**Hello CPUJS !!**

**Hello CPUJS !!**

1. **如何获取到CPUJS**

Github地址：

<https://github.com/grz/cpujs>

1. **如何使用CPUJS**

CPUJS依赖seajs模块加载器：

require([' /cpu'],function(cpujs){

cpujs.bootstrap();*//cpujs初始化*

cpujs.load(\*\*\*);*//加载一个任务*

cpujs.fastLoad(\*\*\*);*//快速加载一个任务，享受最高的执行优先级*

cpujs.load(

[Fn1,fn2,fn3]

);*//加载一批任务，并按序执行，不需要单独传参时可以用这种*

cpujs.load(fn,arguments);*//加载一个任务，并把arguments传给这个函数*

});

1. **如何使用CPUJS来统计页面卡的情况**

require([' /cpu'],function(cpujs){

cpujs.bootstrap();*//cpujs初始化*

cpujs.Stat.doCPUStat([0,15000],report); *//统计从现在开始15秒内的cpu开销情况*

});

Function report(**dataArray**){

…

}

调用cpujs.Stat.doCPUStat接口之后，CPUJS回去采集数据，等到点的时候会把数据整理好回传给回调函数，dataArray包含的数据项如下：

|  |  |
| --- | --- |
| dataArray下标 | 含义 |
| 0 | **CPU100%的次数（卡住）,这个数据可以反应出一段时间内卡的情况** |
| 1 | CPU大于等于90%的次数 |
| 2 | CPU大于等于80%的次数 |
| 3 | CPU大于等于70%的次数 |
| 4 | CPU大于等于60%的次数 |
| 5 | CPU大于等于50%的次数 |
| 6 | CPU大于等于40%的次数 |
| 7 | CPU大于等于30%的次数 |
| 8 | CPU大于等于20%的次数 |
| 9 | CPU大于等于10%的次数 |
| 10 | CPU大于等于0%的次数，一段时间内统计的总点数 |
| 11 | **CPU总体开销，Monitor曲线的面积，可以用来衡量一段时间内的运行开销** |
| 12 | **CPU100%占比，衡量一段时间内卡住的时间占总体时间的百分比** |