

MDMERGE

I developed `mdmerge` for my own usage. I was tired to update my main `README.md` file when it contains some codes (all or part of) from an existing file.

When the source file change, your `README.md` must be manually updated.

So I create my own **Node.js** application to update my `README.md` which using other files thanks to a custom instruction which is invisible in the rendered document (also in PDF document).

Currently, the application is not fully finished but the main feature are finished.

Getting Start

The Command Line Interface

`mdmerge` have not got any option. The only expected value is the file to process :

```
mdmerge README.md
```

The Instruction

The detailed instruction hereafter must be placed at the begin of the line in a comment block in your Markdown file : `[](<instruction>)`

Once parsed by `mdmerge`, a closing instruction has been inserted. That will allows the application to update the included parts.

Please find below, the generic form of the instruction `<instruction>` :

The element separator is the colon char (:)

- `<instruction>` = `@import><pathToFile>[...:<inclusionInstruction>][:<codeBlockLanguage>]`
 - `<pathToFile>` : This is the relative path to the file used for this inclusion.
 - `:<inclusionInstruction>` : You can insert none to many of the inclusion instruction separated by `:`.
- `<inclusionInstruction>` = `<line>[,<startOffset>][,<endOffset>]`
 - `<line>` : You can specified the line to toggle the inclusion.
 - `<startOffset>` : You can specified from which char you want to begin the inclusion on the line.
 - `<endOffset>` : You can specified when you stop the inclusion on the line
- `:<codeBlockLanguage>` : You can ask to `mdmerge` to insert your content in a Markdown code block in a specified language. You can omit `<codeBlockLanguage>` to let your favorite render to identify automatically the language (Example with **Highlight.js**)

The `<inclusionInstruction>` toggle the inclusion.

The generic rule is the **odds** elements will start the inclusion whereas the **evens** will stop the inclusion. When the total number of `<inclusionInstruction>` is **odd**, the last inclusion process will inclusion until the **EOF** is met.

Please find an example of instruction :

```
[] (@import>bin/mdmerge.js:53:79:js)
```

The instruction can be read as "Please, include here the JavaScript code from file `mdmerge.js`, from line `53` to line `79` in a code block `js`".

The result is the following

```

/**
 * Afficher un message dans le stream.
 *
 * @param message Message à afficher
 * @param level Niveau de message. 0=OK,1=KO,2=WARN
 */
function log(message, level = 0, args = []){
    // 0 = Success
    // 1 = Error
    // 2 = Warn
    let levels = [
        {color: "Green", name: "SUCCESS", return: 0},
        {color: "Red", name: "ERROR", return: 1},
        {color: "Yellow", name: "WARNING", return: 0},
    ];

    console.log(
        "[ " +
        colors.fg[levels[level].color] +
        levels[level].name +
        colors.Reset +
        " ] : " +
        message
    );

    return levels[level].return;
}

```

Usage Cases

Please find below an exhaustiv list of usage cases and their development status

Instruction	Explanation	Dev. Status
@import>test/incl.md	Include the whole content of file <code>incl.md</code> in folder <code>test</code>	DONE
@import>test/incl.md:	Include the whole content of file <code>incl.md</code> in folder <code>test</code> in a code block	DONE
@import>test/incl.md:js	Include the whole content of file <code>incl.md</code> in folder <code>test</code> in a code block JavaScript <code>js</code>	DONE
@import>test/incl.md:53	Include the content of file <code>incl.md</code> from line <code>53</code> to the end of file <code>EOF</code>	DONE
@import>test/incl.md:53:	Include the content of file <code>incl.md</code> from line <code>53</code> to the end of file <code>EOF</code> in a code block	DONE
@import>test/incl.md:53:js	Include the content of file <code>incl.md</code> from line <code>53</code> to the end of file <code>EOF</code> in a code block JavaScript <code>js</code>	DONE
@import>test/incl.md::79	Include the content of file <code>incl.md</code> from the beginning to the line <code>79</code>	DONE
@import>test/incl.md::79:	Include the content of file <code>incl.md</code> from the beginning to the line <code>79</code> in a code block	DONE
@import>test/incl.md::79:js	Include the content of file <code>incl.md</code> from the beginning to the line <code>79</code> in a code block JavaScript <code>js</code>	DONE
@import>test/incl.md:53:79	Include the content of file <code>incl.md</code> from line <code>53</code> to the line <code>79</code>	DONE
@import>test/incl.md:53:79:	Include the content of file <code>incl.md</code> from line <code>53</code> to the line <code>79</code> in a code block	DONE
@import>test/incl.md:53:79:js	Include the content of file <code>incl.md</code> from line <code>53</code> to the line <code>79</code> in a code block JavaScript <code>js</code>	DONE