Some points about my solution:

1. Performing get request with one request per millisecond frequency, is useless because the response time is totally depends on network quality. For example, on my pc, I got the response with about every 10 milliseconds delay. So even if we do such request, we probably couldn't get a clock with one millisecond precision. Nevertheless, I was initially used a Timer with 1 milliseconds frequency and perform request on each period. The responses has been received about every 10 ms and after some time, the server couldn't response the requests at all. It makes sense because such usage could occupy all server process strength and make it down for other users. So after all, I decided to perform request each time I got the response of my previous request. It may be not the best solution for clock. Because for example, using GPS time could be much much better clock in precision (about 100 ns); But my solution works on my iPhone with about 20 milliseconds precision, approx. Of course it could be awful if the network is unreachable. So, there isn't any mathematical proof about the precision of this clock.

Below, you can find some discussions about uncertainty in accuracy of NTPs, which is similar concept to this test project:

https://serverfault.com/questions/508586/is-there-research-material-on-ntp-accuracy-available

2. You can see primary results of every millisecond request below. The response delay is more than 9 ms in these examples, due to network quality:

```
Optional("{\"datetime\": \"2019-02-10
    16:22:27.355462\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.375043\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.364133\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.408384\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.418206\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.451424\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.483900\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.513573\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.527280\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.462046\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.527667\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.472298\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.469874\"}")
Optional("{\"datetime\": \"2019-02-10
    16:22:27.483067\"}")
```

Probably, server doesn't respond me after a while which I was requesting every 1 millisecond, due to server policies:

```
2019-02-10 19:53:32.988211+0330
    TakeHomeTest[1316:275176] Task
    <C5E7F18A-
    F6D6-405A-912E-35F8BE857EE6>.<3441>
    finished with error - code: -1001
2019-02-10 19:53:32.996646+0330
    TakeHomeTest[1316:275176] Task
    <8BC0B261-8611-497D-AF56-
    DAD5D08F8F77>.<3442> finished with
    error - code: -1001
2019-02-10 19:53:32.997182+0330
    TakeHomeTest[1316:275176] Task
    <81922FA9-2FBF-486D-910E-413004B51F5
    5>.<3443> finished with error -
    code: -1001
2019-02-10 19:53:32.997743+0330
    TakeHomeTest[1316:275176] Task
    <B174B8BB-8687-470F-B500-
    CF3023711BBE>.<3444> finished with
    error - code: -1001
```

You can find my final solution at:

https://github.com/abbassabeti/Draggable-Rotating-Clock

Regards,

Abbas Sabeti

+98 912 248 23 49

abbassabetinejad@yahoo.com

abbassabetinejad@gmail.com