Middleware

Writing Our Own

Level 2- Part II

Writing the logger module

We assign our logger function to **module.exports** in order to export it as a Node module and make it accessible from other files

app.js
logger.js
public

logger.js

```
module.exports = function(request, response, next) {







}
```

*The Node module system follows the **CommonJS** specification*

Tracking the start time for the request

We use the **Date** object to track the start time.

app.js
logger.js
public

logger.js

```
module.exports = function(request, response, next) {
  var start = +new Date();



  next();
}
```

plus sign converts date Object to **milliseconds**

moves request to the **next** middleware in the stack

Assigning the readable stream

Standard out is a **writeable stream** which we will be writing the log to

app.js
logger.js
public

logger.js

```
module.exports = function(request, response, next) {
  var start = +new Date();
  var stream = process.stdout;




  next();
}
```

Reading the url and HTTP method

The **request** object gives us the **requested URL** and the HTTP **method** used

- app.js
- **logger.js**
- **public**

logger.js

```
module.exports = function(request, response, next) {
  var start = +new Date();
  var stream = process.stdout;
  var url = request.url;
  var method = request.method;


  next();
}
```

Listening for the finish event

The **response** object is an **EventEmitter** which we can use to **listen** to events

- app.js
- logger.js
- **public**

logger.js

```
module.exports = function(request, response, next) {
  var start = +new Date();
  var stream = process.stdout;
  var url = request.url;
  var method = request.method;

  response.on('finish', function() {
    ...
  });

  next();
}
```

event handler function runs **asynchronously**

the **finish** event is emitted when the response has been handed off to the OS

Calculating the request interval

- app.js
- **logger.js**
- **public**

logger.js

```
module.exports = function(request, response, next) {
  var start = +new Date();
  var stream = process.stdout;
  var url = request.url;
  var method = request.method;

  response.on('finish', function() {
    var duration = +new Date() - start;
    ...
  });
  next();
}
```

Calculate the duration of the request

Composing the log message

app.js
logger.js
public

logger.js

```
module.exports = function(request, response, next) {
  var start = +new Date();
  var stream = process.stdout;
  var url = request.url;
  var method = request.method;

  response.on('finish', function() {
    var duration = +new Date() - start;
    var message = method  + ' to ' + url +
      '\ntook ' + duration + ' ms \n\n';

    ...
  });
  next();
}
```

Printing and moving along

We call the **write** function on the writeable **stream** in order to print the log

app.js
logger.js
public

logger.js

```
module.exports = function(request, response, next) {
  var start = +new Date();
  var stream = process.stdout;
  var url = request.url;
  var method = request.method;

  response.on('finish', function() {
    var duration = +new Date() - start;
    var message = method  + ' to ' + url +
      '\ntook ' + duration + ' ms \n\n';
    stream.write(message);
  });

  next();
}
```

prints the log message

Using the logger module

We **require** and **use** our logger module in our application

app.js
logger.js
public

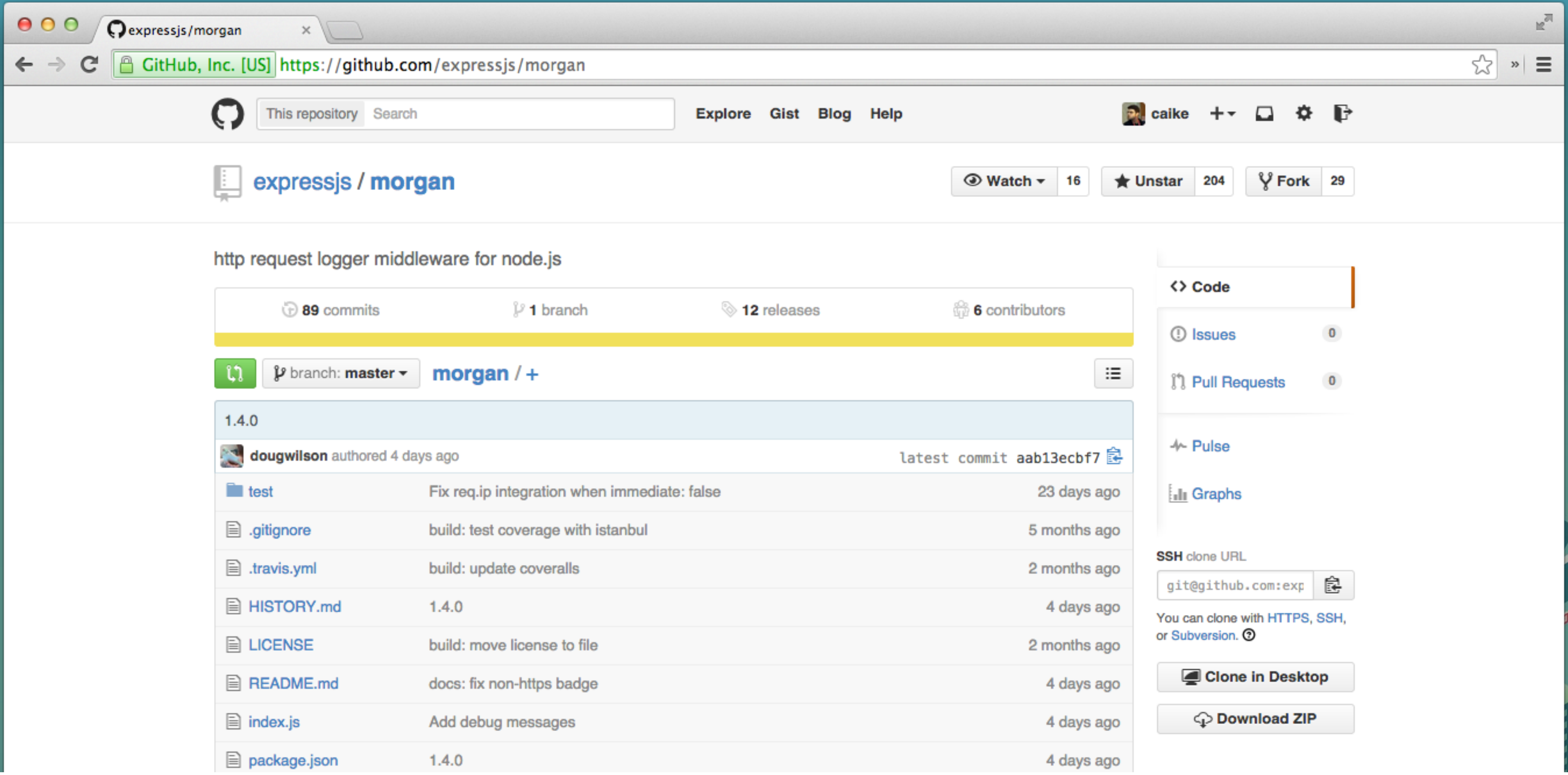
require and use our module

app.js

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

var logger = require('./logger');
app.use(logger);

app.use(express.static('public'));

app.get('/blocks', function(request, response) {
  var blocks = ['Fixed', 'Movable', 'Rotating'];
  response.json(blocks);
});

app.listen(3000, function () {
  console.log('Listening on 3000 \n');
});
```

Reading the source for Morgan

https://github.com/expressjs/morgan



http request logger middleware for node.js

89 commits · 1 branch · 12 releases · 6 contributors

branch: master · morgan / +

1.4.0

dougwilson authored 4 days ago · latest commit aab13ecbf7

test	Fix req.ip integration when immediate: false	23 days ago
.gitignore	build: test coverage with istanbul	5 months ago
.travis.yml	build: update coveralls	2 months ago
HISTORY.md	1.4.0	4 days ago
LICENSE	build: move license to file	2 months ago
README.md	docs: fix non-https badge	4 days ago
index.js	Add debug messages	4 days ago
package.json	1.4.0	4 days ago