**JMeter：生成漂亮的多维度的HTML报告**

2017年08月24日 18:42:26 [wuyou10206](https://me.csdn.net/wuyou10206) 阅读数：9961 标签： [测试](http://so.csdn.net/so/search/s.do?q=%E6%B5%8B%E8%AF%95&t=blog)[性能测试](http://so.csdn.net/so/search/s.do?q=%E6%80%A7%E8%83%BD%E6%B5%8B%E8%AF%95&t=blog)[jmeter](http://so.csdn.net/so/search/s.do?q=jmeter&t=blog)[html](http://so.csdn.net/so/search/s.do?q=html&t=blog)[报告](http://so.csdn.net/so/search/s.do?q=%E6%8A%A5%E5%91%8A&t=blog)更多

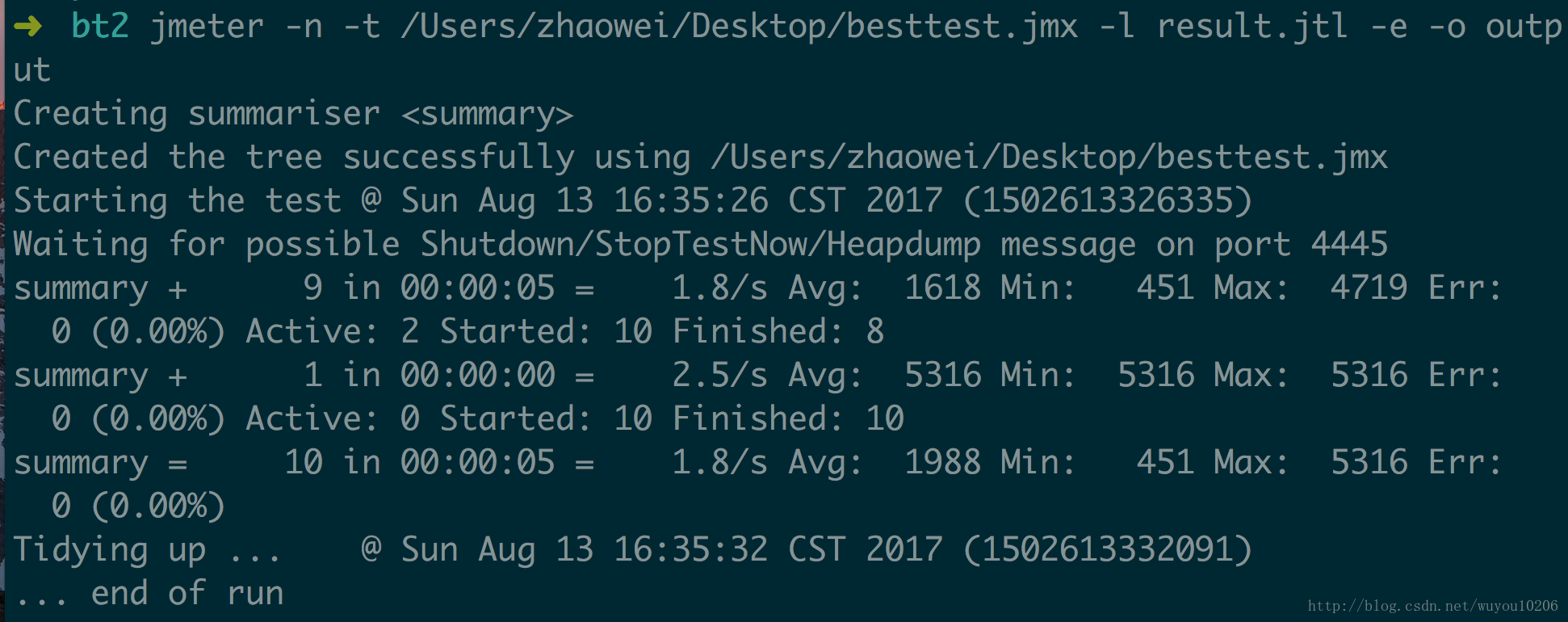
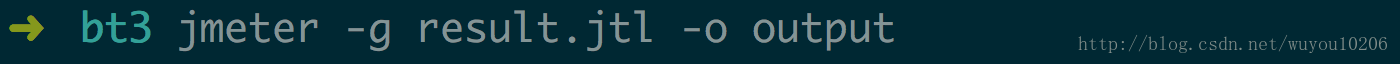
个人分类： [软件测试](https://blog.csdn.net/wuyou10206/article/category/7125663)

**JMeter：生成漂亮的多维度的HTML报告**

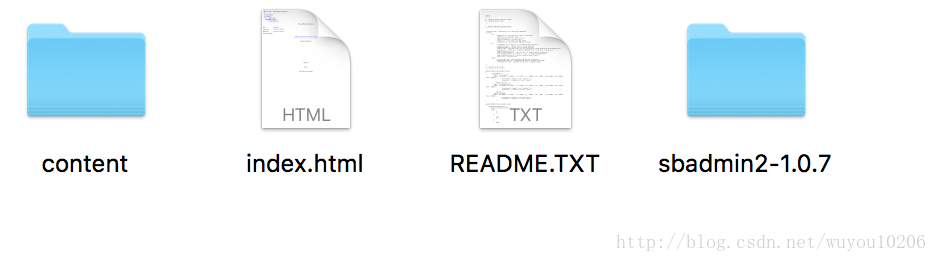
我们做性能测试的时候会经常使用一些性能测试工具，我个人比较喜欢Jmeter这个工具，但是JMeter这个工具在生成测试报告方面一直有所欠缺。但是JMeter在3.0以后解决了这个问题。

JMeter3.0以后引入了Dashboard Report，用于生成HTML页面格式图形化报告的扩展模块，但是请大家注意一个问题JMeter3.0直接生成报告会出现中文乱码，建议大家使用3.1之后的版本，我这里使用的是3.2。

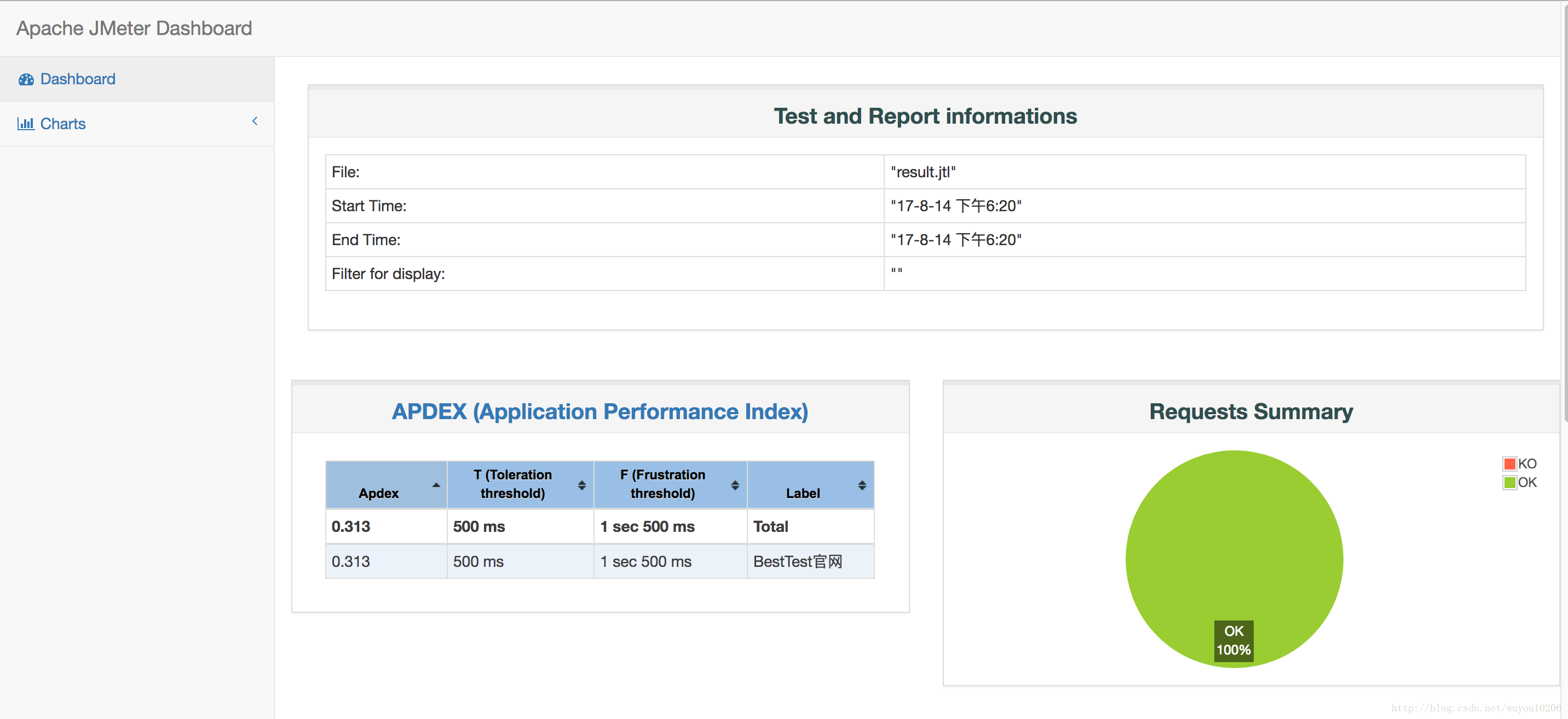
那么是如何生成HTML报告的呢，我们一起来看一下。

一、如果没有.jtl文件，运行如下命令：   
jmeter -n -t source.jmx -l result.jtl -e -o /tmp/ResultReport   
我们来看一下这条命令的参数：   
-n ：以非GUI形式运行Jmeter   
-t ：source.jmx 脚本路径   
-l ：result.jtl 运行结果保存路径（.jtl）,此文件必须不存在   
-e ：在脚本运行结束后生成html报告   
-o ：用于存放html报告的目录   
在这之前呢大家需要自己准备好JMeter的脚本，如下是我执行命令截图：   
  
二、如果已经存在结果文件（.jtl）,可运行如下命令生成报告   
jmeter -g result.jtl -o /tmp/ResultReport   
参数：   
-g : result.jtl 已经存在的.jtl文件的路径。   
-o ：用于存放html报告的目录   
这种没有日志打印，下面是截图：   


无论采用哪种方式，执行完后会在执行的目录生成如下文件或文件夹：

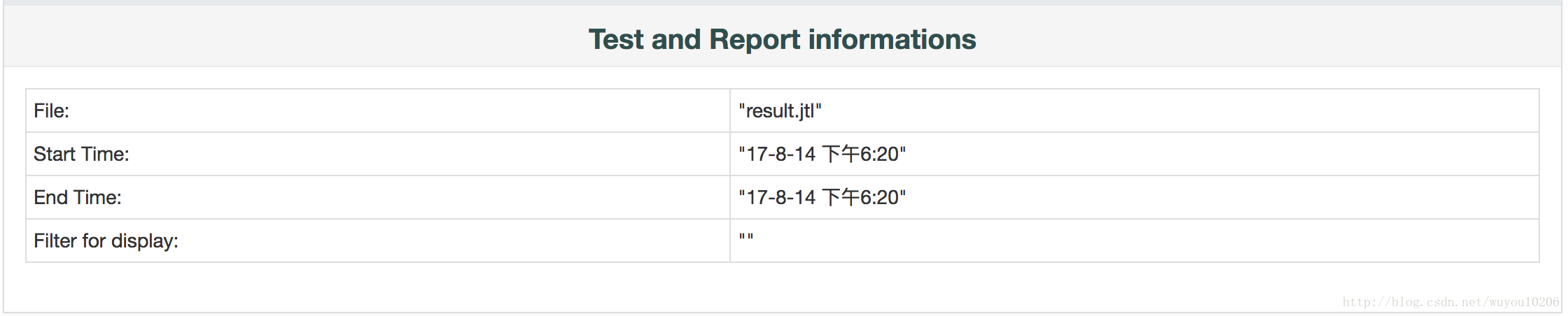
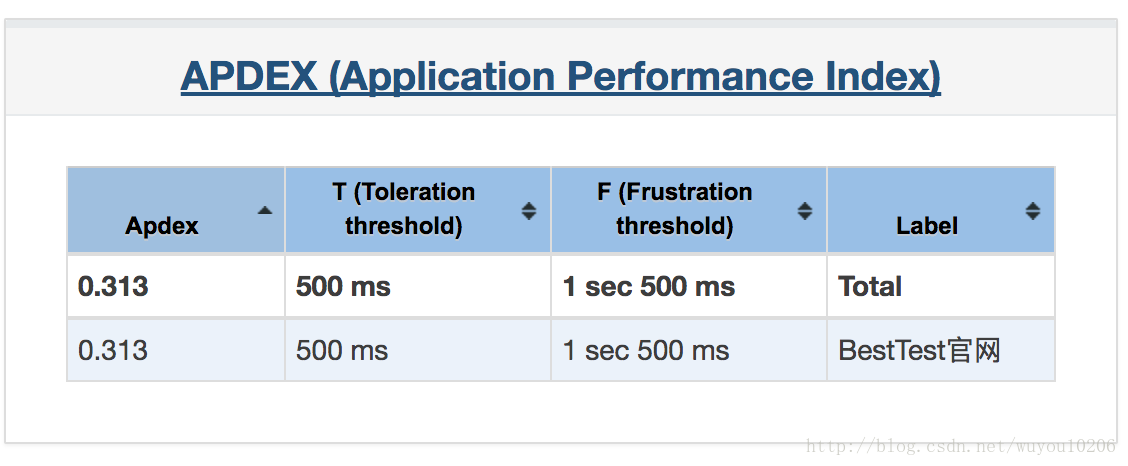
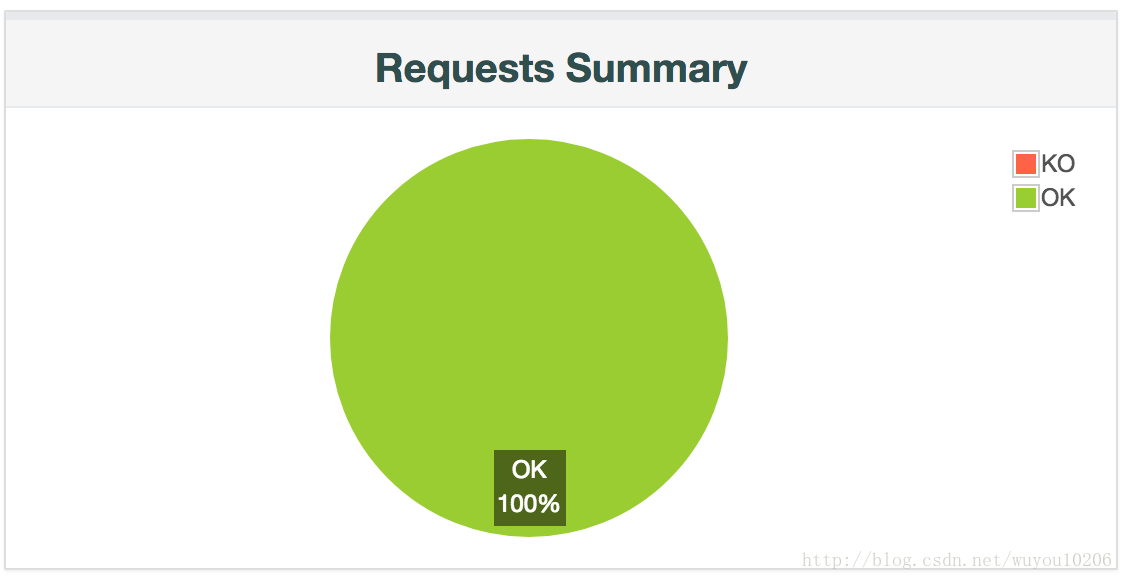
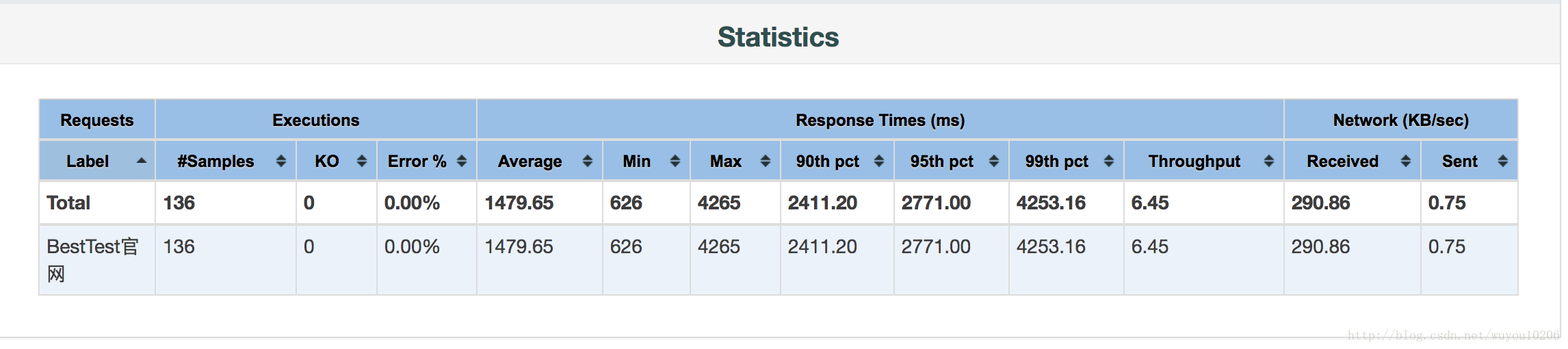
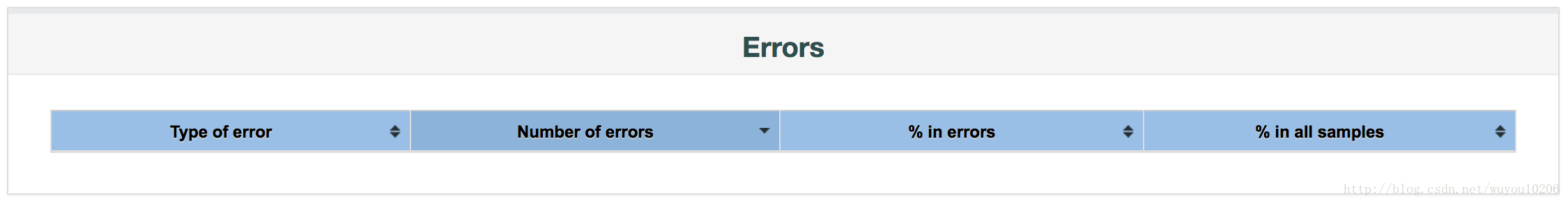
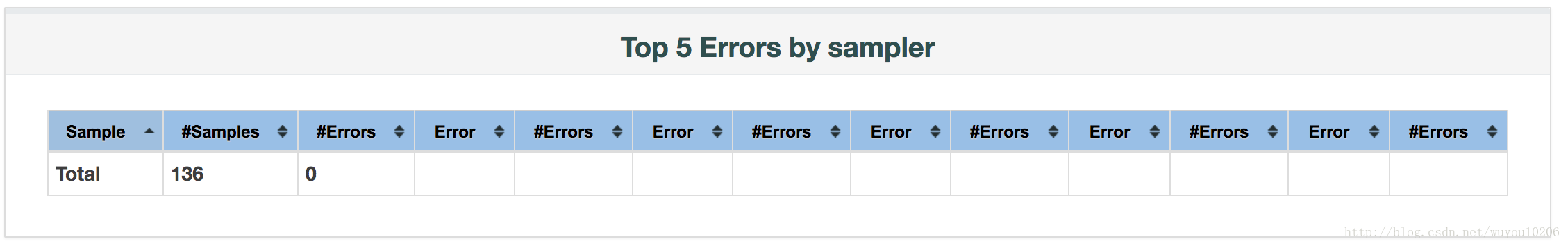


双击打开index.html文件，就可以看到报告了。如图：



**这个报告总体分为两部分：Dashboard和Charts。**

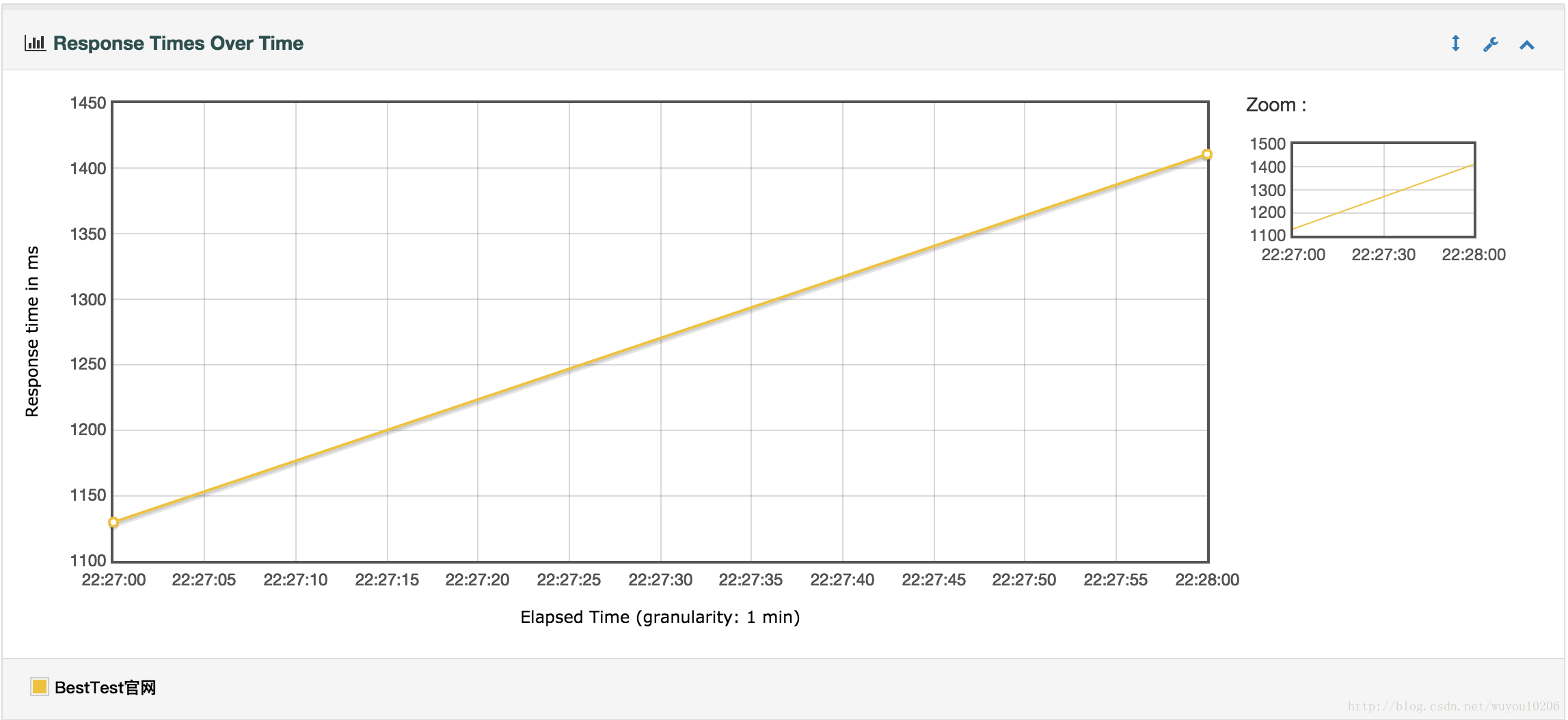
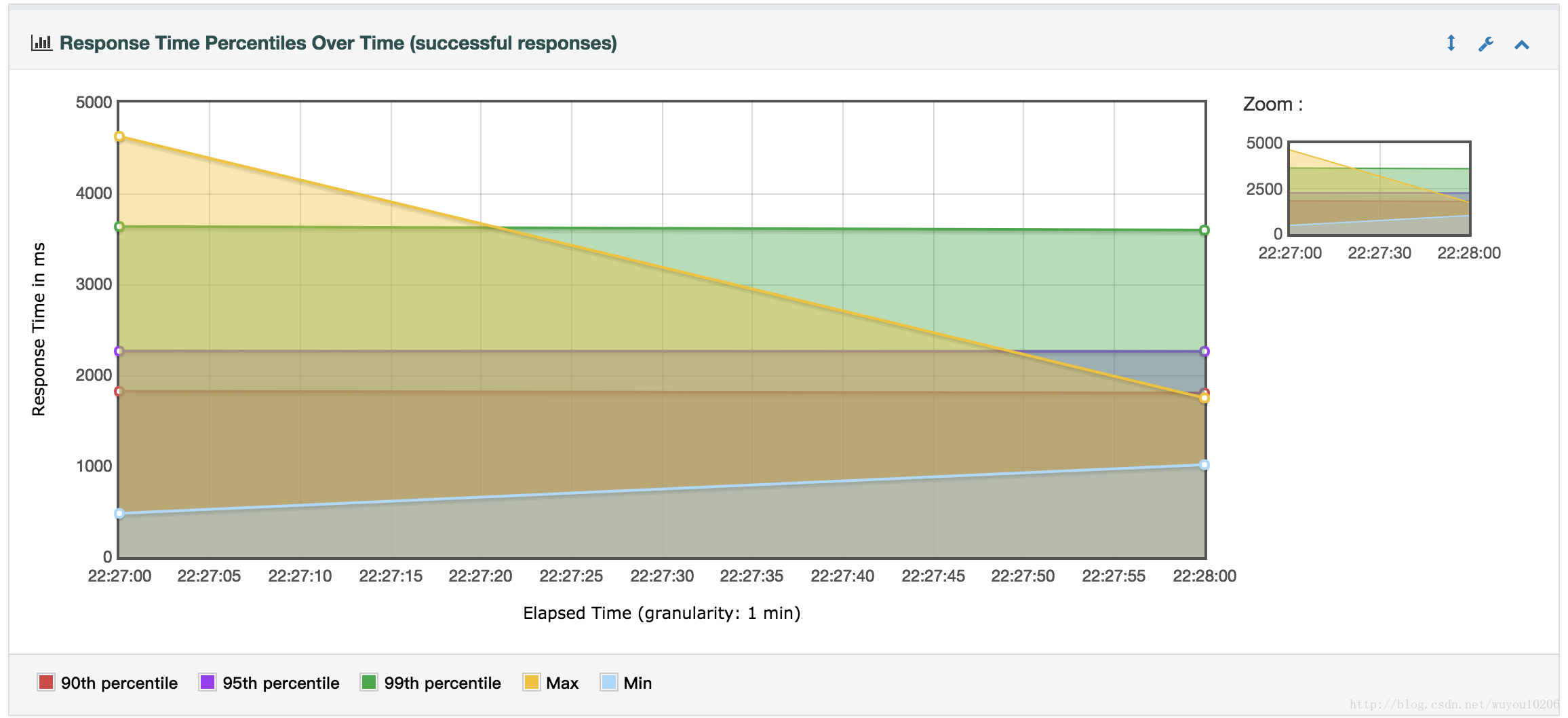
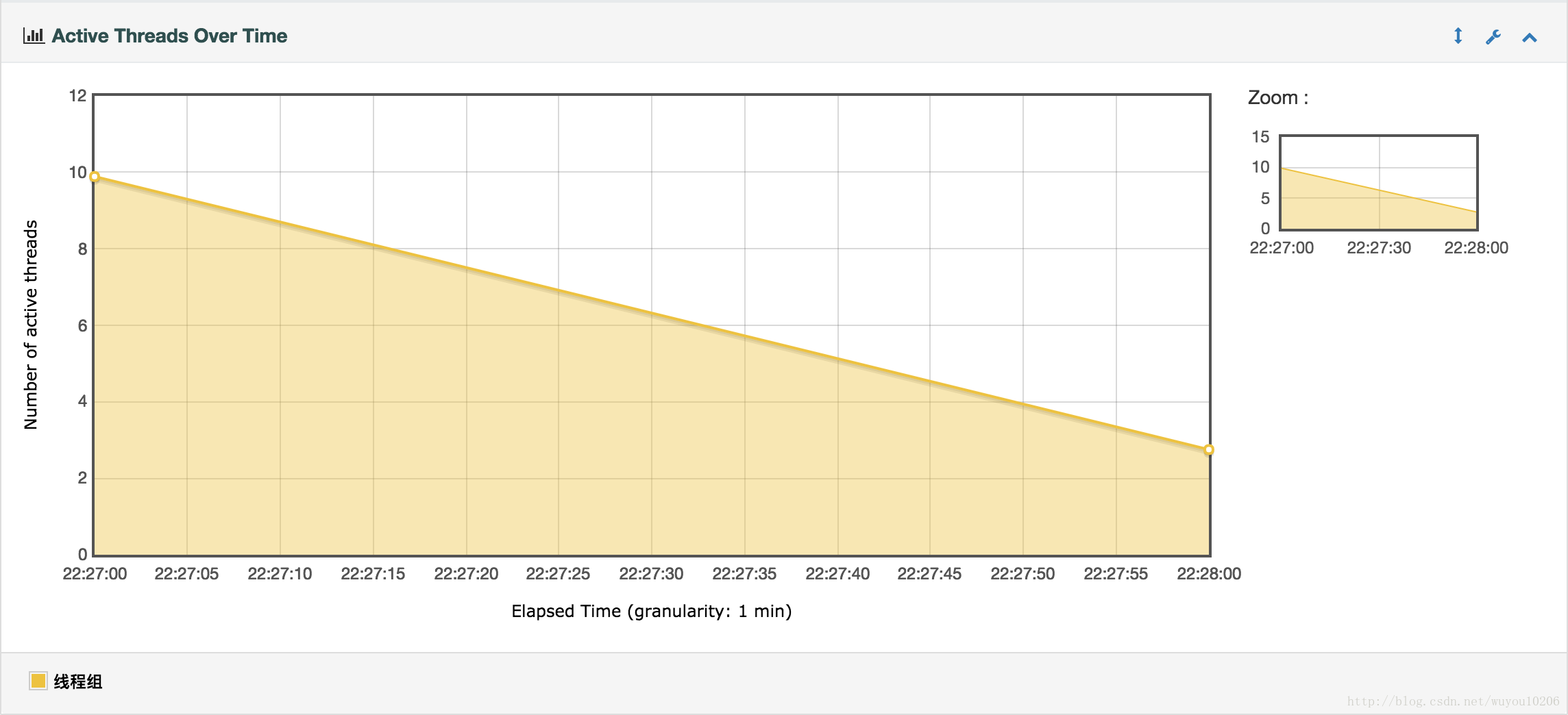
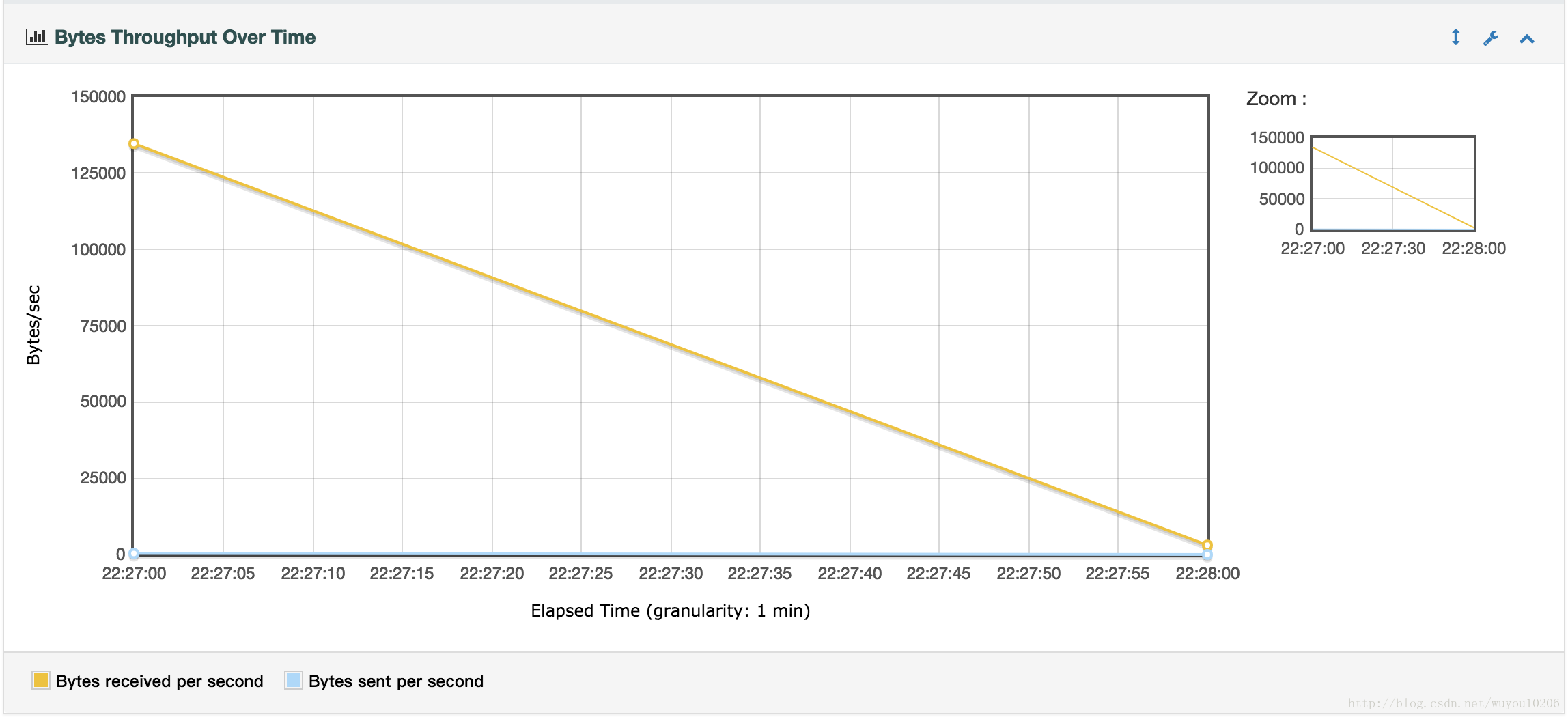
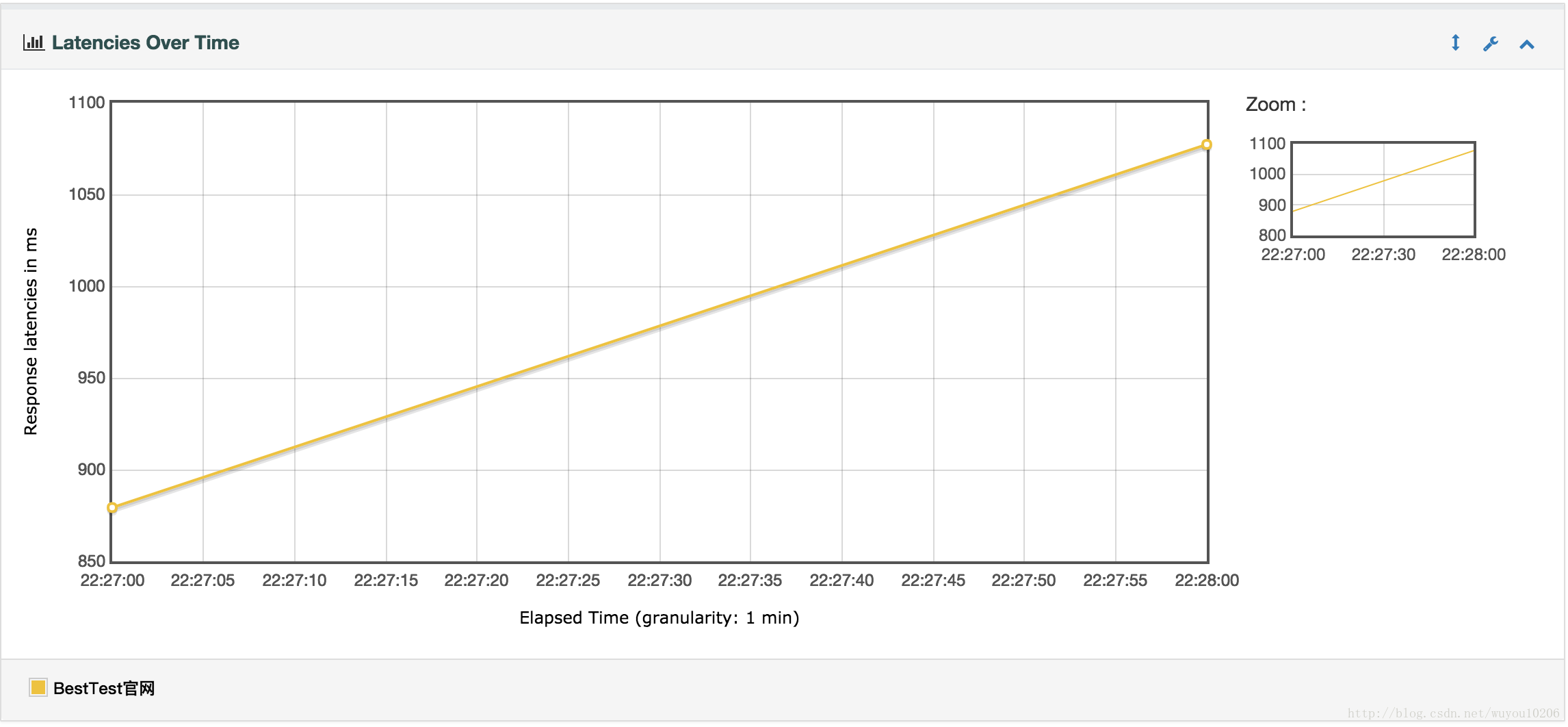
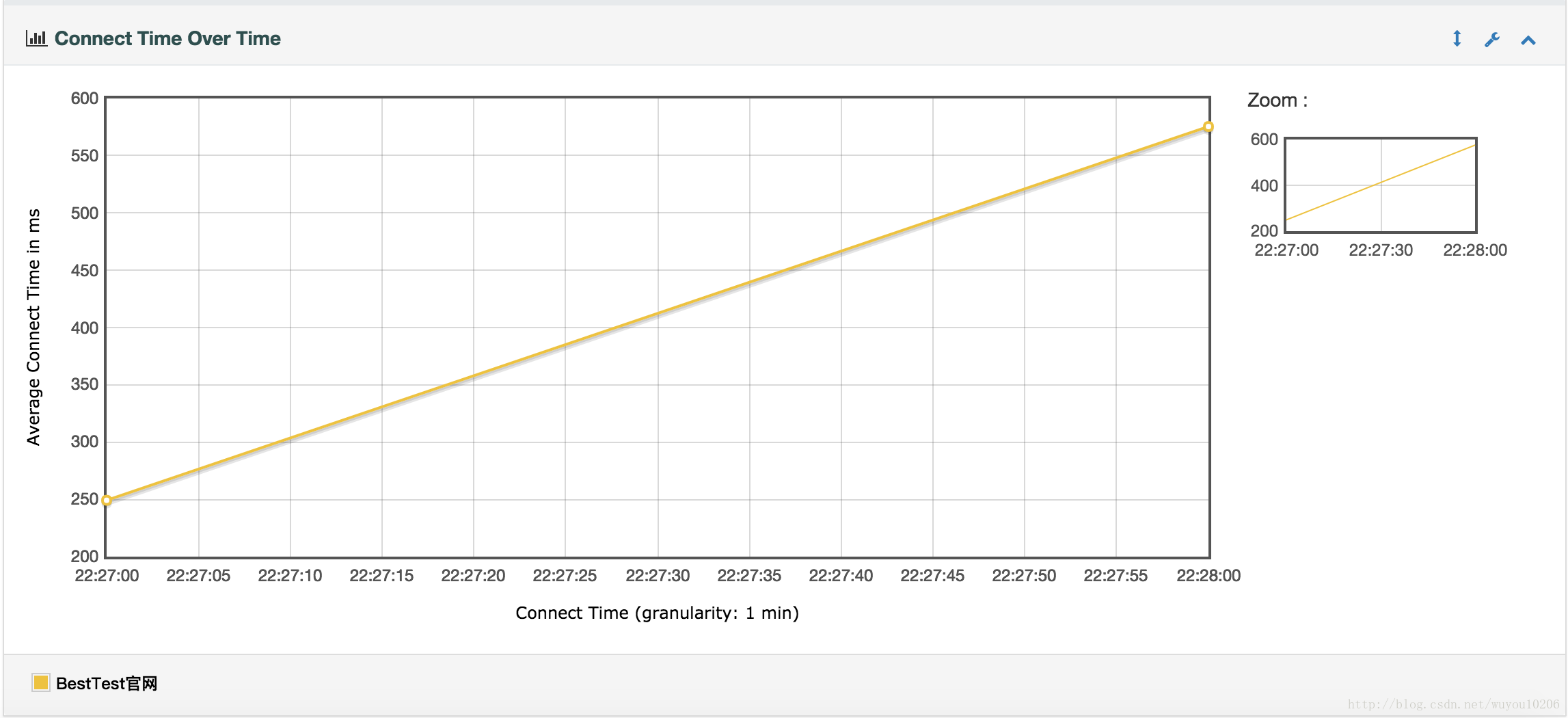
**先看一下Dashboard中都有什么？**

Test and Report informations：测试和报告信息: 测试结果保存文件/测试开始时间/测试结束时间/展示过滤器。   
  
APDEX(Application Performance Index)：应用程序性能满意度的标准，范围在 0-1之间，1表示达到所有用户均满意，可以在配置文件设置。   
  
Requests Summary: 请求的通过率(OK)与失败率(KO)，百分比显示。   
  
Statistics: 数据分析，基本将 Summary Report 和 Aggrerate Report 的结果合并。   
  
Errors: 错误情况，依据不同的错误类型，将所有错误结果展示。   
  
Top 5 Errors by sampler：Top5错误信息采样   


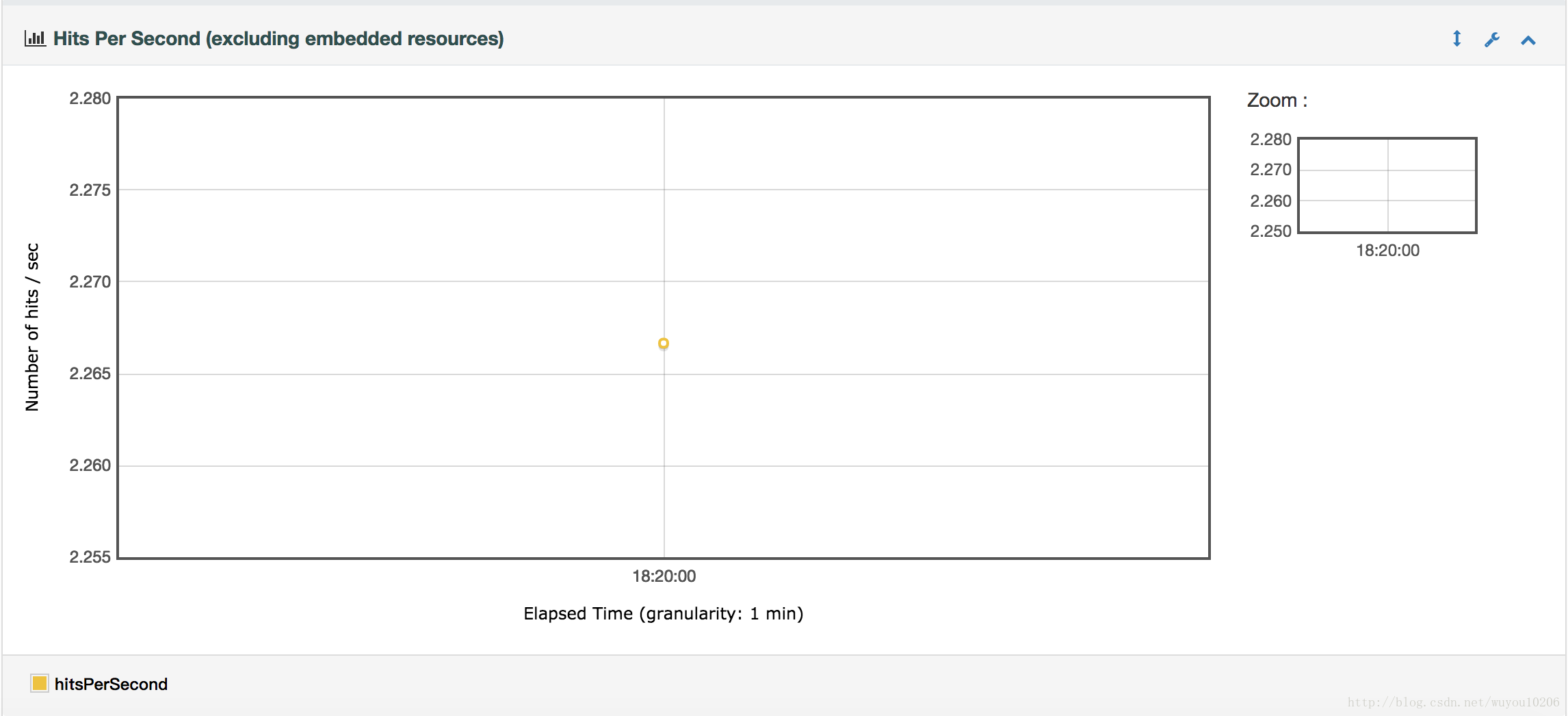
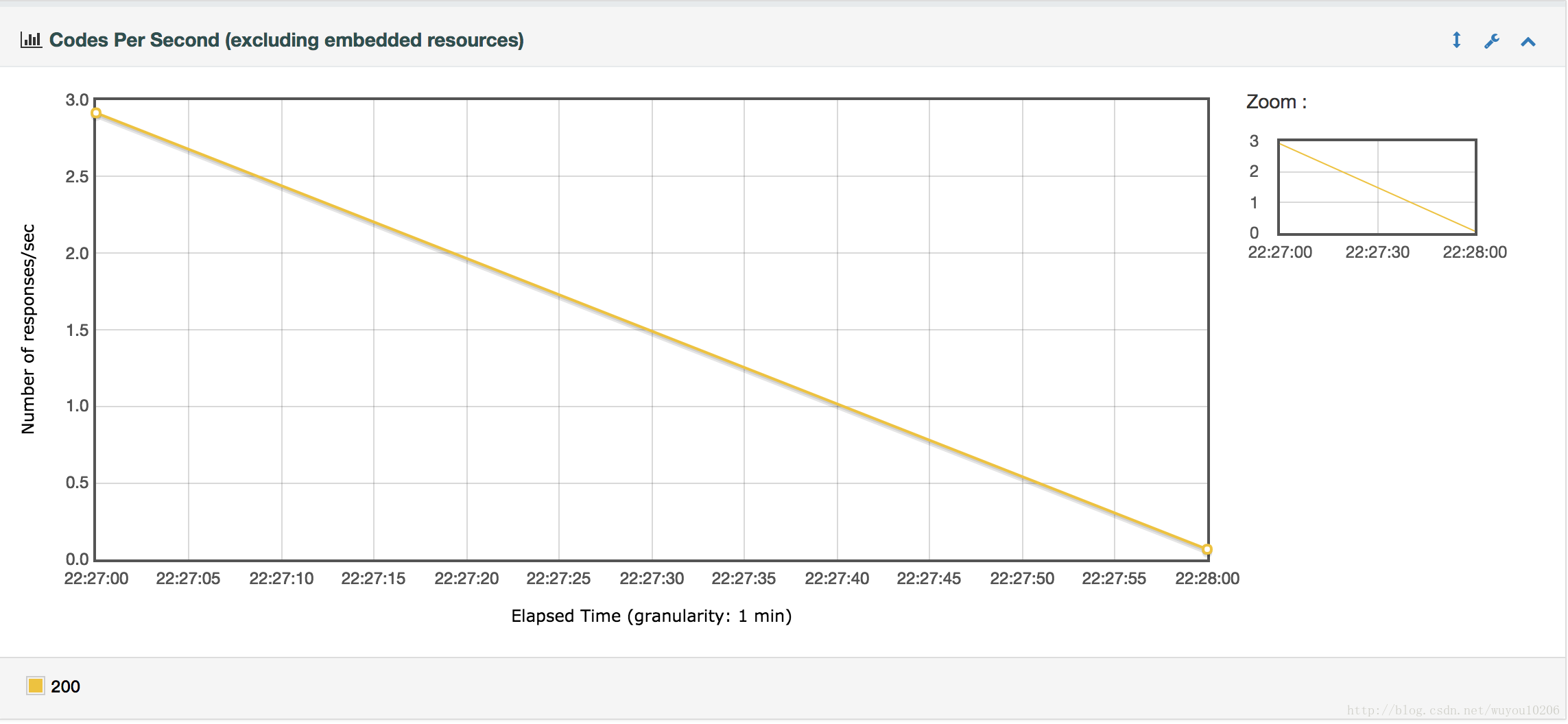
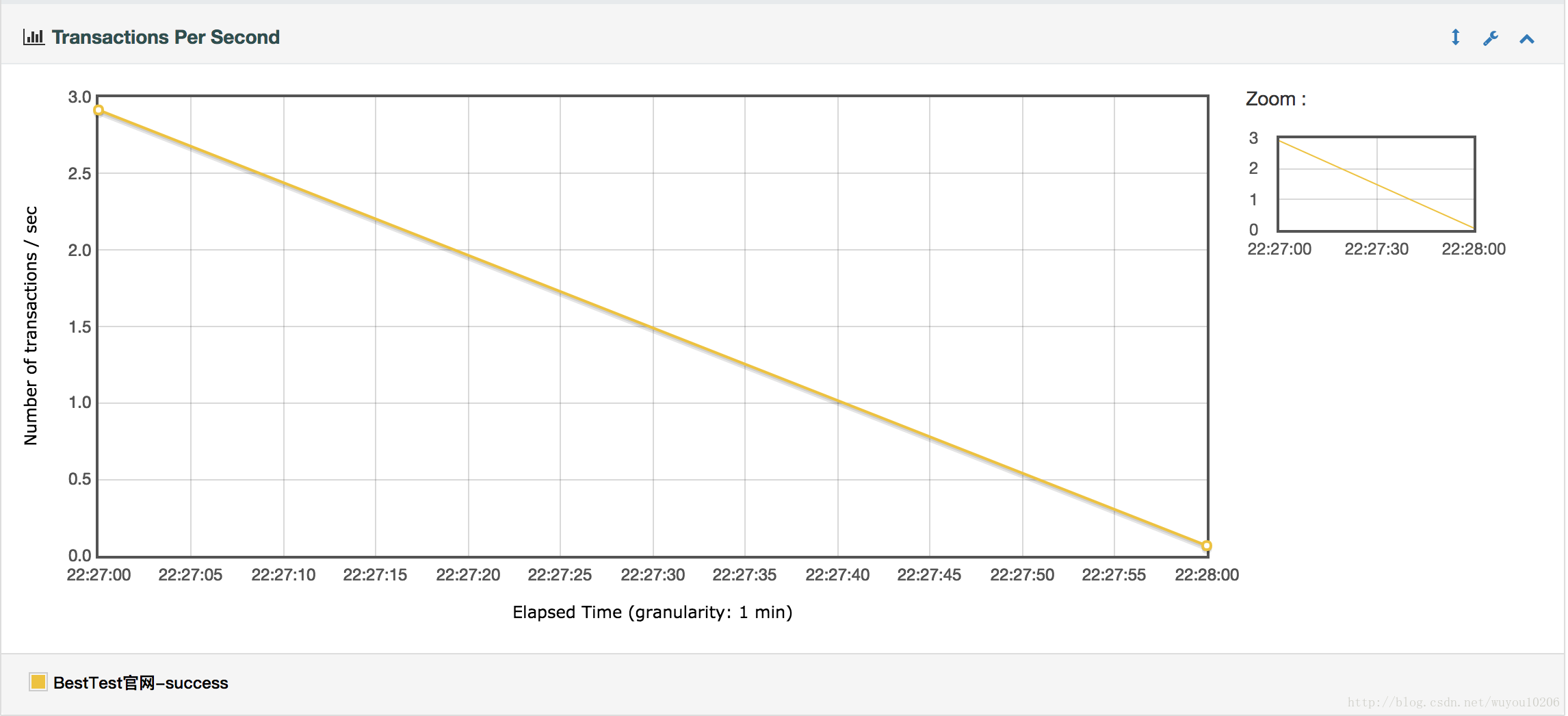
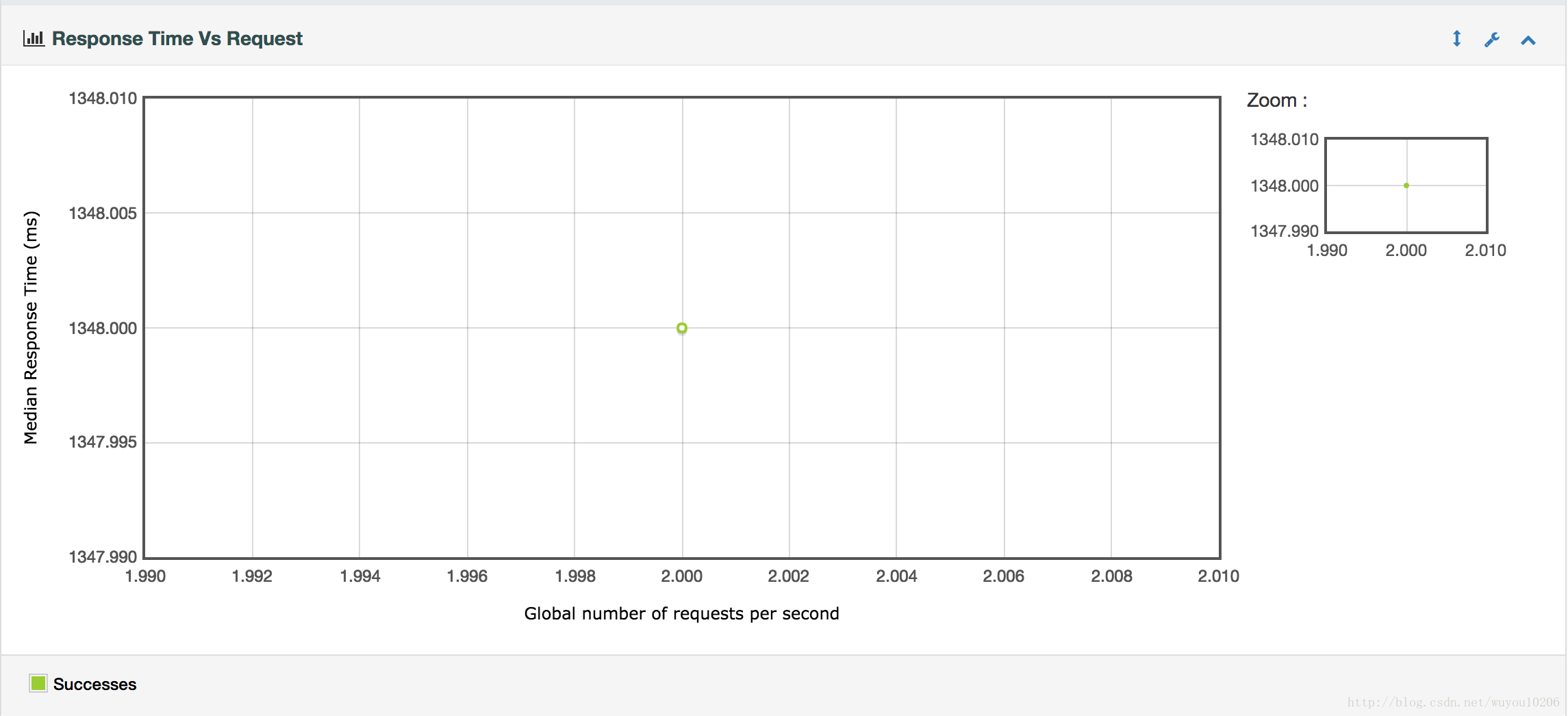
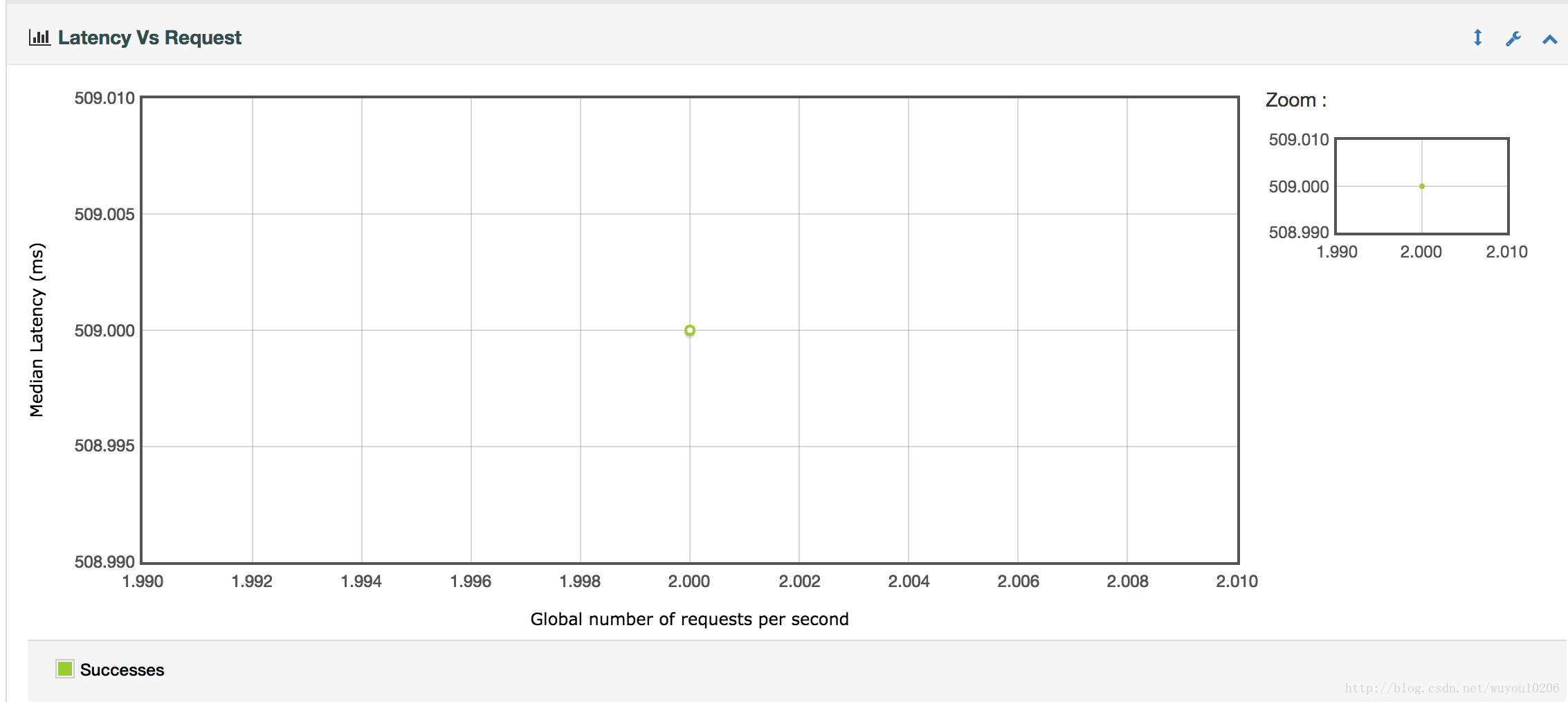
**接下来看一下Charts里的内容：**

Charts里如图所示分为三大块：时间维度信息(Over Time)、吞吐量(Thorughput)、响应时间(Response Times)

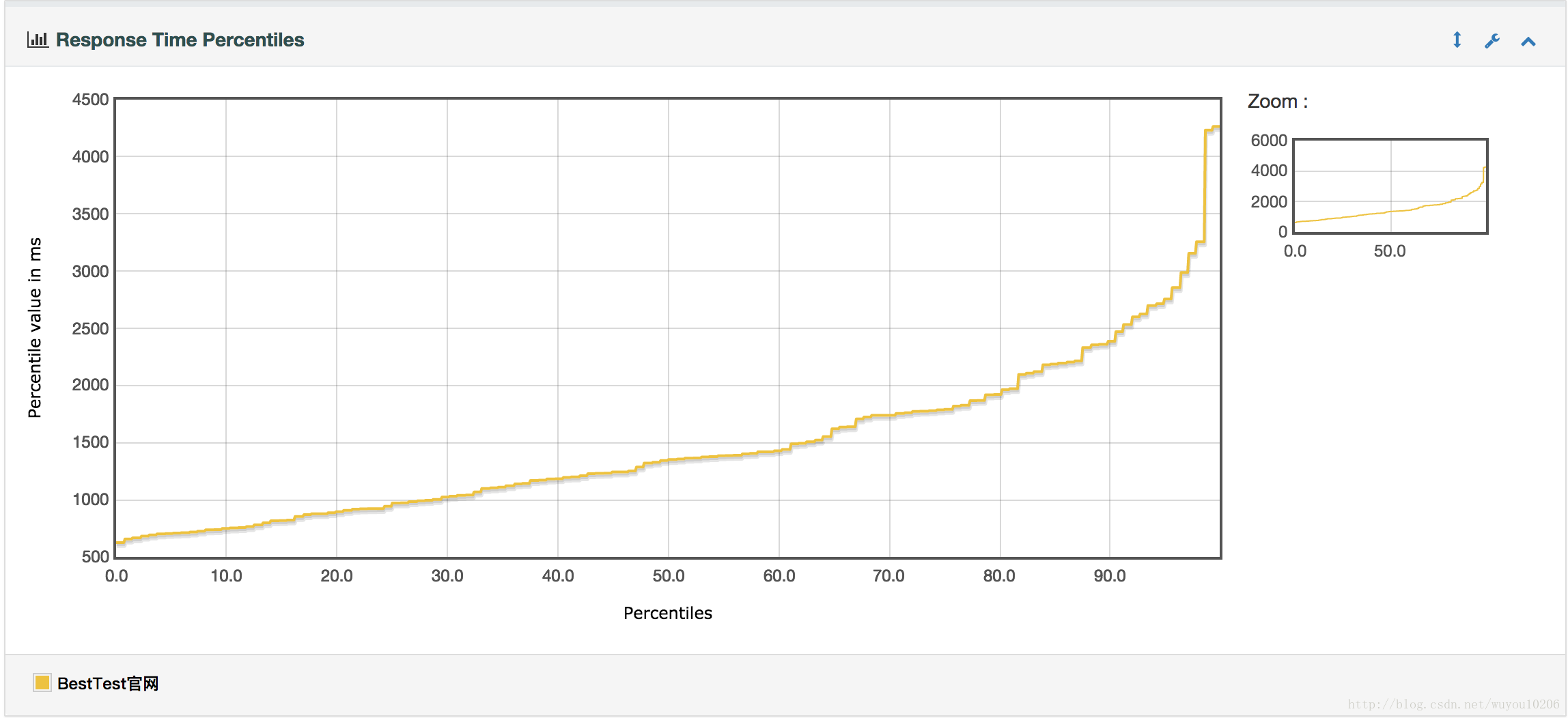
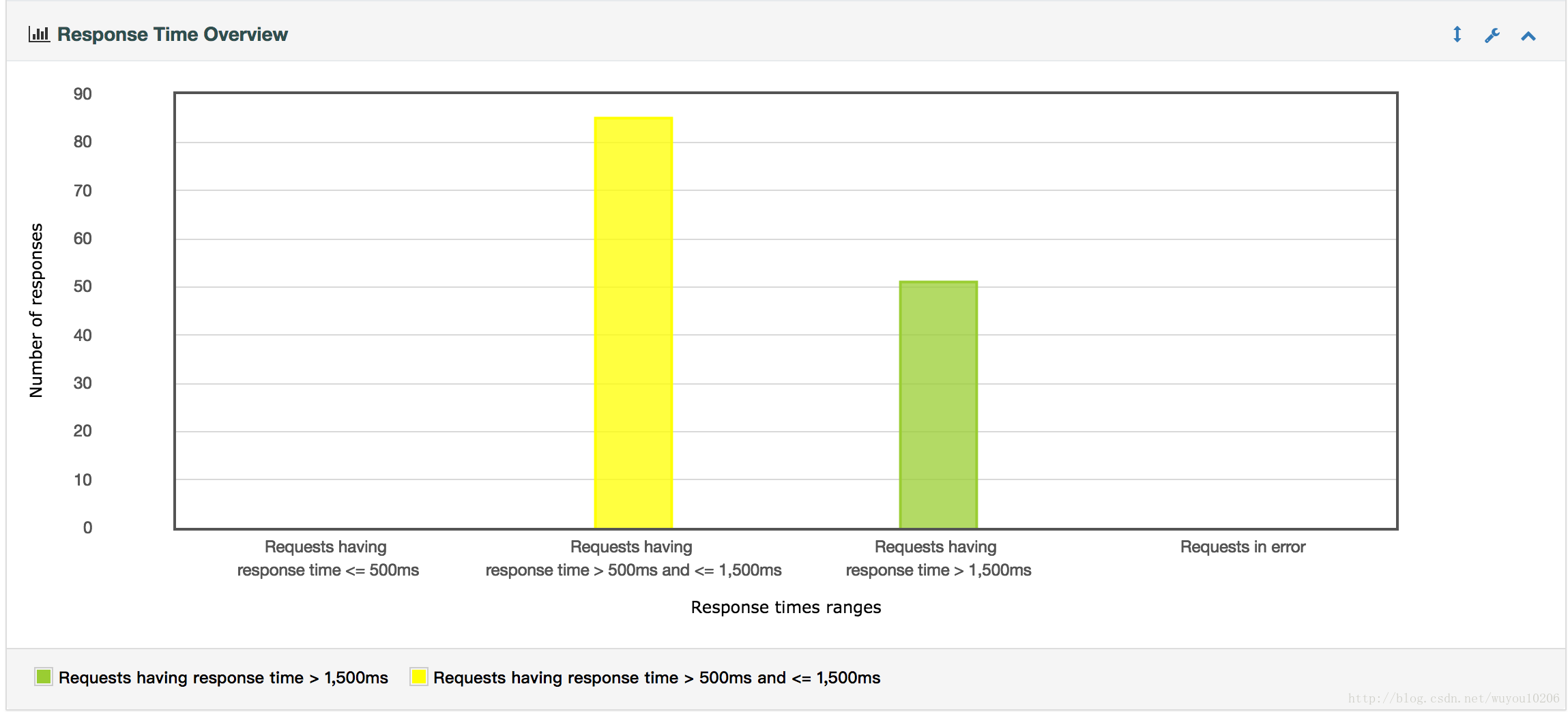
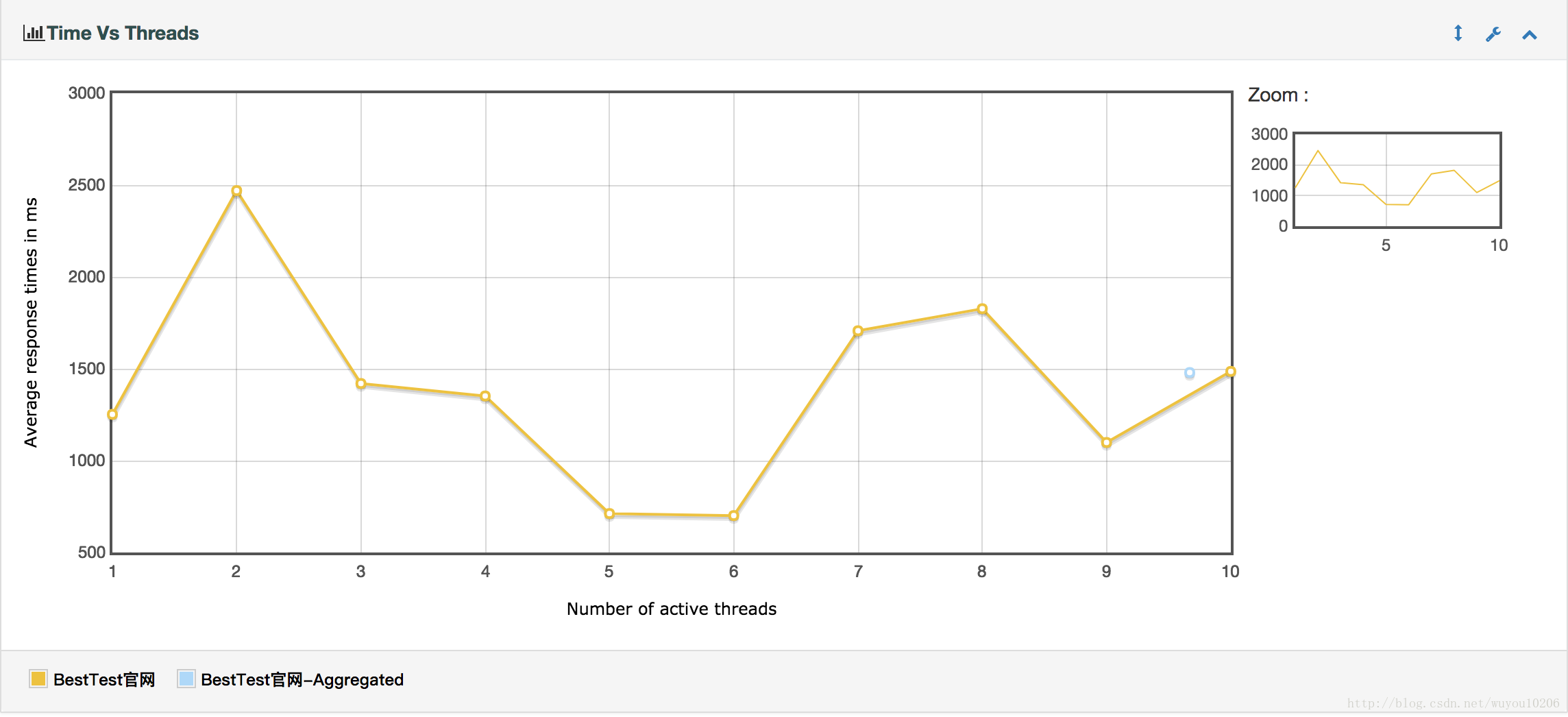
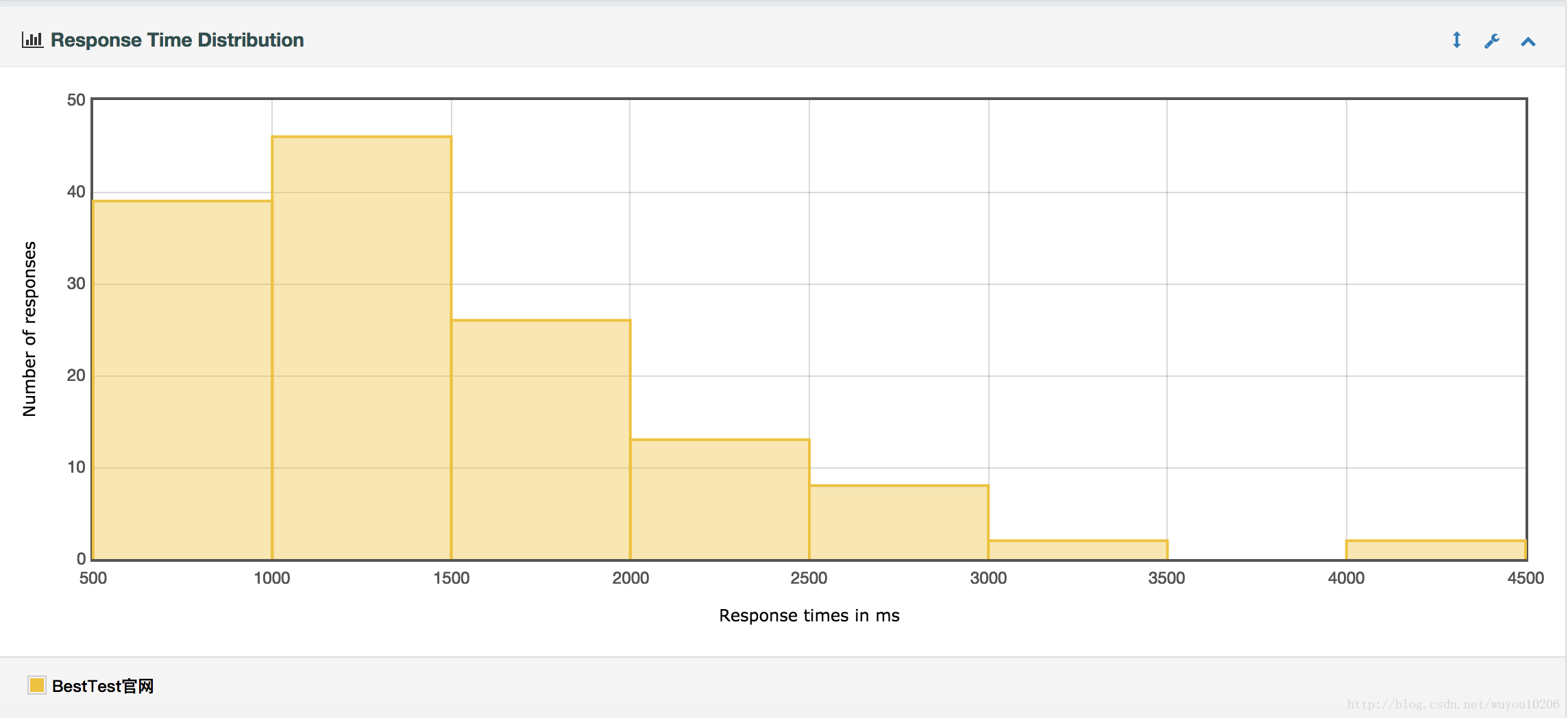


**Over Time**   
● Response Times Over Time脚本运行时间内响应时间分布曲线   
  
● Response Time Percentiles Over Time (successful responses) 脚本运行时间内成功响应的请求，响应时间百分位   
  
● Active Threads Over Time 脚本运行时间内的活动线程分布   
  
● Bytes Throughput Over Time脚本运行时间内的吞吐量，单位是byte   
  
● Latencies Over Time脚本运行时间内毫秒级的响应延时   
  
● Connect Time Over Time脚本运行时间内平均连接时间   


**Throughput**

● Hits Per Second (excluding embedded resources) 每秒点击数曲线   
  
● Codes Per Second (excluding embedded resources)每秒状态码分布曲线   
  
● Transactions Per Second 每秒事物数曲线   
  
● Response Time Vs Request 响应时间中值与每秒请求数关系曲线   
  
● Latency Vs Request 延迟时间中值与每秒请求数关系曲线   


**Response Times**

● Response Time Percentiles毫秒级百分位响应时间曲线   
  
● Response Time Overview 响应时间概述柱状图   
  
● Time Vs Threads 活动线程与平均响应时间变化曲线   
  
● Response Time Distribution响应时间分布图   


怎么样，这样一个测试报告是不是很炫，自己试试吧。