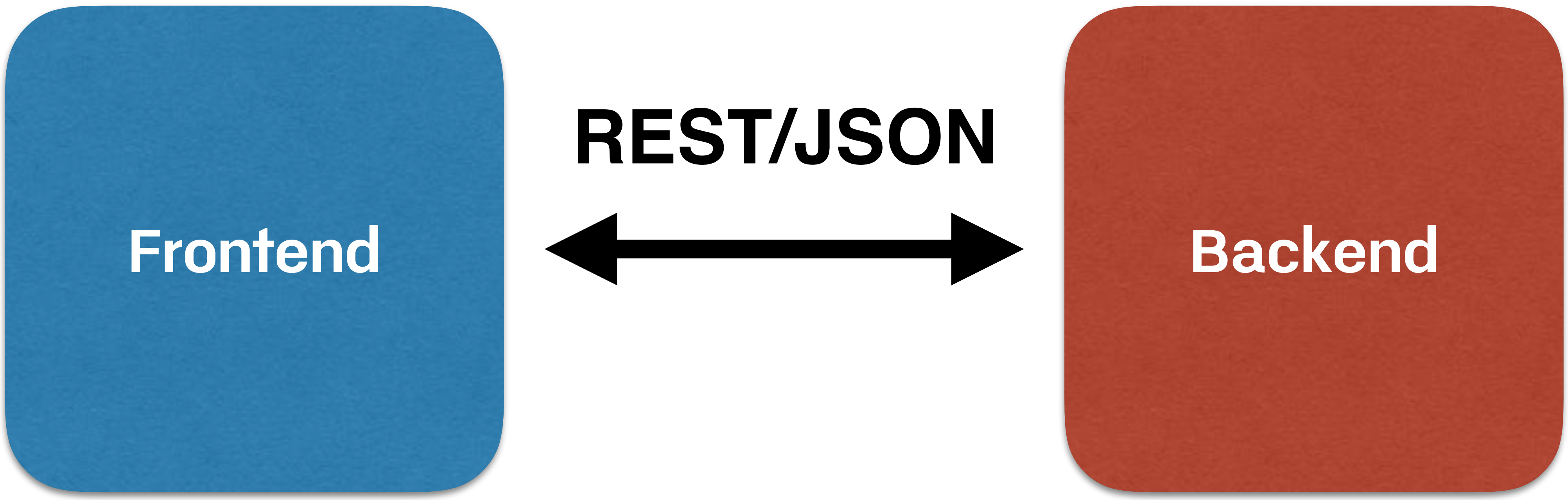
Working with Highchart

@somkiat.cc

Structure

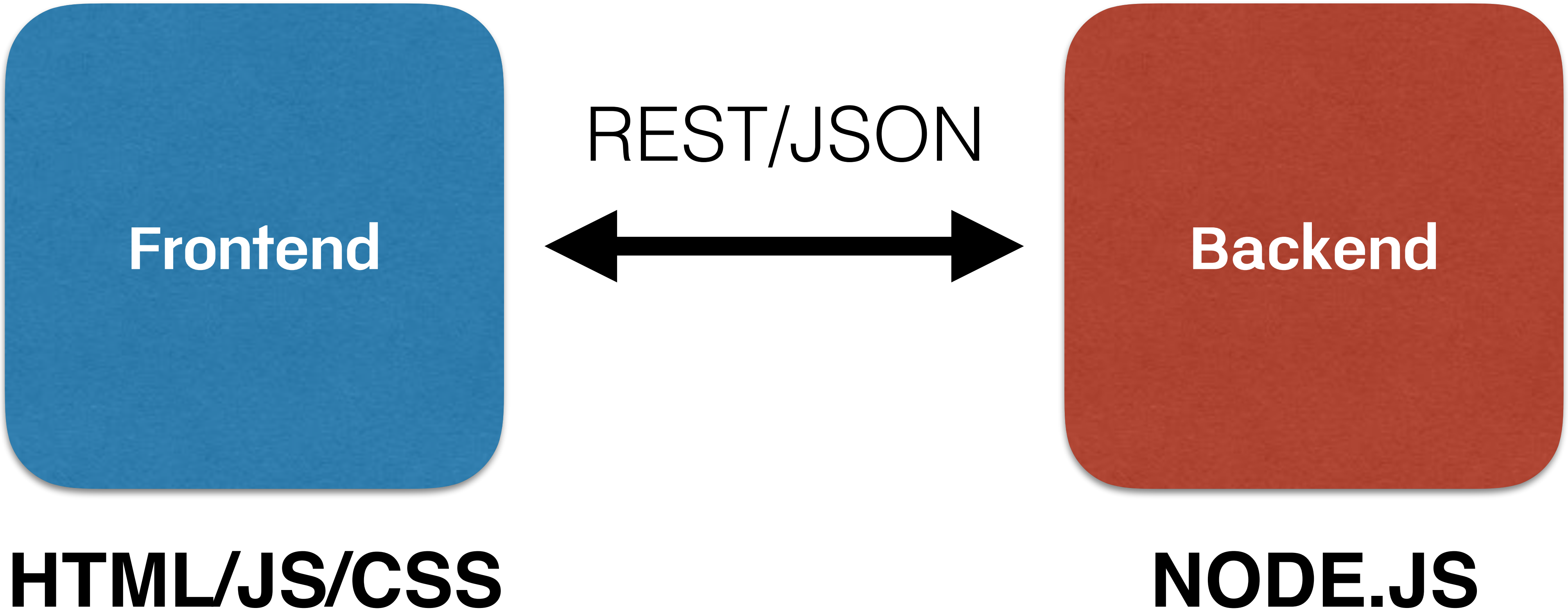


Frontend

REST/JSON

Backend

Technologies



Frontend

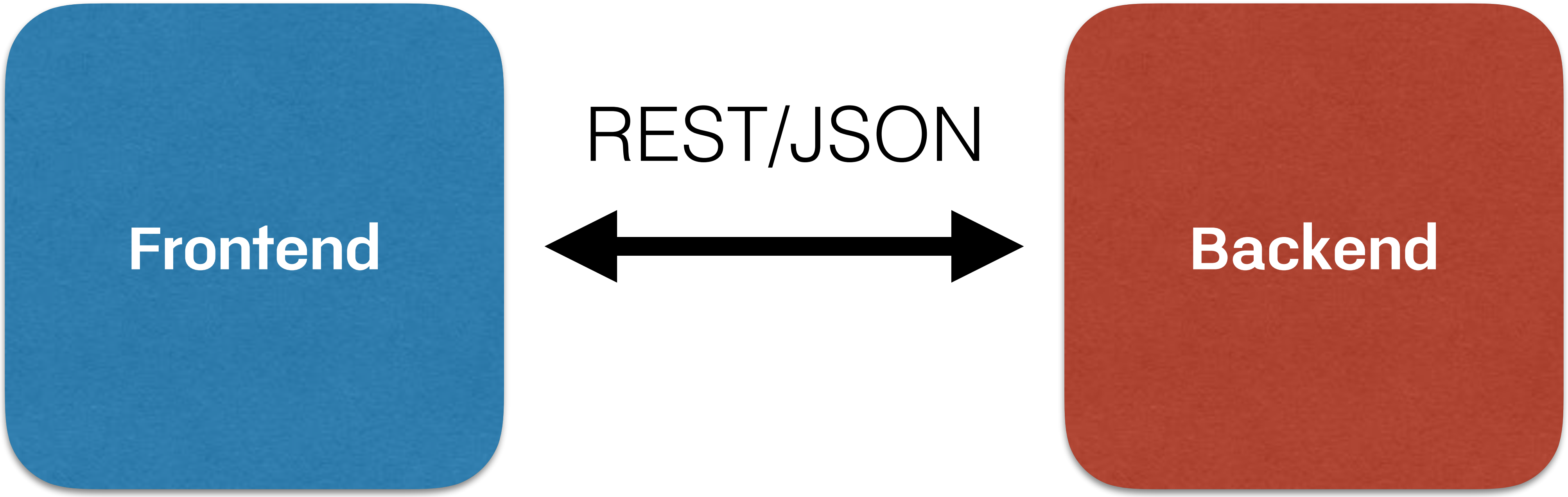
REST/JSON

Backend

HTML/JS/CSS

NODE.JS

Dependency manager



REST/JSON

Frontend

Backend

HTML/JS/CSS

Bower

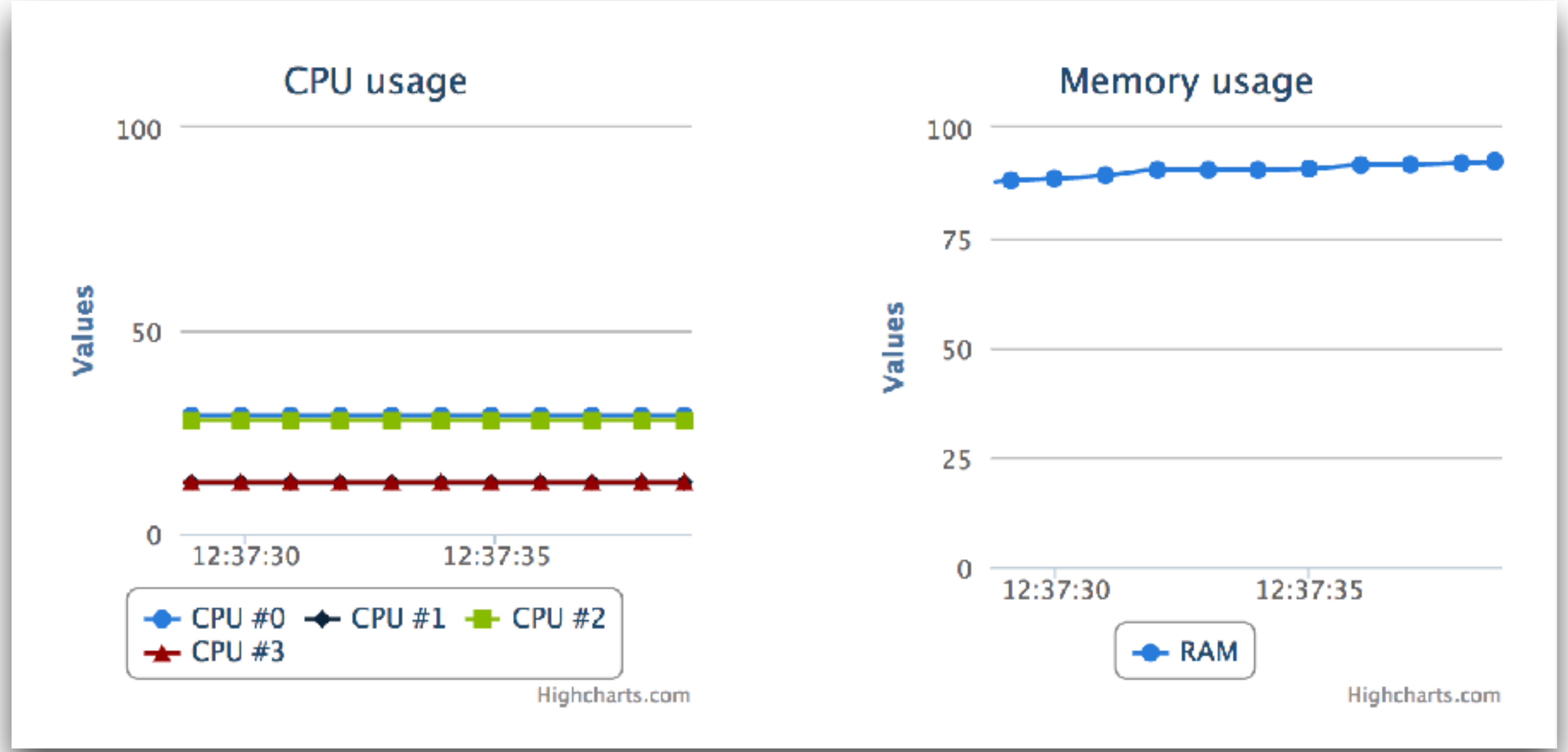
NODE.JS

NPM

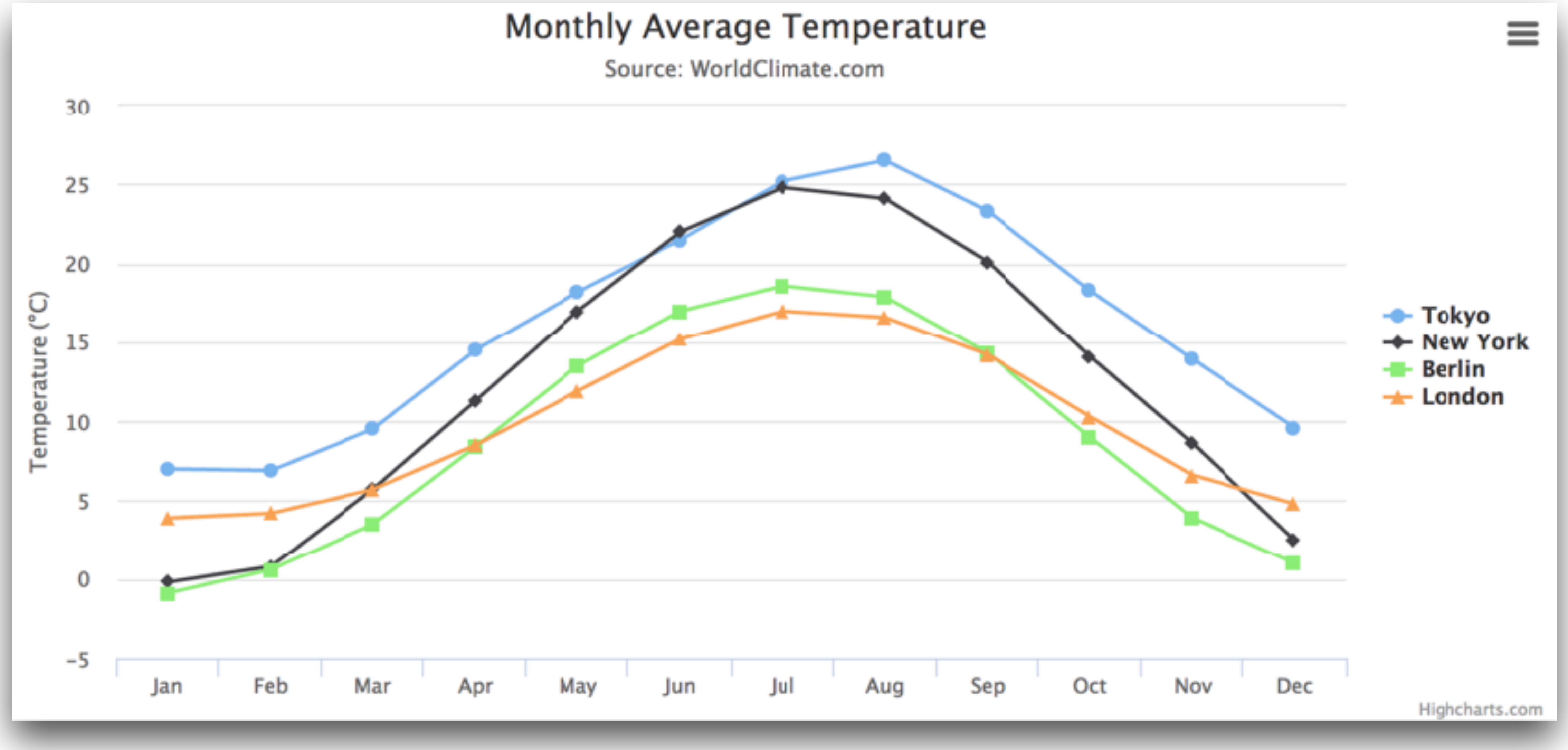
Monitoring CPU and Memory

Realtime ...

Workshop


CPU usage
100
50
0
Values
12:37:30
12:37:35
CPU #0
CPU #1
CPU #2
CPU #3
Highcharts.com
Memory usage
100
75
50
25
0
Values
12:37:30
12:37:35
RAM
Highcharts.com

Create graph with Highchart



http://www.highcharts.com/demo/line-basic

Line

Title and Subtitle

X-Axis

Y-Axis

Tooltip

Series of data

Manage library with bower

```
$npm install -g bower
```

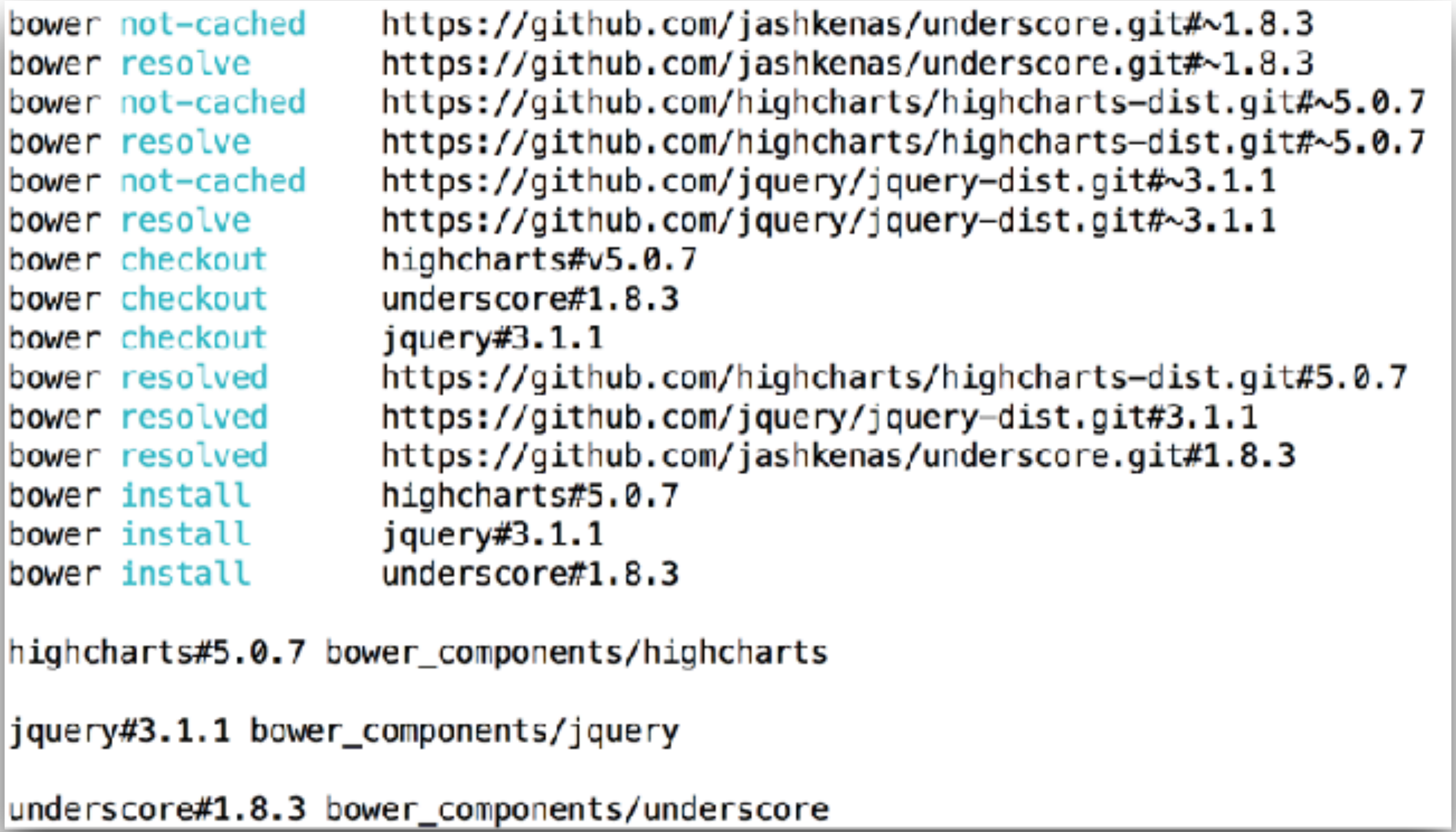
https://bower.io/

Create bower.json

```
{
  "name": "monitoring",
  "dependencies": {
    "jquery": "~3.1.1",
    "highcharts": "~5.0.7",
    "underscore": "~1.8.3"
  }
}
```

Install library

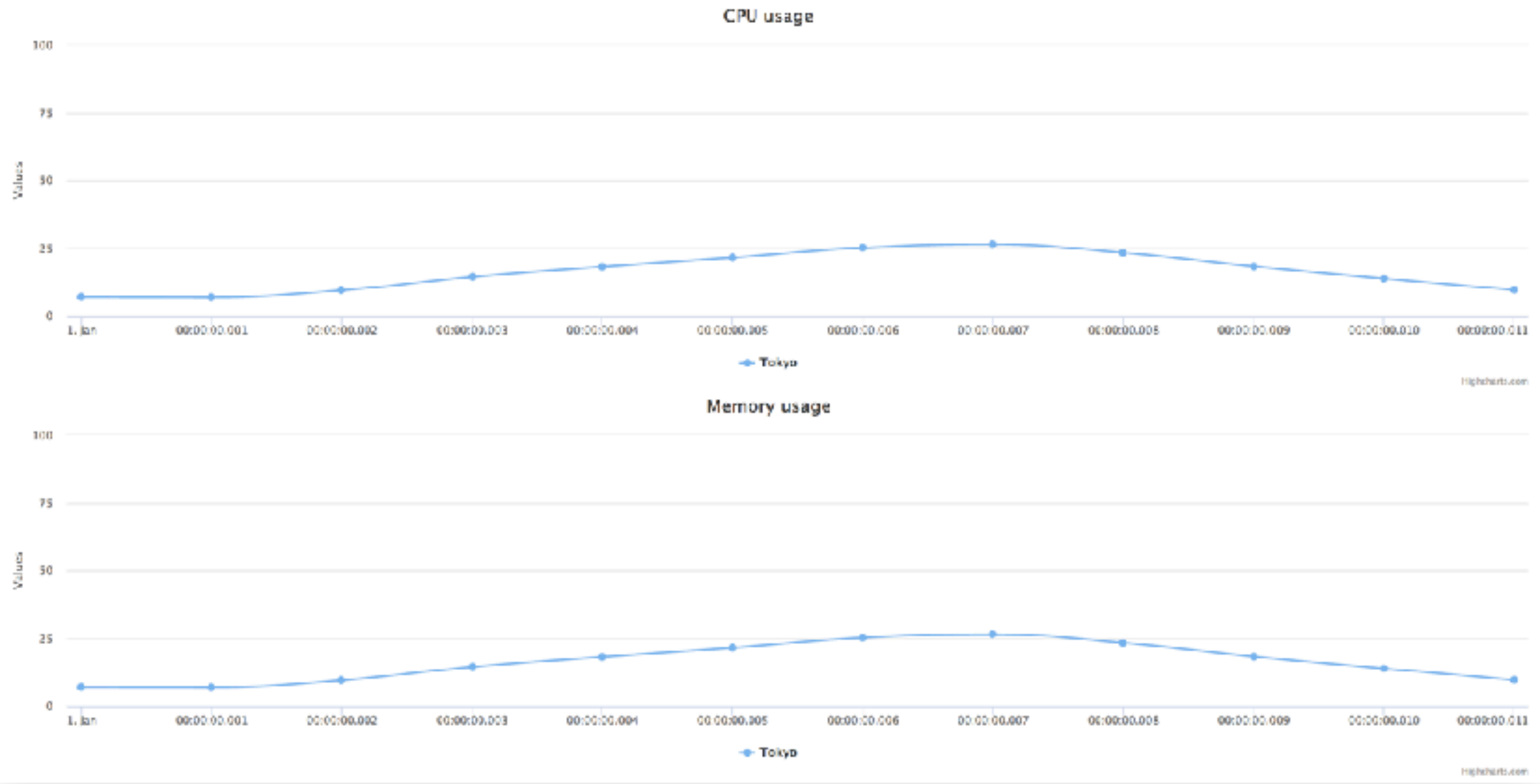
$bower install



```
bower not-cached    https://github.com/jashkenas/underscore.git#~1.8.3
bower resolve       https://github.com/jashkenas/underscore.git#~1.8.3
bower not-cached    https://github.com/highcharts/highcharts-dist.git#~5.0.7
bower resolve       https://github.com/highcharts/highcharts-dist.git#~5.0.7
bower not-cached    https://github.com/jquery/jquery-dist.git#~3.1.1
bower resolve       https://github.com/jquery/jquery-dist.git#~3.1.1
bower checkout      highcharts#v5.0.7
bower checkout      underscore#1.8.3
bower checkout      jquery#3.1.1
bower resolved      https://github.com/highcharts/highcharts-dist.git#5.0.7
bower resolved      https://github.com/jquery/jquery-dist.git#3.1.1
bower resolved      https://github.com/jashkenas/underscore.git#1.8.3
bower install       highcharts#5.0.7
bower install       jquery#3.1.1
bower install       underscore#1.8.3

highcharts#5.0.7 bower_components/highcharts

jquery#3.1.1 bower_components/jquery

underscore#1.8.3 bower_components/underscore
```

Try to use Highchart

index.html

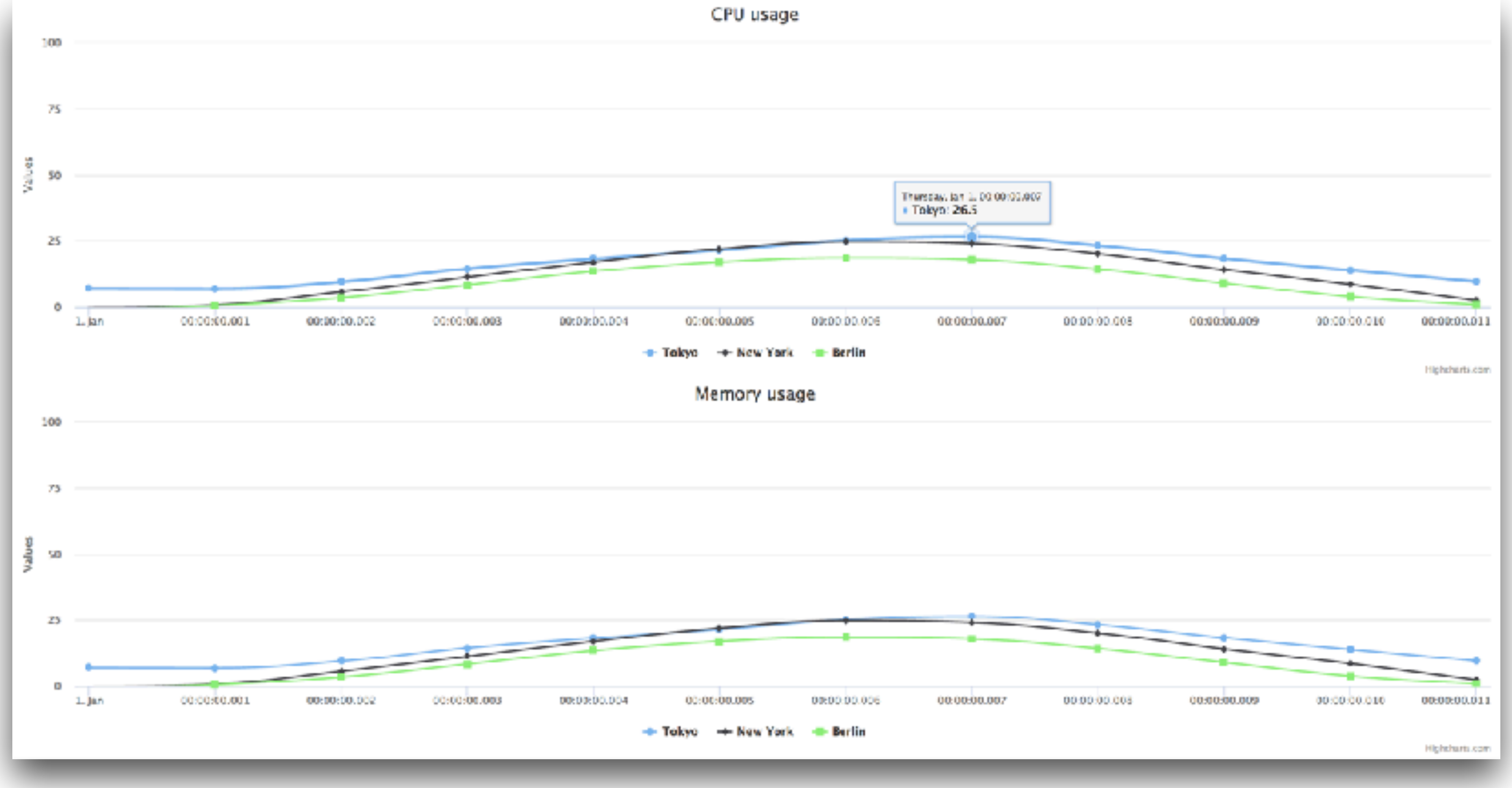


Series of data

```
series: [{
        name: 'Tokyo',
        data: [7.0, 6.9, 9.5, 14.5, 18.2, 21.5, 25.2, 26.5, 23.3, 18.3, 13.9, 9.6]
    }, {
        name: 'New York',
        data: [-0.2, 0.8, 5.7, 11.3, 17.0, 22.0, 24.8, 24.1, 20.1, 14.1, 8.6, 2.5]
    }, {
        name: 'Berlin',
        data: [-0.9, 0.6, 3.5, 8.4, 13.5, 17.0, 18.6, 17.9, 14.3, 9.0, 3.9, 1.0]
    }
```

Add more series

index.html



Create REST APIs

Using express

Name	Enpoint	HTTP Method
CPU usage	/api/cpu	GET
Memory usage	/api/memory	GET

Manage library with npm

create file package.json

```
{
  "name": "monitoring",
  "version": "0.0.0",
  "dependencies": {
    "jquery": "~3.1.1",
    "highcharts": "~5.0.7",
    "express": "~4.14.1"
  }
}
```

Install library

```
$npm install
```

Create REST Server

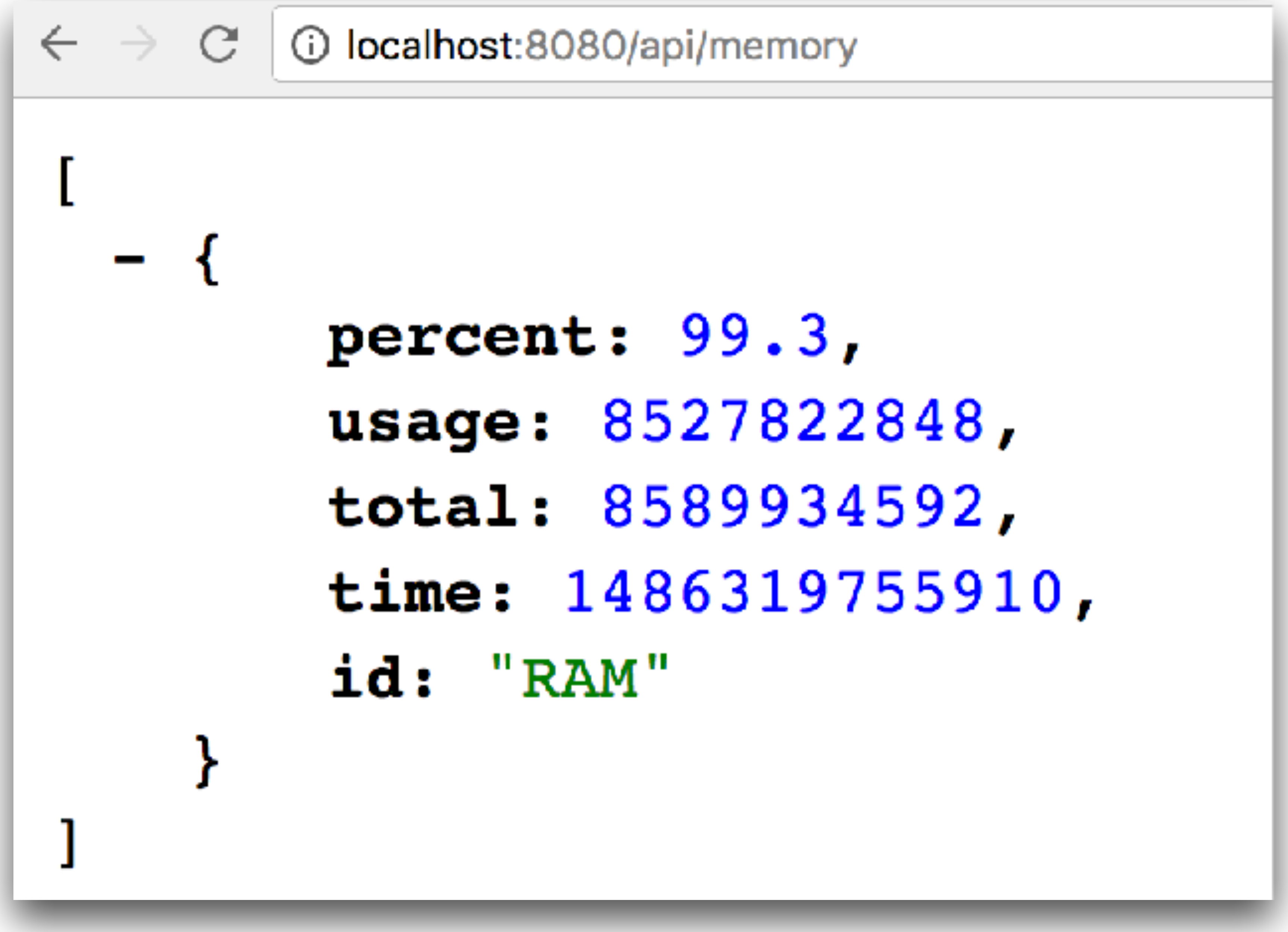
app.js

```
var express = require('express');
var app = express();
app.use(express.static(__dirname));

app.listen(8080);
console.log('Listening on port 8080');
```

Create /api/memory

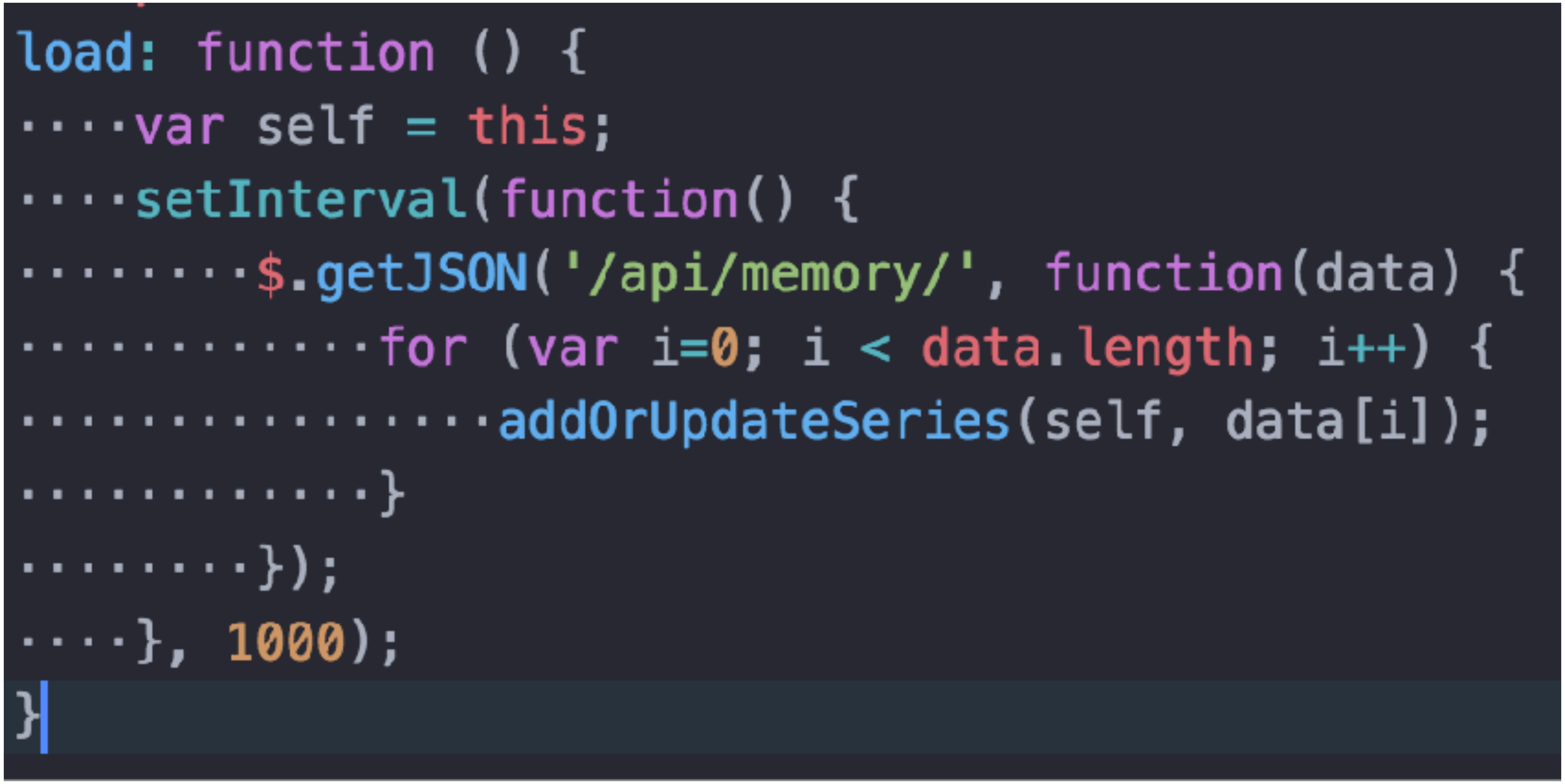
```
var os = require('os');

app.get('/api/memory/', function(request, response) {
    var timestamp = new Date();
    var free = os.freemem();
    var total = os.totalmem();
    var used = total - free;

    response.json([{
        'percent': parseFloat(((used * 100) /
total).toFixed(1)),
        'usage': used,
        'total': total,
        'time': timestamp.getTime(),
        'id': 'RAM'
    }]);
});
```

Testing



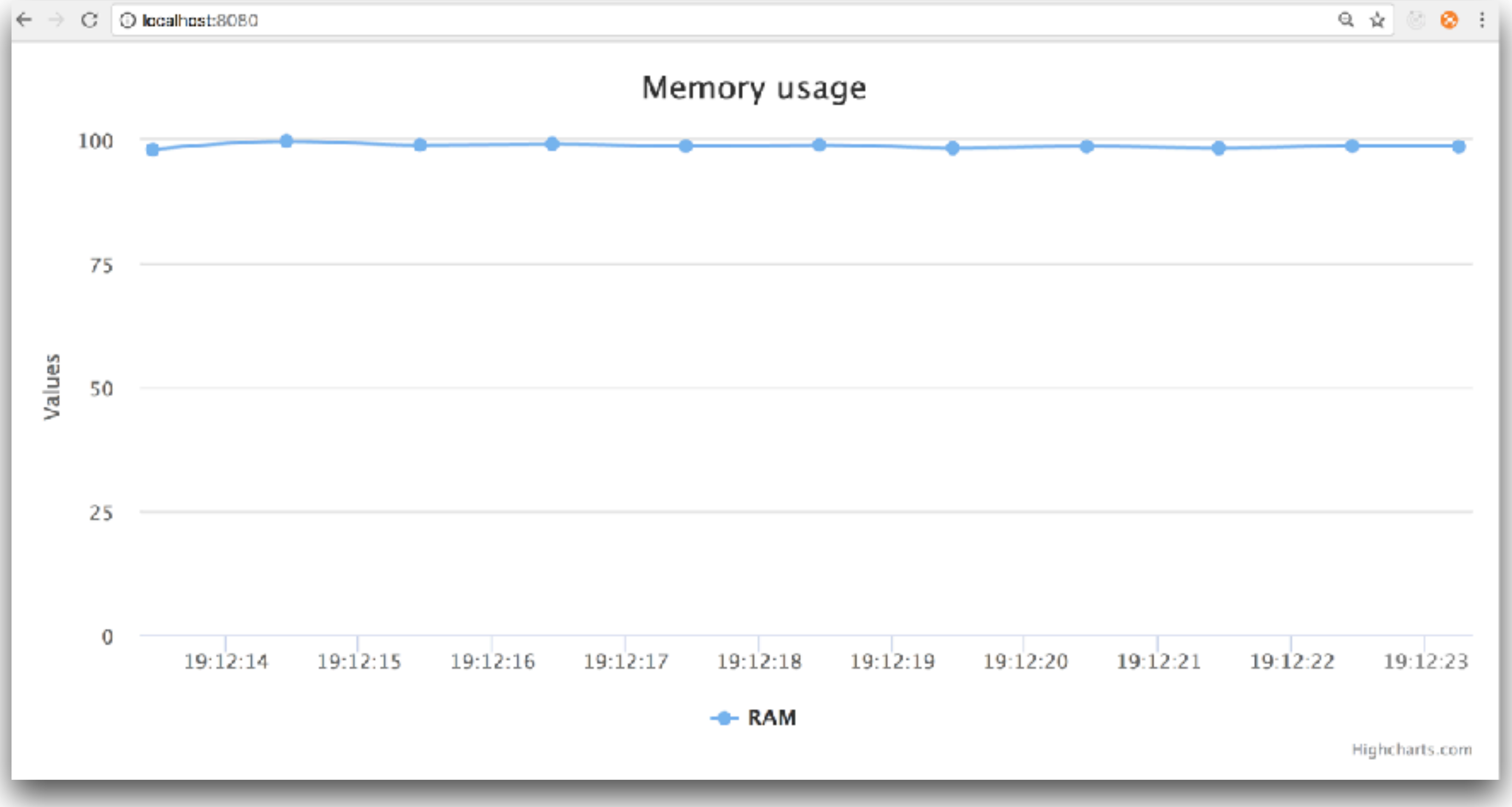
```
[
  - {
        percent: 99.3,
        usage: 8527822848,
        total: 8589934592,
        time: 1486319755910,
        id: "RAM"
    }
]
```

Integrate with frontend



```
load: function () {
    var self = this;
    setInterval(function() {
        $.getJSON('/api/memory/', function(data) {
            for (var i=0; i < data.length; i++) {
                addOrUpdateSeries(self, data[i]);
            }
        });
    }, 1000);
}
```

Memory usage



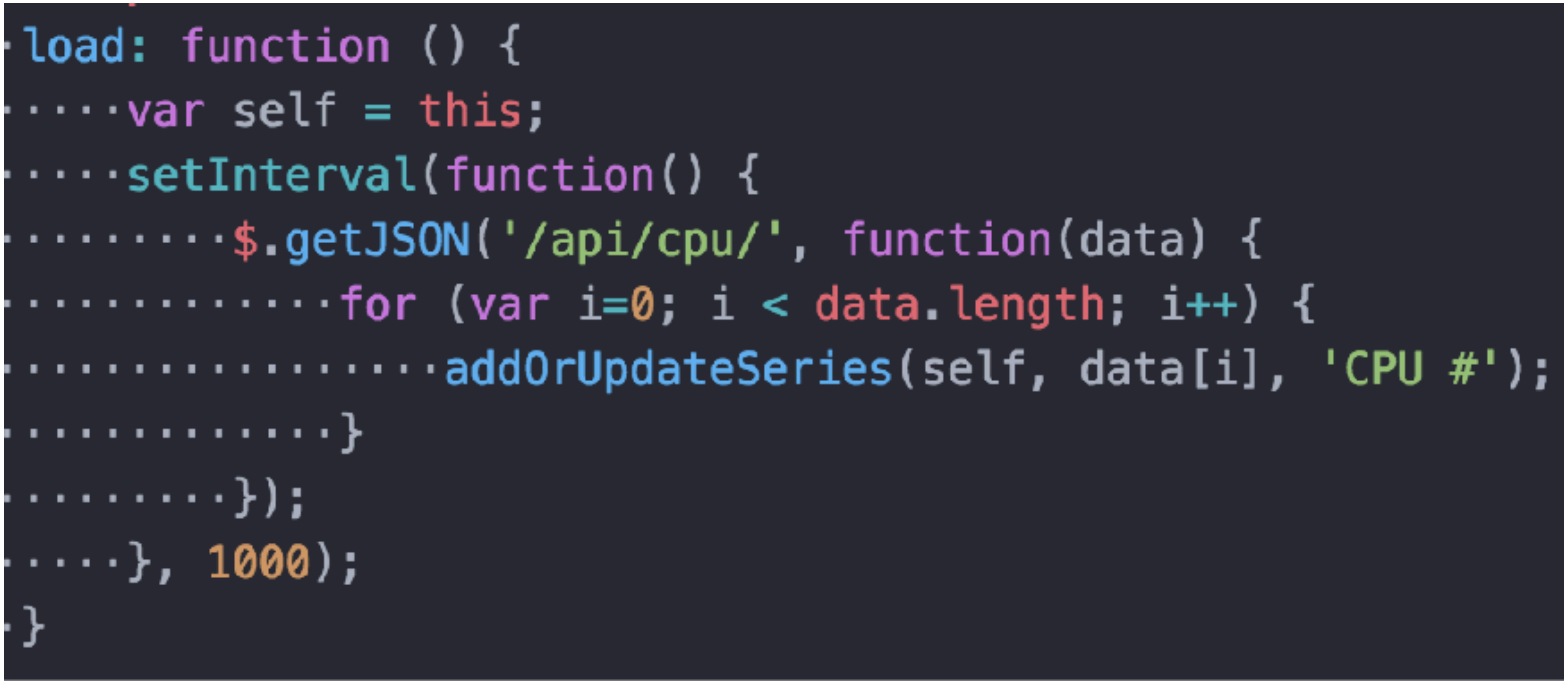
Create /api/cpu

```
app.get('/api/cpu/', function(request, response) {
    response.json(get_cpu_percentages());
});
```

Create /api/cpu

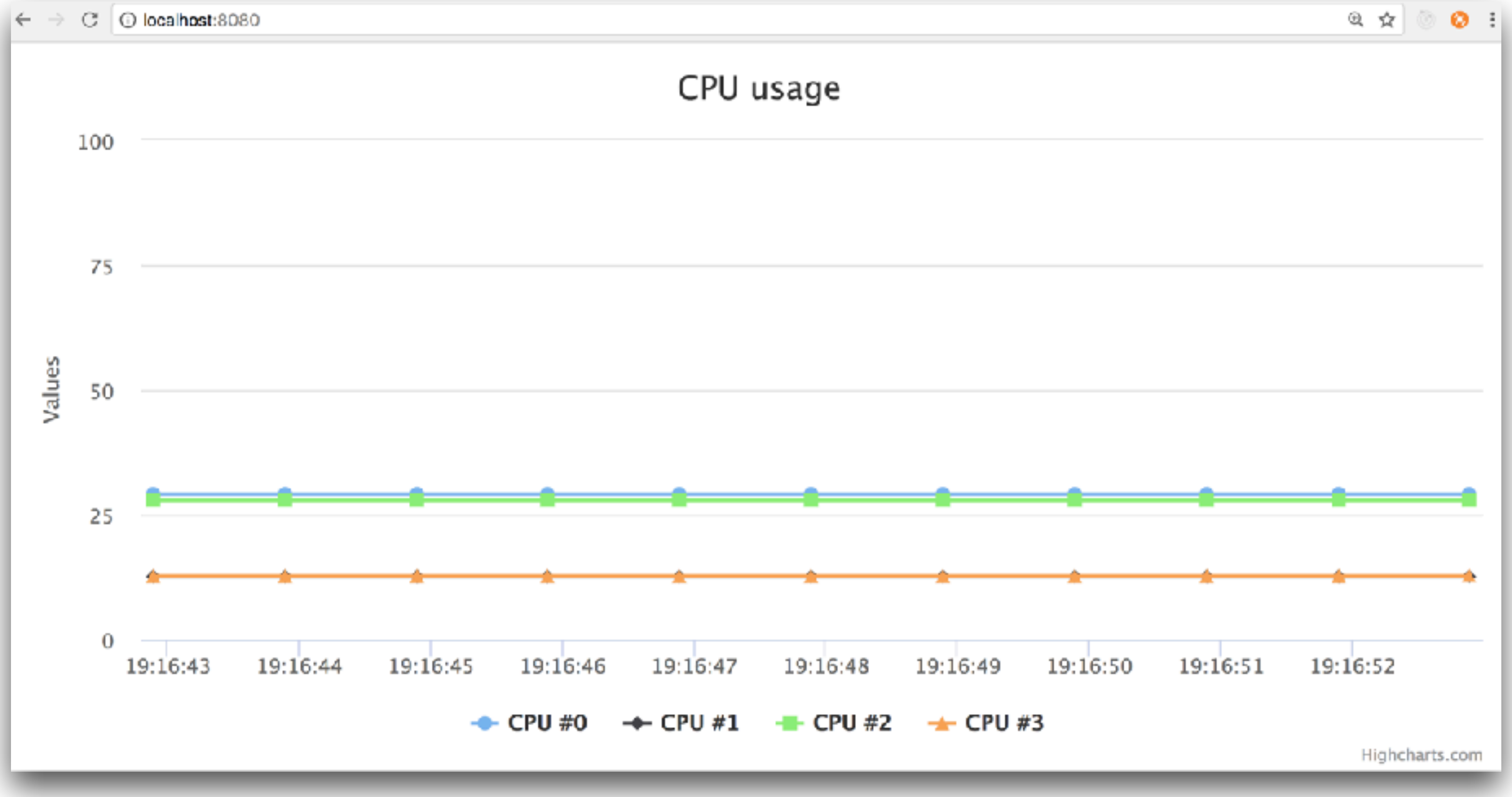
```
var get_cpu_percentages = function(time) {
    var cpus = os.cpus();
    var timestamp = time || new Date();

    var cpu_percentages = underscore.map(cpus, function(cpu, key) {
        var values = underscore.values(cpu.times);
        var total = underscore.reduce(values, sum, 0);
        var idle = cpu.times.idle;

        return {
            'percent': parseFloat((((total - idle) * 100) / total).toFixed(1)),
            'usage': (total - idle),
            'total': total,
            'time': timestamp.getTime(),
            'id': key
        }
    });

    return cpu_percentages;
};
```

Integrate with frontend



```
load: function () {
    var self = this;
    setInterval(function() {
        $.getJSON('/api/cpu/', function(data) {
            for (var i=0; i < data.length; i++) {
                addOrUpdateSeries(self, data[i], 'CPU #');
            }
        });
    }, 1000);
}
```

CPU usage


localhost:8080
CPU usage
Values
100
75
50
25
0
19:16:43
19:16:44
19:16:45
19:16:46
19:16:47
19:16:48
19:16:49
19:16:50
19:16:51
19:16:52
CPU #0
CPU #1
CPU #2
CPU #3
Highcharts.com