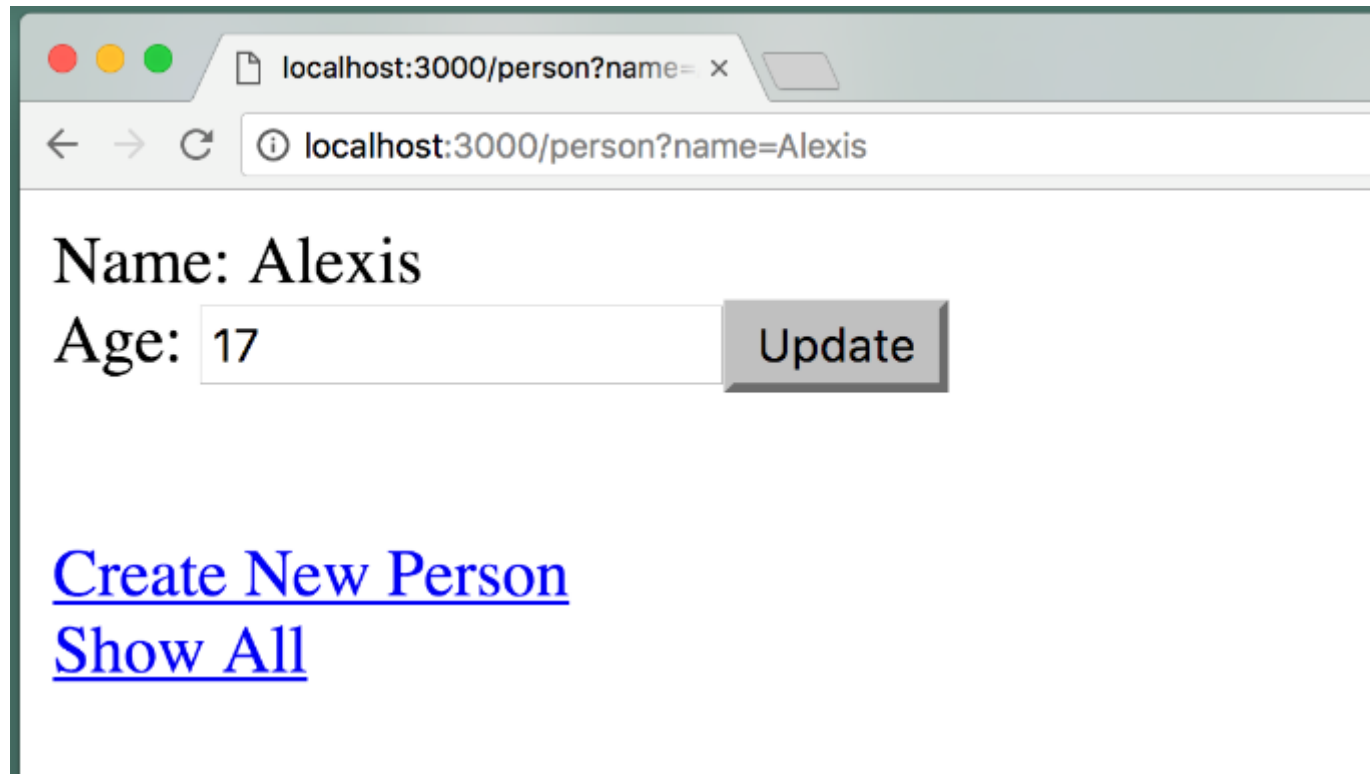
Name: Alexis

Age: 17

[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>
```



A screenshot of a web browser window. The address bar shows the URL `localhost:3000/person?name= Alexis`. The page content displays a form with the following elements:

- The text "Name: Alexis" is shown.
- Below it, the text "Age:" is followed by a text input field containing the value "17".
- To the right of the age input field is a grey button labeled "Update".
- Below the form, there are two blue, underlined links: "Create New Person" and "Show All".



localhost:3000/person?name= Alexis

Guest



localhost:3000/person?name=Alexis



Name: Alexis

Age: 17

Update

[Create New Person](#)

[Show All](#)

[←](#) [→](#) [↻](#) [localhost:3000/person?name=Alexis](#) [+](#) [⋮](#)

Name: Alexis

Age: [Update](#)

[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](#)

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```



```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```



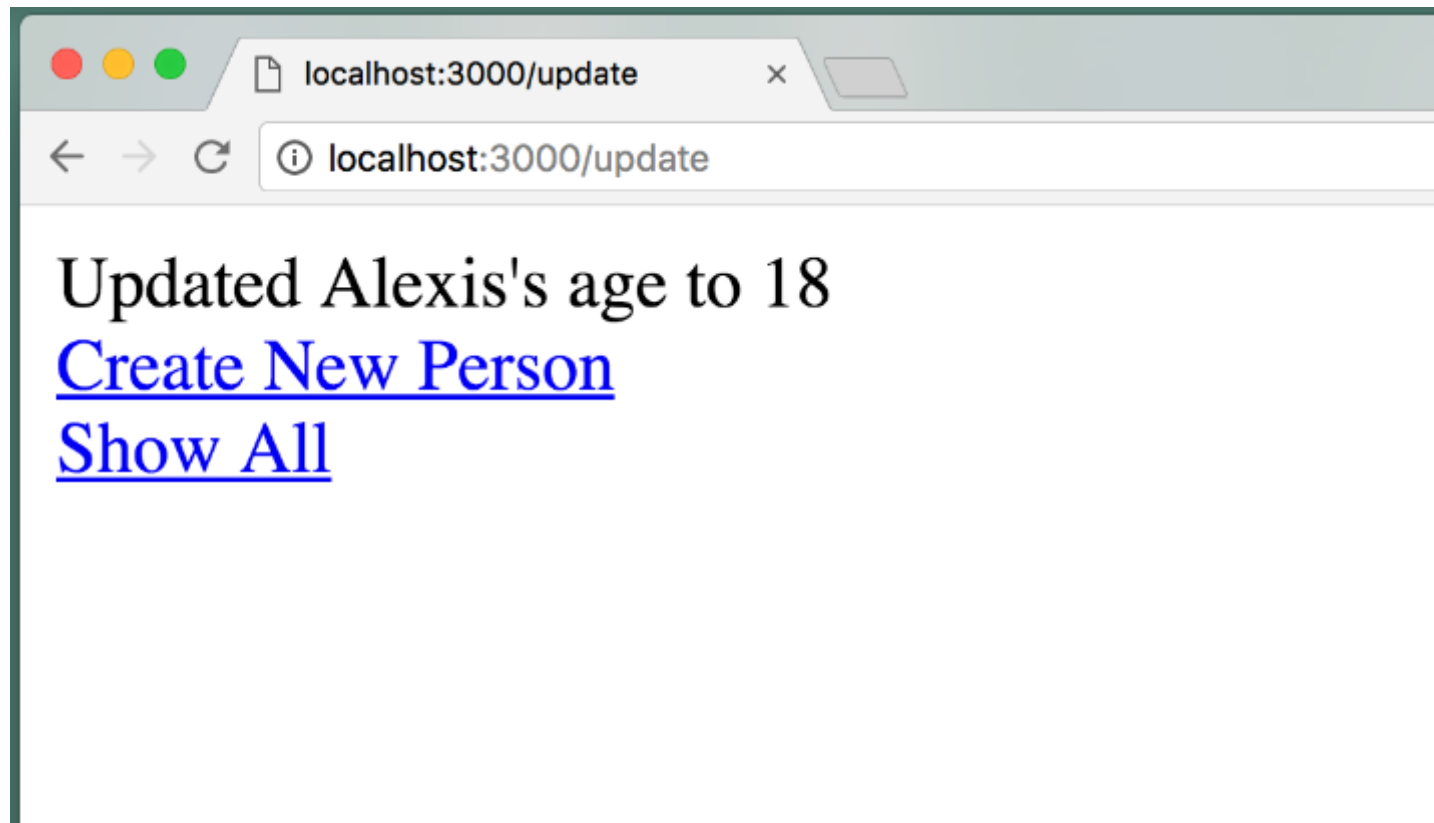
```
app.use('/update', (req, res) => {
  var updateName = req.body.username;
  Person.findOne( { name: updateName }, (err, person) => {
    if (err) {
      res.type('html').status(500);
      res.send('Error: ' + err);
    }
    else if (!person) {
      res.type('html').status(200);
      res.send('No person named ' + updateName);
    }
    else {
      person.age = req.body.age;
      person.save( (err) => {
        if (err) {
          res.type('html').status(500);
          res.send('Error: ' + err);
        }
        else {
          res.render('updated', { person: person });
        }
      });
    }
  });
});
});
```

```
<!-- 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>
```

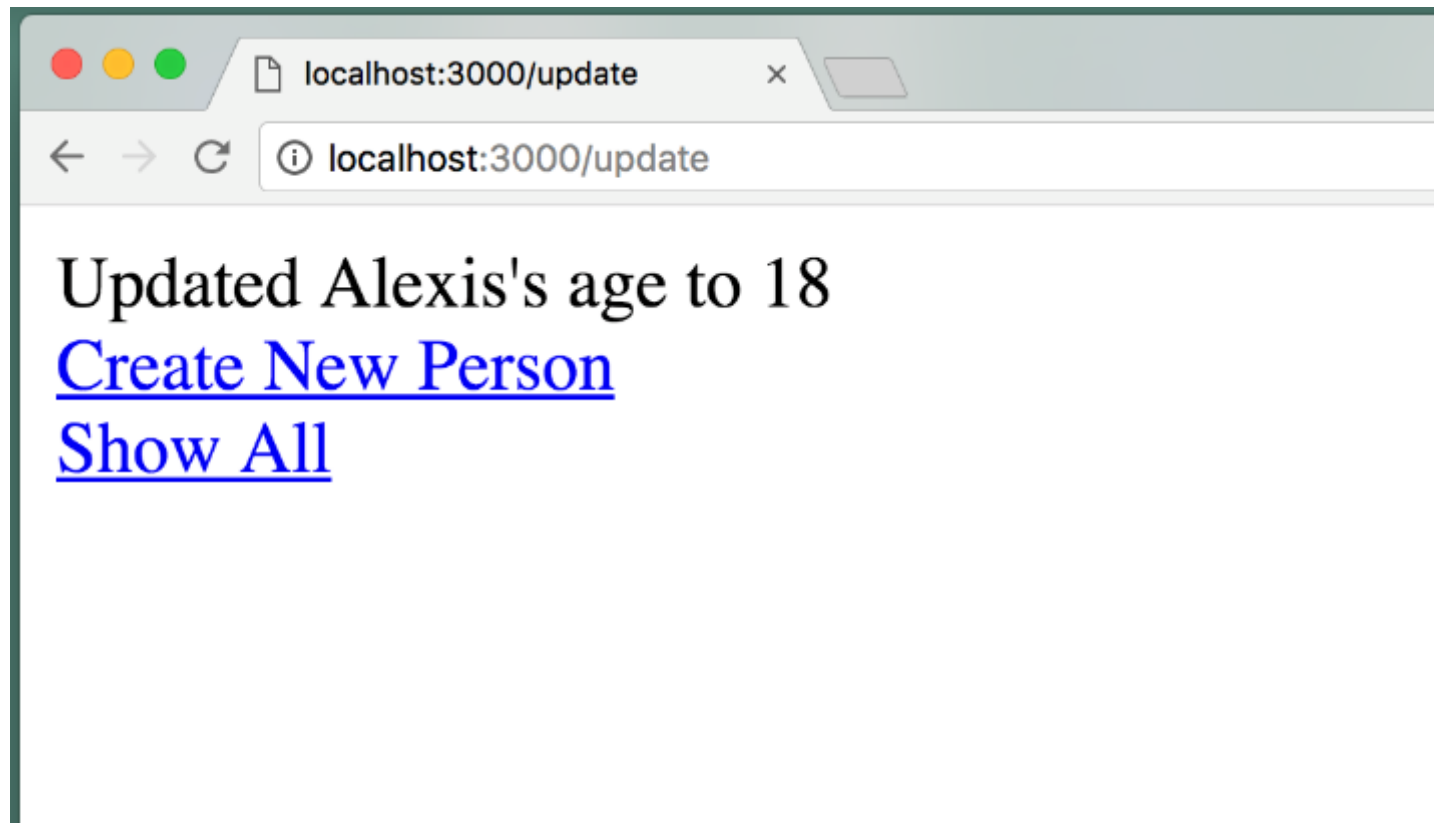


```
<!-- This is views/updated.ejs -->
```

```
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>
```

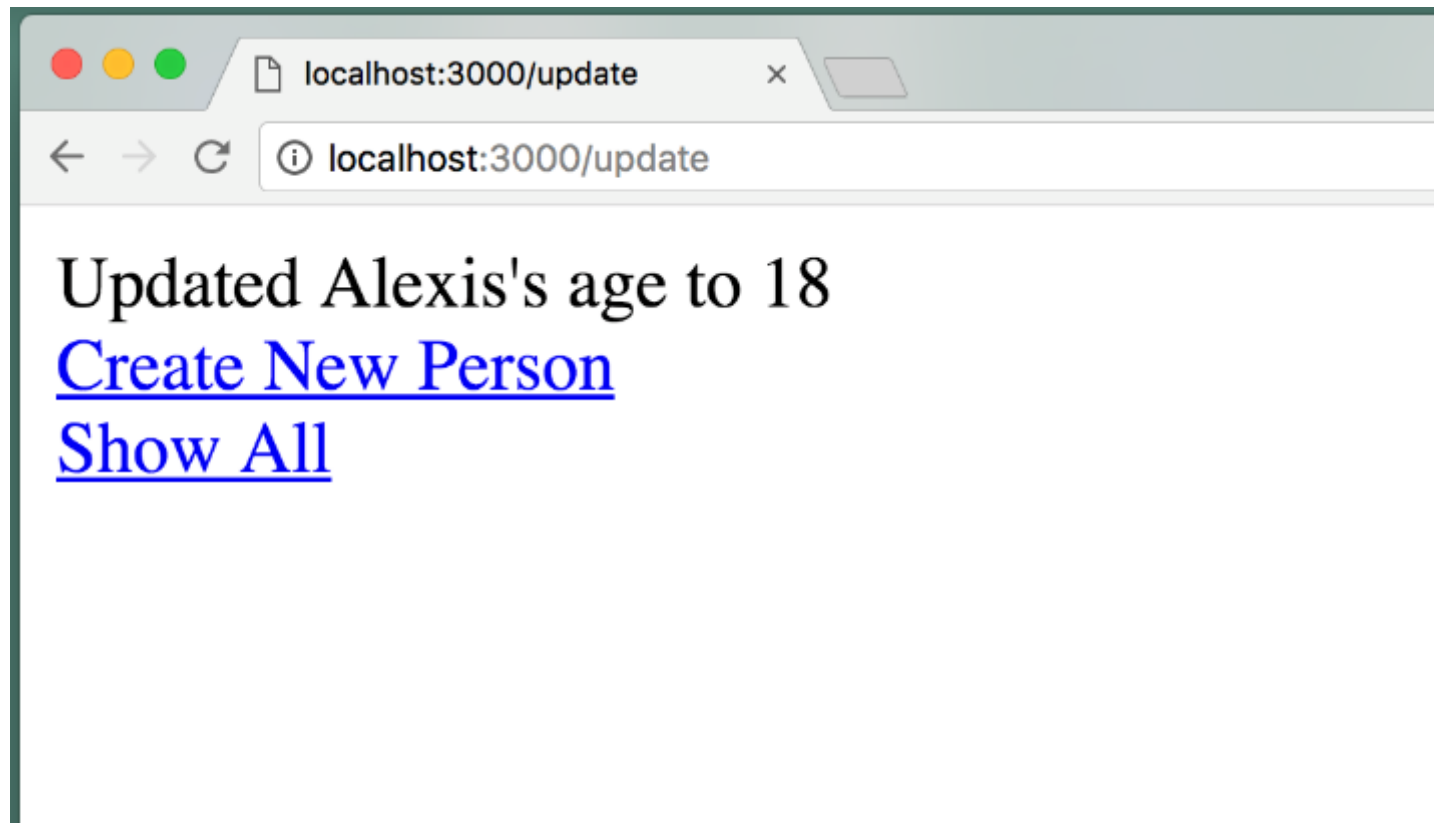


```
<!-- This is views/updated.ejs -->
```

```
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>
```

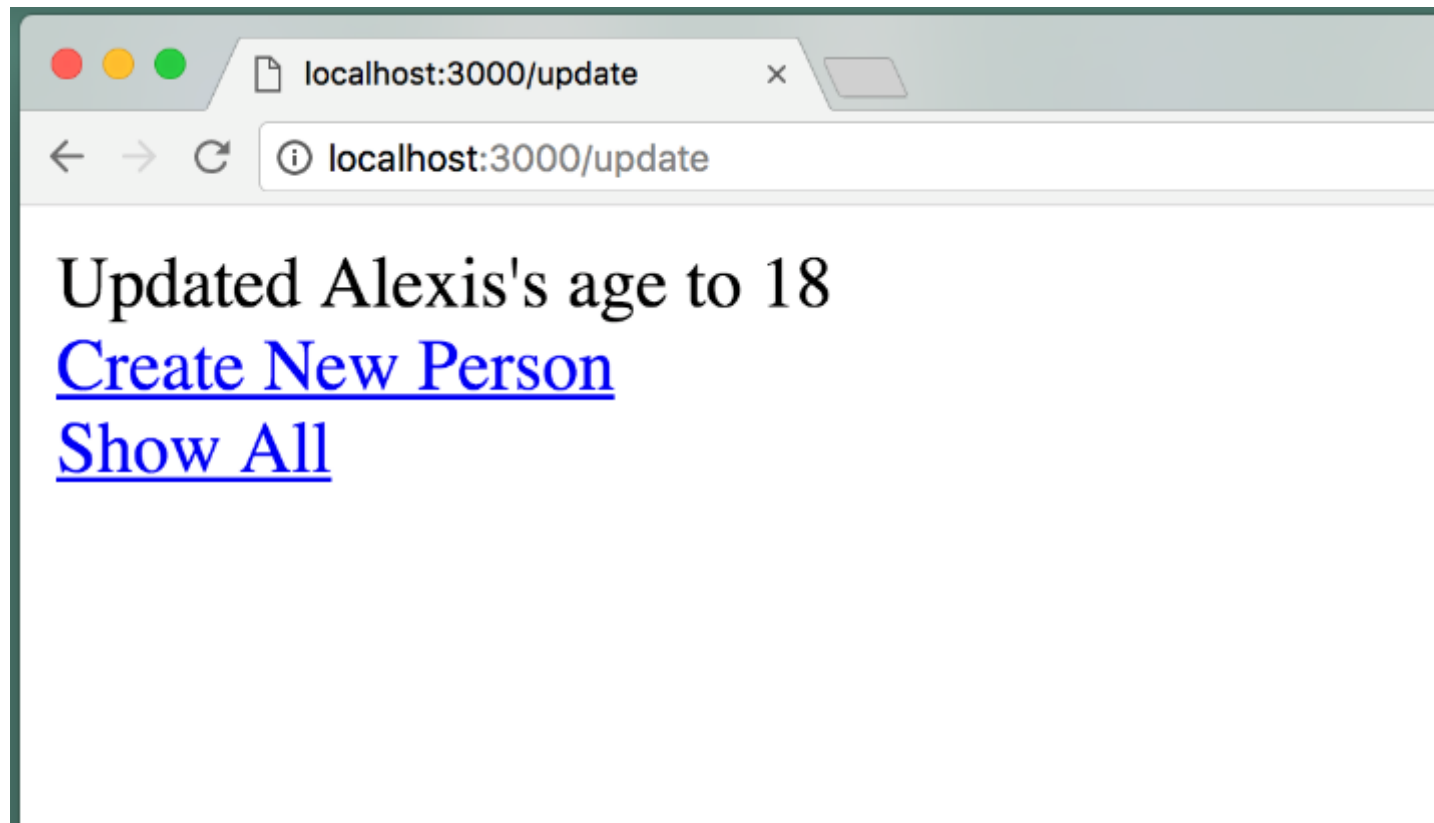


```
<!-- This is views/updated.ejs -->
```

```
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