Documentation

Webserver

The webserver's code is structured as follows:

1. Class variables
2. The constructor
3. Update methods:
    a. Port update
    b. Root directory update
    c. Maintenance directory update
    d. Status update
4. File path - computing method
5. HTTP Response - sending method
6. Handling client method
7. Non-persistent connection establishing method
8. Staring and running the webserver

GUI

The GUI's code is structured as follows:

1. Class variables +
2. Pannels, Borders, and Texts
3. Start-Stop button
4. Maintenance checkbox
5. Port value
6. Root directory value
7. Maintenance directory value
8. Field editability method
9. Starting and running the GUI and webserver together

Webserver

Class variables

1. HandledRequest:
    - A variable that is used to catch the sequence of paths
    - Needed for testing purposes
2. Webserver_port:
    - A variable holding the current webserver port value
3. Webserver_root_directory:
    - A variable holding the current webserver root directory path
4. Webserver_maintenance_directory:
    - A variable holding the current webserver maintenance directory path
5. Webserver_status:
    - A variable holding the current status of the webserver
    - Possible values:
        i. Stopped
        ii. Running
        iii. Maintenance

The constructor

Initializes the relevant class variables:

1. Webserver_port
2. Webserver_root_directory
3. Webserver_maintenance_directory
4. Webserver_status

Since the code of the webserver is run by the developer, the developer will give correct initial values for the above-mentioned class variables.
The correctness of the class variables refers to:

1. Ideally, an unestablished port value - not enforced - between 0 and 65535
2. A root directory that contains:
    a. Index directory
    b. Index.html file inside the index directory
    c. Maintenance directory
    d. Maintenance.html inside the maintenance directory
    e. Notfound directory
    f. Notfound.html inside the Notfound directory
    g. Serverdown directory
    h. Serverdown.html inside the Serverdowndirectory
    i. Favicon.ico file

3. A maintenance directory that contains:
    a. Maintenance.html file inside the maintenance directory given
4. Webserver status:
    a. Either Stopped, Running, or Maintenance
    b. Since this is set by the developer automatically, correctness is assumed, thus no explicit verification is made.

Update methods

Port update:

- The port value is updated with the new value
- Ideally, an unestablished port is given, but it is not enforced. The error will be caught accordingly in other methods.
- Accepted values for an update, are values between 0 and 65535

Root directory update:

- Must contain:
    - Index directory
    - Index.html file inside the index directory
    - Maintenance directory
    - Maintenance.html inside the maintenance directory
    - Notfound directory
    - Notfound.html inside the Notfound directory
    - Serverdown directory
    - Serverdown.html inside the Serverdowndirectory
    - Favicon.ico file

Maintenance directory update:

- Must contain:
    - Maintenance.html file inside the maintenance directory given

Status update:

- Either Stopped, Running, or Maintenance
- Since this is set by the developer automatically, correctness is assumed, thus no explicit verification is made.

File path - computing method

Returns the absolute path equivalent for the requested URL. It converts requests to absolute paths for requests made right upon entering the webpage - that have a shorter, incomplete path -, as well as for subsequent requests - that already have the complete path-.

The Path.get method replaces the "/" in a text with "\\".

HTTP Response - sending method

Since we deal with a webserver, the requests and responses are in HTTP format. Concretely and HTTP response looks like this:

1. Header containing:
    a. Type: HTTP
    b. Version: 1.1
    c. Status: 200 OK for a response to a correct request
    d. Type fo content: html, css
2. Payload containing:
    a. The actual webpage content

Handling client method

Here, the actual exchange of request and response takes place. Streams of bidirectional communication are opened, the request is saved as soon as it arrives, the raw path request is extracted, the complete path is computed and saved for testing purposes, and then an answer is given according to the webserver's status and file content stored.

As noted, the responses are sent based on content type as well. This is done so that in special cases - Maintenance and Stopped webserver - the first answer (HTML) is one that addresses the error, but the rest (CSS, favicon) have to be sent correctly without an error warning.

Possible errors are:

1. A null client object was given
2. The client suddenly disconnects and the webserver wants to handle their request

Non-persistent connection establishing method

Here, the client and webserver socket are created and opened, a single request is handled, after which both sockets are closed. Thus, we have a non-persistent HTTP connection.

Possible errors are:

1. The port onto which the webserver wants to listen is already established
2. The client tries to connect to a port at the same time another one is connecting/handling their request on the same port.

Staring and running the webserver

This happens in the main method. There, the webserver is started on an already free port, with a correct root and maintenance directory, having the status as running. After that, the webserver is listening uninterruptedly in its port for new client connections.

GUI

Class variables

1. The webserver variable: Will be bound to the instance of the to-be-created webserver
2. The port variable: The port on which the webserver will start listening
3. The index variable: The root directory that meets the requirements
4. The maintenance variable: The maintenance directory that meets the requirements
5. The status variable: The webserver's status that will be used for the frame

In addition, the following declarations are made:

1. The (main) frame: That will hold all the panels, the button, and the checkbox
2. The Start-Stop button: Initially on turned off ("Start server" is shown)
3. The maintenance checkbox: Initially unchecked and disabled - see line 140
4. The (to-be-inserted) port: Initially on 8080 - a usually free port
5. The (to-be-inserted) root directory: Initially a correct directory
6. The (to-be-inserted) maintenance directory: Initially a correct directory

Pannels, Borders, and Texts

Three panels are created:

1. Web Server information
2. Web Server control
3. Web Server configuration

Their borders are set as lowered ETCHED - see 34 - 41.
Their layout is established - 43 - 45.
The text fields/buttons are filled in, set to be editable or noneditable, their borders are/not suppressed, colors (where needed) were set, and aligned correctly.
The three panels are filled with the corresponding text fields - maintaining the layouts.

Start-Stop button

The checkbox is unticked, and an ActionListener is attached.
If the webserver was previously stopped, its status is set on Running, the editability of fields is set accordingly, the server status, server address, and listening port are updated in the Web Server information panel.
Else, its status is set on Stopped, and the abovementioned fields are updated accordingly

Maintenance checkbox

If the checkbox has been ticked, the webserver goes into Maintenance mode, the editability of fields is set accordingly, the server status, server address, and listening port are updated in the Web Server information panel.
Else, its status is set on Running, and the abovementioned fields are updated accordingly.

Port value

After the new port value is entered and the Enter button is pressed, the value is extracted and the (webserver) port variable and webserver internal port variable are updated.

Root directory value

After the new root directory value is entered and the Enter button is pressed, the value is extracted and it is checked whether it meets the requirements. If yes, then it is displayed that it has met the requirements by a green X and the root directory is updated, if not, a red O is shown.

Maintenance directory value

After the new maintenance directory value is entered and the Enter button is pressed, the value is extracted and it is checked whether it meets the requirements. If yes, then it is displayed that it has met the requirements by a green X and the maintenance directory is updated, if not, a red O is shown.

Field editability method

Here, the webserver status is updated, the frame title is set, and the editability of the entry fields for the webserver port, root and maintenance directory, and of the checkbox are set.

SetEnabled = The ability to give user input can be restricted temporarily
SetEditabible= Cannot or Can enter text

Starting and running the GUI and webserver together

The webserver variable is bound to an instance, the status is set to Stopped, the port, root and maintenance directory are set, the user interface is opened, and the webserver continuously listens to new clients' requests.