



Video 4.3

Express Routing

Chris Murphy

# Review

---

- Node.js and Express represent HTTP requests and responses using JavaScript objects
- We can use these objects' properties and functions to dynamically generate the content that is sent in response to a request

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

```
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```

# Express Routing

---

```
var express = require('express');  
var app = express();  
  
app.use('/about', (req, res) => {  
  res.send('This is the about page.');  
});  
  
app.use('/login', (req, res) => {  
  res.send('This is the login page.');  
});  
  
app.use( /*default*/ (req, res) => {  
  res.status(404).send('Not found!');  
});  
  
app.listen(3000, () => {  
  console.log('Listening on port 3000');  
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

```
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```



# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

**app.use('/login', (req, res) => {**  
 res.send('This is the login page.');

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```

# Express Routing

---

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

app.use('/about', (req, res) => {
  res.send('This is the about page.');
```

  

```
});

app.use('/login', (req, res) => {
  res.send('This is the login page.');
```

  

```
});

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
```

  

```
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
```

  

```
});
```

# Express Middleware

---

- A **middleware** is a function that is invoked in the handling of an HTTP request
- It is used in the “middle” between receiving a request and sending a response
- Multiple middlewares can be chained together on the same request



# Middleware: Serving Static Files

- The simplest middleware is **express.static**, which serves static files that are locally stored

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- The simplest middleware is **express.static**, which serves static files that are locally stored

```
var express = require('express');  
var app = express();  
  
app.use('/public', express.static('files'));  
  
app.use( /*default*/ (req, res) => {  
  res.status(404).send('Not found!');  
});  
  
app.listen(3000, () => {  
  console.log('Listening on port 3000');  
});
```

# Middleware: Serving Static Files

- The simplest middleware is **express.static**, which serves static files that are locally stored

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- The simplest middleware is **express.static**, which serves static files that are locally stored

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- The simplest middleware is **express.static**, which serves static files that are locally stored

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- The simplest middleware is **express.static**, which serves static files that are locally stored

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

```
<!-- This is files/index.html -->

<html>
<body>
<h1>Hello!</h1>

<!-- File is files/images/kitty.jpg -->
</body>
</html>
```

```
<!-- This is files/index.html -->

<html>
<body>
<h1>Hello!</h1>

<!-- File is files/images/kitty.jpg -->
</body>
</html>
```



```
<!-- This is files/index.html -->

<html>
<body>
<h1>Hello!</h1>

<!-- File is files/images/kitty.jpg -->
</body>
</html>
```

```
<!-- This is files/index.html -->

<html>
<body>
<h1>Hello!</h1>

<!-- File is files/images/kitty.jpg -->
</body>
</html>
```

```
<!-- This is files/index.html -->

<html>
<body>
<h1>Hello!</h1>

<!-- File is files/images/kitty.jpg -->
</body>
</html>
```

```
<!-- This is files/index.html -->

<html>
<body>
<h1>Hello!</h1>

<!-- File is files/images/kitty.jpg -->
</body>
</html>
```



```
<!-- This is files/index.html -->

<html>
<body>
<h1>Hello!</h1>

<!-- File is files/images/kitty.jpg -->
</body>
</html>
```



# Middleware: Serving Static Files

- We can use the response object to send back specific HTML files as needed

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- We can use the response object to send back specific HTML files as needed

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- We can use the response object to send back specific HTML files as needed

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).send('Not found!');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```



# Middleware: Serving Static Files

- We can use the response object to send back specific HTML files as needed

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- We can use the response object to send back specific HTML files as needed

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- We can use the response object to send back specific HTML files as needed

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware: Serving Static Files

- We can use the response object to send back specific HTML files as needed

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

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Defining and Using Middleware

---

- Middleware functions can contain any amount of JavaScript code with any functionality
- They take three parameters: **req**, **res**, and **next**
- **next()** must be called at the end of the function to invoke the next middleware or the final response

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```



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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);
  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

app.use(logger);

app.use('/public', express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

// app.use(logger);

app.use('/public', logger, express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

// app.use(logger);

app.use('/public', logger, express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```



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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

// app.use(logger);

app.use('/public', logger, express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

// app.use(logger);

app.use('/public', logger, express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

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

var logger = (req, res, next) => {
  var url = req.url;
  var time = new Date();
  console.log('Received request for ' + url +
              ' at ' + time);

  next();
};

// app.use(logger);

app.use('/public', logger, express.static('files'));

app.use( /*default*/ (req, res) => {
  res.status(404).sendFile(__dirname + '/404.html');
});

app.listen(3000, () => {
  console.log('Listening on port 3000');
});
```

# Middleware Chaining

---

- Middleware functions are called in the order in which they are specified
- Each uses the same Request and Response objects
- A middleware function can modify the Request so that it can then be used by subsequent middleware functions “downstream” in the route

```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```





```
var nameFinder = (req, res, next) => {
```

```
  var name = req.query.name;
```

```
  if (name) req.username = name.toUpperCase();
```

```
  else req.username = 'Guest';
```

```
  next();
```

```
}
```

```
var greeter = (req, res, next) => {
```

```
  res.status(200).type('html');
```

```
  res.write('Hello, ' + req.username);
```

```
  next();
```

```
}
```

```
var adminName = (req, res, next) => {
```

```
  req.username = 'Admin';
```

```
  next();
```

```
}
```

```
app.use('/welcome', nameFinder, greeter,
```

```
      (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,
```

```
      (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}  
  
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}  
  
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}  
  
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );  
  
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```





```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```





```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => {  
  var name = req.query.name;  
  if (name) req.username = name.toUpperCase();  
  else req.username = 'Guest';  
  next();  
}
```

```
var greeter = (req, res, next) => {  
  res.status(200).type('html');  
  res.write('Hello, ' + req.username);  
  next();  
}
```

```
var adminName = (req, res, next) => {  
  req.username = 'Admin';  
  next();  
}
```

```
app.use('/welcome', nameFinder, greeter,  
        (req, res) => { res.end(); } );
```

```
app.use('/admin', adminName, greeter,  
        (req, res) => { res.end(); } );
```



# Middleware, Routes, and Routers

- We may find that the same combinations of middleware functions are being used in multiple routes
- We can combine middleware functions into “sub-routes” using **Routers** and then use those in our routes



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } ));
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } ));
```

```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
    greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
    greeter, footer, (req, res) => { res.end(); } );
```

```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
  greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
  greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
  greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
  greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
    greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
    greeter, footer, (req, res) => { res.end(); } );
```





```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
  greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
  greeter, footer, (req, res) => { res.end(); } );
```

```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
    greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
    greeter, footer, (req, res) => { res.end(); } );
```

```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```





```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
      greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
      greeter, footer, (req, res) => { res.end(); } );
```

```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
app.use('/welcome', logger, nameFinder, header,  
      greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
      greeter, footer, (req, res) => { res.end(); } );
```

```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
var commonRoute = express.Router();
```

```
commonRoute.use(header, greeter, footer);
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }
```

```
var greeter = (req, res, next) => { . . . }
```

```
var adminName = (req, res, next) => { . . . }
```

```
var logger = (req, res, next) => { . . . }
```

```
var header = (req, res, next) => { . . . }
```

```
var footer = (req, res, next) => { . . . }
```

```
var commonRoute = express.Router();
```

```
commonRoute.use(header, greeter, footer);
```

```
app.use('/welcome', logger, nameFinder, header,  
        greeter, footer, (req, res) => { res.end(); } );
```

```
app.use('/admin', logger, adminName, header,  
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, header,
        greeter, footer, (req, res) => { res.end(); } ));

app.use('/admin', logger, adminName, header,
        greeter, footer, (req, res) => { res.end(); } ));
```



```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, header,
        greeter, footer, (req, res) => { res.end(); } ));

app.use('/admin', logger, adminName, header,
        greeter, footer, (req, res) => { res.end(); } ));
```



```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, header,
        greeter, footer, (req, res) => { res.end(); } ));

app.use('/admin', logger, adminName, header,
        greeter, footer, (req, res) => { res.end(); } ));
```



```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, header,
      greeter, footer, (req, res) => { res.end(); } );

app.use('/admin', logger, adminName, header,
      greeter, footer, (req, res) => { res.end(); } );
```





```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, commonRoute,
        (req, res) => { res.end(); } );

app.use('/admin', logger, adminName, header,
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, commonRoute,
        (req, res) => { res.end(); } );

app.use('/admin', logger, adminName, header,
        greeter, footer, (req, res) => { res.end(); } );
```



```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, commonRoute,
        (req, res) => { res.end(); } ));

app.use('/admin', logger, adminName, header,
        greeter, footer, (req, res) => { res.end(); } ));
```



```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, commonRoute,
        (req, res) => { res.end(); } ));

app.use('/admin', logger, adminName, header,
        greeter, footer, (req, res) => { res.end(); } ));
```

```
var nameFinder = (req, res, next) => { . . . }

var greeter = (req, res, next) => { . . . }

var adminName = (req, res, next) => { . . . }

var logger = (req, res, next) => { . . . }

var header = (req, res, next) => { . . . }

var footer = (req, res, next) => { . . . }

var commonRoute = express.Router();
commonRoute.use(header, greeter, footer);

app.use('/welcome', logger, nameFinder, commonRoute,
        (req, res) => { res.end(); } );

app.use('/admin', logger, adminName, commonRoute,
        (req, res) => { res.end(); } );
```



# Summary

---

- **Routing** allows us to specify different functionality for different HTTP requests
- Routing uses **middleware** functions, each of which handles a different part of the functionality
- Middleware functions can be chained together and can pass values to each other by modifying the Request object
- **Routers** allow us to combine middleware functions into common “sub-routes”