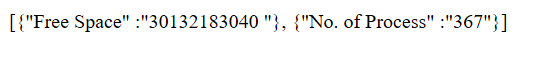
**WebApp**

**API used :**

**GET :** <http://localhost:8080/WebApp/job>

Use to retrieve the current system metrics (Free Memory Space / No of running process)

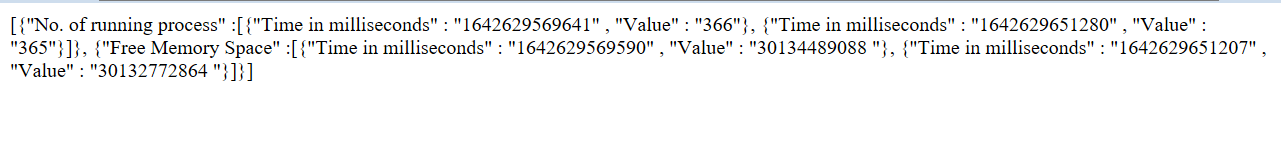
**Result :**



**GET :** [http://localhost:8080/WebApp/History?time1=<time1](http://localhost:8080/WebApp/History?time1=%3ctime1)>

Takes time in milliseconds (time1) as input and provides the system metrics for the provided time

**Result :**



**Prerequisite :**

1. Create a text file : ‘Running\_Process.txt’ with header as ‘TimeStamp|Value’
2. Create a text file : ‘Free\_Memory.txt’ with header as ‘TimeStamp|Value’

**Details :**

GetCurrentData.java : Provides the current system data (No of process running and free memory metrics ).This is implemented using Callable module in Java . This class will make a call to command prompt and retrieve the details

InitializeLogging.java : Starts logging of the metrics in an external file at a frequent interval of time once the server is up and running.

The job to collect the no. of running process runs every 2 mins and logs the timestamp and the respective value whereas Free Space memory calculator job runs every 3mins. It executes the command with the help of [Runtime class](https://www.geeksforgeeks.org/java-lang-runtime-class-in-java/).

GetHistoryData.java : It takes time in millisecond as input and generates the metrices from that time.

Example if time provided is 2 min, then it will return the data from the last 2 mins. We have used [Future](https://docs.oracle.com/javase/10/docs/api/java/util/concurrent/Future.html) interface methods to obtain the result generated by a Callable object and manage its state