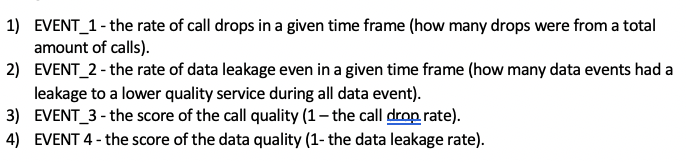
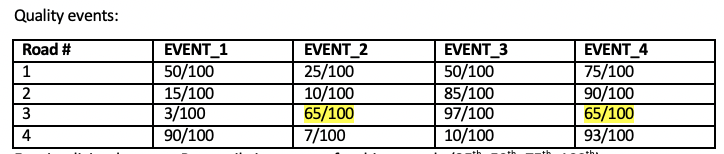
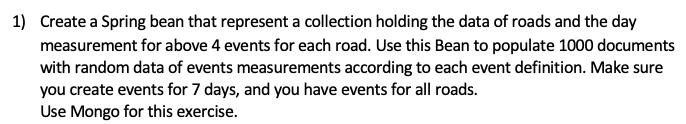
This document is an explanation how I understand the task and results of my code.

1. Let’s assume the events on road may be shown as a chart. And each road (road element?) has a chart per day.

So, each chart can be converted into 4 events measurements   
  
As I understand “in given time frame” – is per day. So, entries may look like:

1. Event 1 – cd / cn
2. Event 2 – 1 – Event 1 value
3. Event 3 – dl / du
4. Event 4 – 1 – Event 3 value

The data from example table is corelate with my point of view, except cells marked yellow (I decided it’s just a typo?)  


1. Then I was trying to understand the task 1 requirement:  
   

So, I decided the bean should generate data events measurements, not the events itself. I not clearly understand what exactly means to populate 1000 documents. From the point 1. I assumed that 1 road may have 1 events measurement per day. Does it mean we need to generate 1000 docs for each type for each day and in this case 1000 means number of roads. Or 1000 it’s a total count, and the number of roads is 1000 / 4 / 7? The result of second case is float number, so I decided to choose first one.

1. As I understand, the calibration is an aggregating the events measurements per week and divided the group of roads by percentile value. I will try to explain my point of view graphically:

So based on this chart, all roads with an “event\_n” value less or equals 0.8 (percentile from the task) has rank 1, 0.8 – 0.9 – ran 2, etc.

1. Ass the events measurements where random based, most of the roads have rank1. It may be changed by passing different percentile through the app.properties. Actually, all parameters are configurable, so you can play with them.
2. Results (was fetched by unit tests, so the request and response has a structure presented by unit tests print() method:

|  |
| --- |
| MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/765/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_1","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/463/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_1","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/875/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_2","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/474/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_1","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/625/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_1","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/828/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_1","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/810/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_1","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/239/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_1","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/888/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_1","EVENT\_2":"RANK\_1","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = []  MockHttpServletRequest:  HTTP Method = GET  Request URI = /road/481/ranks  Parameters = {}  Headers = []  Body = null  Session Attrs = {}  Handler:  Type = com.example.demo.controllers.RoadController  Method = com.example.demo.controllers.RoadController#getRanks(String)  Async:  Async started = false  Async result = null  Resolved Exception:  Type = null  ModelAndView:  View name = null  View = null  Model = null  FlashMap:  Attributes = null  MockHttpServletResponse:  Status = 200  Error message = null  Headers = [Content-Type:"application/json"]  Content type = application/json  Body = {"ranks":{"EVENT\_1":"RANK\_2","EVENT\_2":"RANK\_3","EVENT\_3":"RANK\_1","EVENT\_4":"RANK\_1"}}  Forwarded URL = null  Redirected URL = null  Cookies = [] |