

# J2SE HTML Editor

---

## Definitions

ProgNett AS is considered as the buyer in this contract. The developer is considered as the seller in this contract.

## Description

We have an HTML editor for Java, but it creates very bad html tags (wrong tags and too many tags) and is hard to customize. It is therefore decided to create a new HTML Editor from scratch.

The basic functionality of the editor is very simple. A user with none knowledge of html should be able to edit text, insert pictures/links and other common text editing operations. The editor will transform the user input into the correct html tags.

For more advanced users, it should be possible to edit the html source directly. It is not expected nor required that an html editor can fix all problems, and therefore it must be possible to edit the html source directly in those special cases.

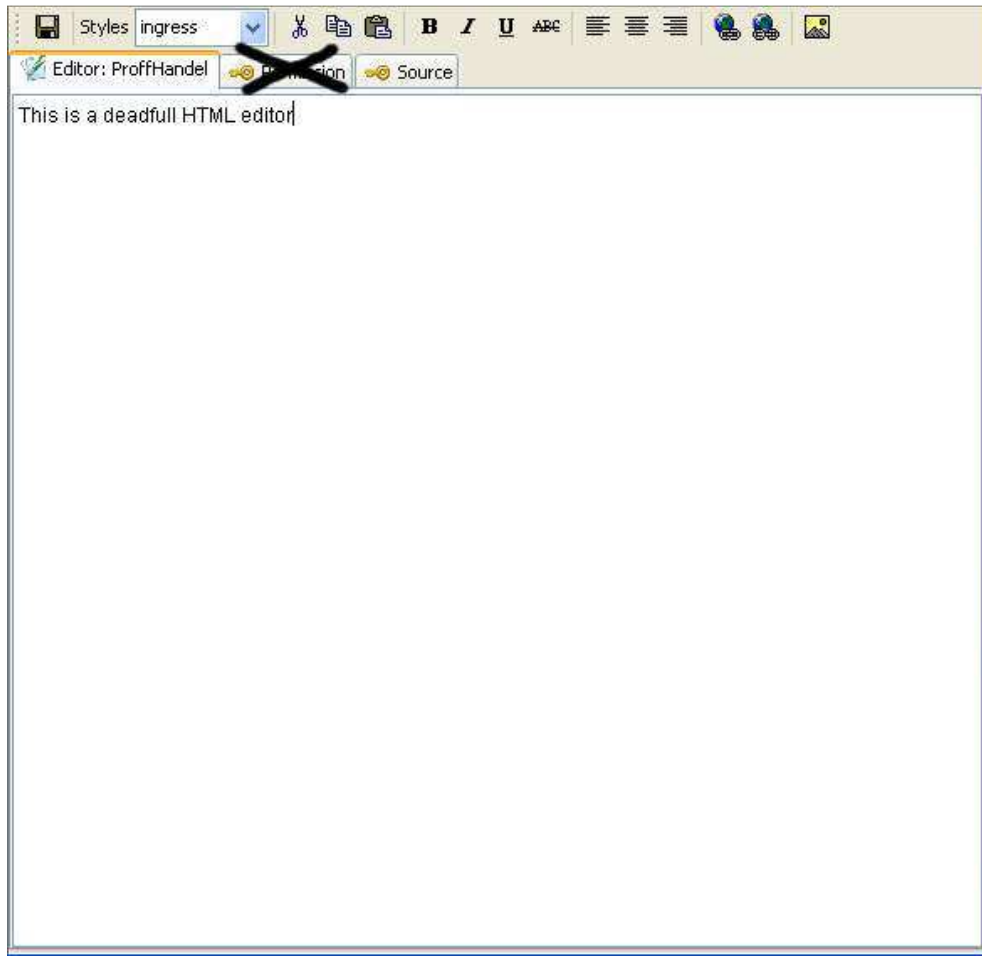
## Technical overview

The project must be developed in J2SE only. No native libraries or extensions may be used, since the editor must be able to operate on different platforms.

Third party libraries can't be used, unless they are licensed under the LGPL license. It would be a great plus if none third party libraries are used.

The editor itself must be broken into two components, the html editor and the source editor. They should be implemented as two JPanels. See picture 1 for an example UI.

Graphical elements in the project are something that will be replaced, and homemade buttons etc can therefore be used.



Picture 1. Example of already implemented HTML editor.

The implemented UI could be like this, but it is not a requirement.

## Requirements

### The requirements for the html editor panel are:

- Must support the following actions (each action can be a button on a toolbar)
  - Insert hypertext link
    - A simple `JOptionPane.showInputDialog(..)` can be used for input
  - Edit hypertext link
    - The already entered value should be changeable
  - Remove hypertext link
    - Removes the hyperlinke, but not the text
    - The while text must not be selected in order to apply the remove action. It should be enough to only have the mouse cursor somewhere inside the text
  - Insert picture

- A simple JOptionPane.showInputDialog(..) can be used for input
- Edit picture
  - The already entered value should be changeable
- Remove picture
  - Remove complete picture
- Insert script
  - A simple JTextArea can be used for input
  - All text entered should be placed between two script tags
  - The editor must display the script as a simple icon
- Edit script
  - The already entered value should be changeable
- Remove script
  - Remove complete script text
- Insert table
  - A JDialog must be used for input
  - The JDialog must contains fields like
    - Specify number of rows
    - Specify number of cols
    - Specify table width
    - Specify table border
  - Tables should be view with a border in the editor, even though the user wants border=0
- Edit table
  - The already entered value should be changeable
- Remove table
  - Complete table with text is removed
- Insert ordered list
  - Starts a new ordered list
  - If text is selected, it should be the first element in the list
  - If none text is selected, a empty list item is presented
  - Enter should insert a new list item
  - Enter pressed in an empty list item should close the list and move cursor after the list end tag (</li>)
- Remove ordered list
  - If all list elements are selected, the whole list is removed
  - If only some list elements are selected, only those are removed
- Apply bold format to selected text
- Remove bold format to selected text
- Apply underline format to selected text
- Remove underline format to selected text
- Apply strikethrough format to selected text

- Remove strikethrough format to selected text
- Apply italic format to selected text
- Remove italic format to selected text
- Append a CSS style to selected text
  - The css styles should be selected from a combobox
  - The values in the combobox must be loaded from a valid css file
  - Only div and span styles should be loaded from the css file. All other styles should be ignored.
  - Div and span should be removed from the css style name when presented in the combobox
    - If a style is like this “div.myStyle”, it should be presented like myStyle in the combobox
- If a text is formatted with a span style, the operations bold, italic, underline and strikethrough must be disabled.
- Append left align text to selected text
- Append center align text to selected text
- Append right align text to selected text
- Indent selected text
  - This should be controlled by applying a css div tag to the selected text
- Save to file
  - The html output should be saved to a file

### Must support the following mouse events in the html editor panel

- Right click any link / picture / script and select “remove” or “edit properties”
  - Remove has the same operation as the Remove link / picture / script operations specified above
  - Edit properties has the same operation as Edit link / picture / script operations specified above
- Right click on a table’s border (only border) and select “remove” or “edit properties”
  - Remove has the same operation as remove table specified above
  - Edit properties has the same operation as edit table specified above
- It should be possible to right click on any picture and select properties... from properties, you can change all image properties and save... the properties should then be updated.

### Requirements for the html source panel

- It should be possible to edit the source code directly
- Tags in source view should be highlighted

- It should be easy to change the highlighting color.

## General requirements

- The html output should be valid html at all times. This means well formed tags (ending and closing tags).
- Special characters (like &) should be converted using to unicode (£ should be converted to &#163;)
- Text should not have any css style or html tag applied default
- A simple enter should break the current tag working in
  - insert closing tag and then two br tags
- A shift+enter should insert a single br tag, but not break the current tag working with.
- Html tags which is not understood by the editor or is information ignored by the editor should not displayed in the editor.
- The editor must assume that all text entered should be placed between the body tags of an html page.
- It should be possible to paste text from the clipboard directly into the editor.
  - All tags and styles should be removed from the input text.
  - Input from ie. Word should be parsed as good as possible.
    - Remove obvious html tags
    - Remove obvious css styles
    - Only keep normal text
- Comments should be placed where needed
- It should be easy to add new commands.
  - It should be easy to add a new command like “insert JavaScript”.
  - Easy means that it can be documented how to add a new generic command.

## Project requirements

- No class files over 500 lines
- The project must be an eclipse or netbeans project
- Interfaces should be used as much as possible
- No third party library can be required for the editor to operate, except for open source projects with the LGPL license.

## Documentation requirements

- Documentation on how to extend new commands should be provided. Otherwise, code comments are just fine for documentation.

## **Acceptance of code**

The code is accepted when all the requirements above is fulfilled. Exceptions can be made if the seller and buyer agree on modifications and exceptions.

After the code is delivered, the buyer has one week to find bugs. Bugs found must be corrected of the seller within one week after the bugs were reported. The bugfix must not break requirements that worked in the first version.

## **Source code**

All the finished source code belongs to the buyer after the project is closed.

The source code and/or binaries (jar, class files) can't be used by the seller in any means without permission from the buyer.

The seller ensures that the source code is his own work or obtained from a LGPL licensed project.

## **End product**

The end product from the seller should be a zip/rar file with the complete eclipse/netbeans project. All requirements should be included, and the code must compile with errors. A general starter should be provided.