COUNT SPECIFIC CHARACTER - SPRING BOOT MAVEN PROJECT

Sunday, May 14, 2017   10:55 AM

Tools/Technology Used:
JAVA (JDK 1.8)
ECLIPSE STS
GIT
MAVEN
MONGODB
RESTFUL
POSTMAN

Business Requirement
BASE CASE
Count the number of times a particular character occurs in a string.
Ex: ('Vivek', v)
Ouput : 2 (v occurs twice in Vivek)

Converted the CountSpecificCharacter-PlainJava project into Spring Boot Maven project.

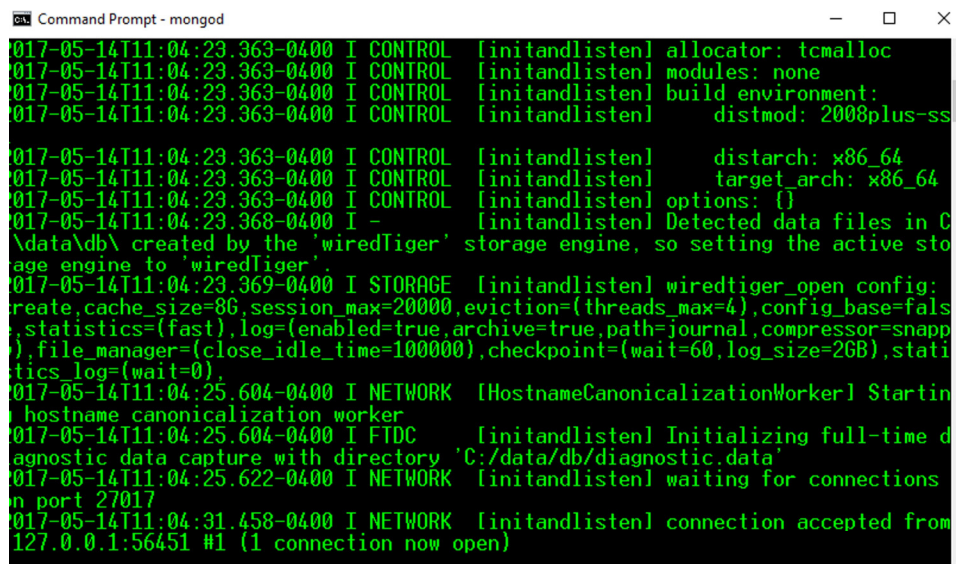Multiple inputs so far have been provided in a text file.
The outputs will also be provided in the text file upon running the service.

Upon running the project, the following URI can be used to run the service to get the result
Of all the outputs that are currently redirected to a output file.

- Currently no interaction is present with Mongodb. But, still mongodb dependency has been added for future implementation.
  To run this project, mongodb port needs to be open on your local.

URI PATH: http://localhost:<server>//count/char/

Sample Output Response (Obtained Through POSTMAN):
[
 {

```
    "1": [
      "drum",
      "d",
      "1"
    ]
  },
  {
    "2": [
      "van",
      "v",
      "1"
    ]
  },
  {
    "3": [
      "sucessful",
      "s",
      "3"
    ]
  },
  {
    "4": [
      "triumph\"",
      "m",
      "1"
    ]
  },
  {
    "5": [
      "queenelizabeth",
      "q",
      "1"
    ]
  },
  {
    "6": [
      "poopooguys",
      "o",
      "4"
    ]
  },
  {
    "7": [
      "samosa",
      "z",
      "0"
    ]
  }
  }
]
```

Next Step Implementations

☐ Integration with MongoDb

- [ ] Perform input validations, Exceptional handling
- [ ] Add additional scenarios in REST
- [ ] Add additional business cases itself
  - Support for Reading/Writing XML
  - Support for Reading/Writing EXCEL
- [ ] JAVA 8 CONCEPTS (Streams, Lambda)
- [ ] Using polymorphism , abstract, overriding concepts
- [ ] Unit testing (Junit + Mockito +Coverage)
- [ ] Add security layer
- [ ] Possible Integration testing
- [ ] Integration to Angular Frontend
- [ ] Building Project using Jenkins
- [ ] Data structure concepts - Binary search, constant time
- [ ] Possible Threading scenario