

Exercise Sheet 7

complete until Tuesday, December 8th

This exercise sheet is about implementing a web application, based on JavaScript, similar to the one we did in the lecture. The following exercises break up this task into subtasks.

Exercise 1

Turn your code from Exercise 1 from the last sheet into a server. The server should speak HTTP and upon a GET request with some prefix, return a valid XML document with the following elements: the prefix you sent, all completions of the prefix, for each completion the size of the inverted list, the size of the longest inverted list of a prefix, the total size of the inverted lists. Come up with some reasonable format for your XML. Test your server with wget or curl or via your browser.

Exercise 2

Write the HTML for a search application with a search field, and an area for showing the result which your server from Exercise 1 will deliver in a nice table. The page should contain some explanation of what it does. The page should pass through a validator without errors or warnings. Make an effort to make the page look nice. Use a CSS stylesheet for fonts, colors, and spacing.

Exercise 3

Building on Exercise 2, add the JavaScript code (using jQuery) that, upon every keystroke in the search field, sends the contained prefix to your backend, and then shows the content of the result XML as a nice HTML table with nice borders and proper spacing. The columns of the table should be: the prefix completion, the size of the inverted list, the percentage of that size relative to the the total size of all the completions. Each row of the table corresponds to one completion. Don't forget to put a header for each column.

Exercise 4

Add an additional column to the table that contains a horizontal bar that illustrates the percentage from the (so far) last column. (The simplest way to make such a horizontal bar is by having a certain character, say a *, stand for a certain amount, say 5%, and the repeat it the appropriate number of times.) Then make the table sortable (using jQuery's table sorter), such that a click on any of the table headers will sort the rows of the table by the contents of that column (don't worry, jQuery will do most of the work for you here).