Backend Coding Challenge

Table of Contents

Change Log

Version	Date	Author	Comment
0.9	10.2016	KC	Initial Version
1.0	12.2017	KC	Update

Coding Task

A mongoDB collection contains documents with the following structure:

```
{
  timestamp: ISODate('2016-02-02'),
  partitions: [
   { key: 5, val: 20 },
   { key: 10, val: 15 },
   { key: 15, val: 55 },
   { key: 35, val: 1 },
   { key: 95, val: 22 },
  ]
}
```

Please create a function, that will, using $aggregate framework from mongodb, calculate the sum of all values for keys (below + equal) and above the threshold in selected time range.
Function should look like:

```
  function sum(startDate, endDate, threshold, callback)
```

where startDate and endDate are the date strings, threshold is a number, callback is called with 2 parameters (error, result);

For example when we have only one document in the collection we will get following result:

```
sum('2016-02-01', '2016-02-03', 10, (err, result) => {
  should.equal(result, { key: 10,  below: 35, above: 78});
});
```

Please create several tests using mocha (https://mochajs.org/) test framework to make sure that everything is working correctly.

Extended Coding Task

This is an optional part, which is compulsory...

Please modify the function above to get an array of thresholds. For example if the values [a, b, c] are passed, then aggregate the values for ranges [0->a],[a->(b-1)], [b->(c-1)], [c->999], where 999 is a hard-coded cap.
For example:

```
sum('2016-02-01', '2016-02-03', [10, 35], (err, result) => {
  should.equal(result, [
    { key: 10,  below: 35},
    { key: 35,  below: 56},
    { key: 999, below: 22}
  ]);
});
```

In case of **any** questions, please, feel free to contact me by this email address: kirill.churikov@thinxnet.com.