Name: Vinayak Kalaskar

Answer 1:

Code Github Link: https://github.com/vinayak-sk/Nokia-Solutions

Answer 2:

Documentation in the code itself

Answer 3:

* Developing a REST API (for example, using Spring) for sending text file over HTTP using a POST request.
* The function that handles the request can read the files from the InputStream and either save it to a local file temporarily using OutputStream (for large files) or cache it in a buffer string by reading all bytes.
* The file or the string can be passed to my code to obtain the list of sorted unique words.
* This list can be sent back to the client a supported format (say, JSON using Jackson ObjectMapper)

Answer 4:

Instead of using a map to store counts of unique words, the words can be inserted into a common database that will be used by all clusters to store count of the words.

The table structure can be:

Occurrences -> Table name

Word –> Varchar

Count –> int

UNIQUE(Word)

The query for reading will be: “Select word, count from occurrences order by count desc, word;”

The insert query for updating/inserting count of words can be (for mysql) “Replace into occurrences (word, count) values(…,…);” This query will update the count for word if it is already present in the database or insert a new row with the word and the count. For this query to work, it is important to define a unique constraint on word column.

Answer 5:

Test Case:

Included on github

Answer 6:

Solution 1:

Server-side Rendering

* We can design the system to provide server-side rendering.
* We can build a web server which will make all the 5 ajax calls to the external REST API to generate data and then render the HTML page sending it to the client.
* Server-side rendering will help us to load the webpage with HTML, CSS, layout, JavaScript and DATA already embedded in HTML.
* Advantage will be that user will see all the data before.
* Disadvantage will be less efficient response time.

Solution 2:

Client-Side Rendering

* Hiding the initial HTML page using display:none for body.
* Developing a callback function which has the other 4 ajax calls inside it.
* On completion of these calls we can change the display to visible.
* Advantage: visually faster response time: user gets to see the partial web page (without the table content) in lesser time.
* Disadvantage: The data is loaded after a delay.