

Distributed Systems Project, Spring 2013

Assignment: Multitier architectures and the Web

In this assignment, you need to write a web-based calculator both in a server-based version and in a version where some of the functionality is migrated on the client side.

Tasks

1. Write a web-based calculator which uses a simple input form and performs all the calculation on the server side.
2. Migrate some of the functionality of the calculator to the client side using Javascript.

Specifications

The server-based application should follow these specifications. The client-side implementation has three steps, which are defined below. Note that the maximum grade you can obtain depends on how many steps you complete.

Write a simple web form which contains two text input fields, one selector for the operation (supported operations +, -, *, and /), and one button for submitting the calculation. The form must use the GET-method. The server will perform the operation and return a line with the calculation, for example "1 + 2 = 3", which the browser displays (along with a new input form). You need to keep the history of all calculations visible on the page. You are free to choose the mechanism for keeping the history.

NOTE: The names of the parameters MUST be arg1, arg2, and op, for the two arguments and the operator, respectively.

Step 1

You need to add functionality on the client side with Javascript as follows. Replace the input form with a single text input field and the submit button. The client-side script should parse the input given in the text field and convert it to the "atomic" operations supported by the server (+, -, *, and /). Precedence of operators goes from left to right, i.e., it is not the usual order of precedence.

Example: Suppose the input form has the text "1 + 2 * 3 / 4". Your script should break this up to 3 operations which are submitted individually to the server. Given the way the server works, all the intermediate results should also be shown on the browser. In this case the output looks like this:

$$1 + 2 = 3$$

$$3 * 3 = 9$$

$$9 / 4 = 2.25$$

Step 2

Extend the script from step 1 to handle parenthesis. Parenthesis change the order of precedence as usual, but the overall order of precedence remains left-to-right. Allow for nested parenthesis up to an unlimited level of nesting.

Step 3

Add functionality to plot sine and cosine functions. The functions are given in the form of “sin(x)”, where x varies in the interval $[-\pi, \pi]$. You need to implement three variants of plotting. For plotting, you use step size 0.1 (or smaller) on the x-axis.

In the first variant, you send the whole line to the server (you need to write a new server-side script to handle this), the server parses the line, plots it in a figure, and sends back an image file (e.g., PNG-format) which the browser displays. You can use for example gnuplot to generate the figure.

In the second variant, plot the figure locally on the client. For implementing the plotting, you are allowed to use the standard JQuery-library (see below for URL). You do not need to contact the server at all.

In the third variant, the server does the calculations and the client does the plotting. On the server side, you are allowed to use only the basic operations +, -, *, and /. The client must send the appropriate calculations to the server to get the approximately correct values to plot. Keep the maximum approximation error to 1% of the true value. You can/should modify the server from step 1 slightly to be more useful. There is no need to keep the history of operations visible on the client side. For implementing the plotting, you are allowed to use the JQuery-library (see below).

Grading

Completing all three steps is mandatory to pass the assignment. Weak implementations can and will receive lower grades even if they implement all steps.

Grading is based on correctly implemented functionality, documentation, and coding style.

Guidelines

The assignment is individual work. Every student must return their own implementation. You can of course discuss any problems you encounter with other students, but sharing code is not allowed and if found, will be considered as plagiarism.

Use users.cs.helsinki.fi for testing your scripts. Instructions can be found at http://www.cs.helsinki.fi/u/lubbers/teaching/material/university/server_cs_users.shtml. **If you have questions about the setup, please contact Liang and not the person who wrote that page.** Alternatively, you can install your own web server on your own machine, but this is not recommended. Running own web servers on the normal department machines is NOT allowed.

No external libraries are allowed, unless otherwise specified in the description of the task. The JQuery library that you are allowed to use is <http://code.jquery.com/jquery-1.8.3.min.js> (later or earlier versions of this library are allowed). Note that use of for example, the flot library (<http://code.google.com/p/flot/>) or other libraries is NOT allowed.

You can freely select the language in which you implement the code on the server side.

Deliverables

Program source code with documentation and a document describing your implementation.

Timeline

The assignment is due on February 22nd at 20:00. No extensions will be given.

Return

Return your code and documentation by email to Liang.Wang@cs.helsinki.fi as one tar-archive. Please indicate clearly your name and student ID in every source code file.