Software Development in Practice

Software Engineering 2018

Bruno Carvalho
up201606517

Diogo Yaguas
up201606165

Tiago Castro
up201606186

November 18, 2018

Contents

1 Issue 2554

When trying to change the zoom value by clicking the zoom button, the zoom label on the right lower corner updates correctly, but when trying to zoom in using the key shortcut (CTRL + '+'/'-') or via the view menu, the zoom label doesn't update.

1.1 Steps

1. Launch Boostnote
2. Click on the zoom button on the right lower corner
3. Change the zoom level
4. Verify that zoom label was successfully altered
5. Change the zoom level using the key shortcut (CTRL + '+'/'-') or the view menu
6. Verify that zoom label was unsuccessfully altered

Result: The key shortcut doesn't work

1.2 Requirements

The key shortcuts for *zoom in*, *zoom out* should modify the zoom label on the right lower corner, however, the key shortcuts just zoom in and zoom out, and don't alter it.

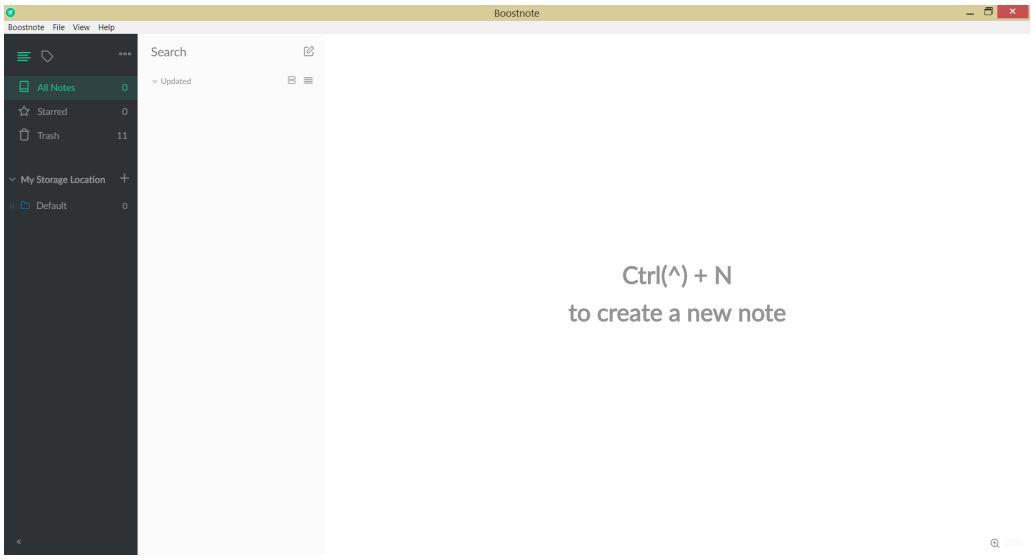


Figure 1: Normal Zoom

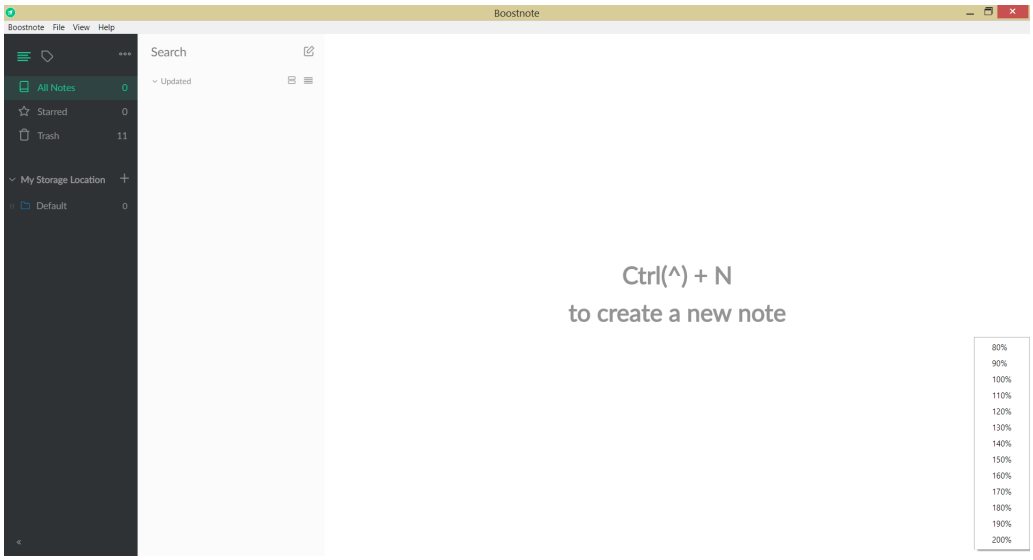


Figure 2: Zoom Menu

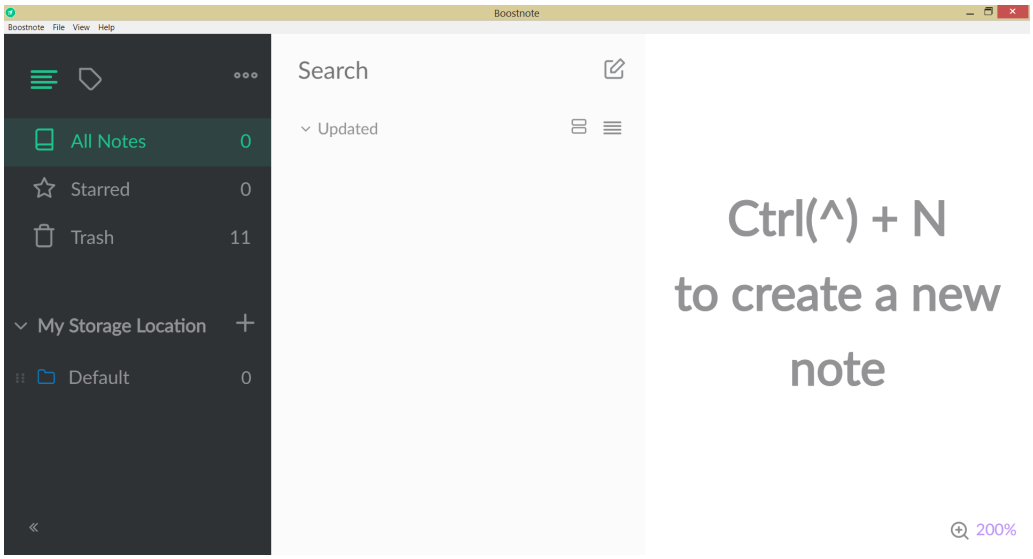


Figure 3: Zoom level at 200% and label update

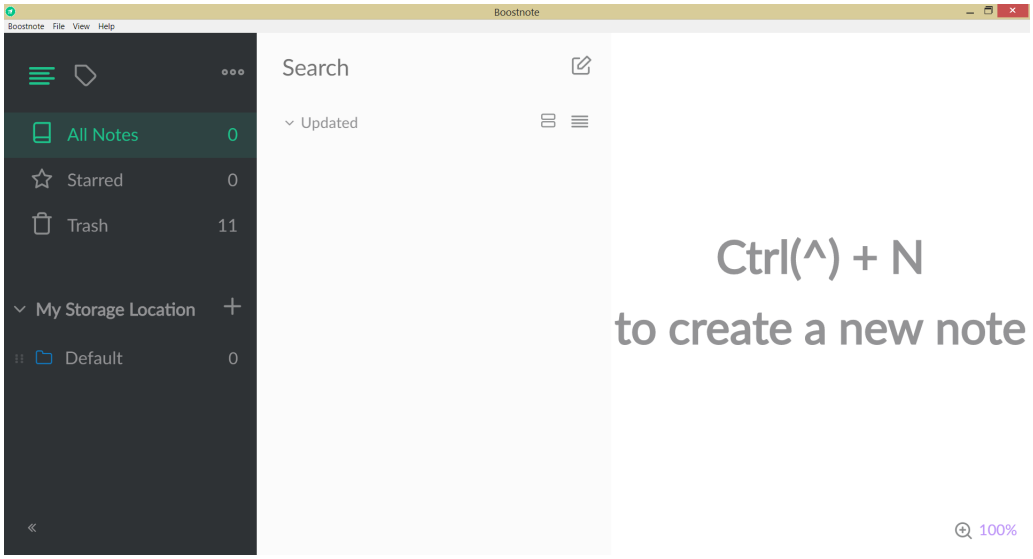


Figure 4: Zoom level increased with key shortcut and label not updated

In **main-menu.js**, the Menu template (Electron) is created and there, the 'Zoom In' and 'Zoom Out' options are added to the view menu.

The menu itens are roles (labels with pre-made listeners) for the zoom in and zoom out buttons as electron has specific listeners for these actions:

- **zoomIn** - Zoom in the focused page by 10%.
- **zoomOut** - Zoom out the focused page by 10%.

CTRL+ is associated to Zoom In and CTRL- is to Zoom Out by association as triggers to modify the zoom.

The problem is that, when you zoom in or out using the views menu, the browser window zoomfactor is altered, but the config isn't altered, only the browser window class. The browser window class is where the zoom is actually set and the config is where the zoom value is kept for easy access.

When the zoom is changed via the status bar zoom (on the right bottom corner), not only is the browser window zoomFactor value changed, but also the config zoom value from which the current zoom label gets its value. The function used to do this is the **handleZoomMenuItemClick** function in **StatusBar/index.js** which uses the **setZoom** and inheretly the **saveZoom** function from **ZoomManager.js**.

The label is defined in the render function of the StatusBar on StatusBar/index.js and as we can see, it get se zoom value from the config.

1.3 Source Code Files

```
312       {
313         role: 'zoomin',
314         accelerator: macOS ? 'CommandOrControl+Plus' : 'Control+='
315       },
316       {
317         role: 'zoomout'
318       }
```

Figure 5: main-menu.js

```
handleZoomMenuItemClick (zoomFactor) {
  const { dispatch } = this.props
  ZoomManager.setZoom(zoomFactor)
  dispatch({
    type: 'SET_ZOOM',
    zoom: zoomFactor
  })
}
```

```
render () {
  const { config, status } = this.context

  return (
    <div className='StatusBar'
      styleName='root'
    >
      <button styleName='zoom'
        onClick={(e) => this.handleZoomButtonClick(e)}
      >
        <img src='../resources/icon/icon-zoom.svg' />
        <span>{Math.floor(config.zoom * 100)}%</span>
      </button>

      {status.updateReady
        ? <button onClick={this.updateApp} styleName='update'>
          <i styleName='update-icon' className='fa fa-cloud-download' /> {i18n.__('Ready to Update!')}
        </button>
        : null
      }
    </div>
  )
}
```

Figure 6: StatusBar/index.js

```
function _saveZoom (zoomFactor) {
  ConfigManager.set({zoom: zoomFactor})
}

function setZoom (zoomFactor, noSave = false) {
  if (!noSave) _saveZoom(zoomFactor)
  remote.getCurrentWebContents().setZoomFactor(zoomFactor)
}
```

Figure 7: ZoomManager.js

```
export const DEFAULT_CONFIG = {
  zoom: 1,
  isSideNavFolded: false,
  listWidth: 280,
  navWidth: 200,
  sortBy: {
    default: 'UPDATED_AT' // 'CREATED_AT', 'UPDATED_AT', 'APLHABETICAL'
  },
  sortTagsBy: 'ALPHABETICAL', // 'ALPHABETICAL', 'COUNTER'
  listStyle: 'DEFAULT', // 'DEFAULT', 'SMALL'
  amaEnabled: true,
  hotkey: {
    toggleMain: OSX ? 'Command + Alt + L' : 'Super + Alt + E',
    toggleMode: OSX ? 'Command + Alt + M' : 'Ctrl + M',
    deleteNote: OSX ? 'Command + Shift + Backspace' : 'Ctrl + Shift + Backspace'
  },
  ui: {
    language: 'en',
    theme: 'default',
    showCopyNotification: true,
    disableDirectWrite: false,
    defaultNote: 'ALWAYS_ASK' // 'ALWAYS_ASK', 'SNIPPET_NOTE', 'MARKDOWN_NOTE'
  },
  editor: {
    theme: 'base16-light',
    keyMap: 'sublime',
    fontSize: '14',
    fontFamily: win ? 'Segoe UI' : 'Monaco, Consolas'
```

Figure 8: ConfigManager.js

```
const mainWindow = new BrowserWindow({
  x: windowSize.x,
  y: windowSize.y,
  width: windowSize.width,
  height: windowSize.height,
  useContentSize: true,
  minWidth: 500,
  minHeight: 320,
  autoHideMenuBar: showMenu,
  webPreferences: {
    zoomFactor: 1.0,
    enableBlinkFeatures: 'OverlayScrollbars'
  },
  icon: path.resolve(__dirname, '../resources/app.png')
})
```

Figure 9: main-window.js

1.4 System Architecture

To better understand the system and how the modules interact, a diagram was created in which the different dependencies in boostnote can be observed. This type of diagram was picked because it gives an overview of the system without going into too much detail as the project is far too big. The most important dependency to this issue is the electron dependency as it is the framework used for the application.

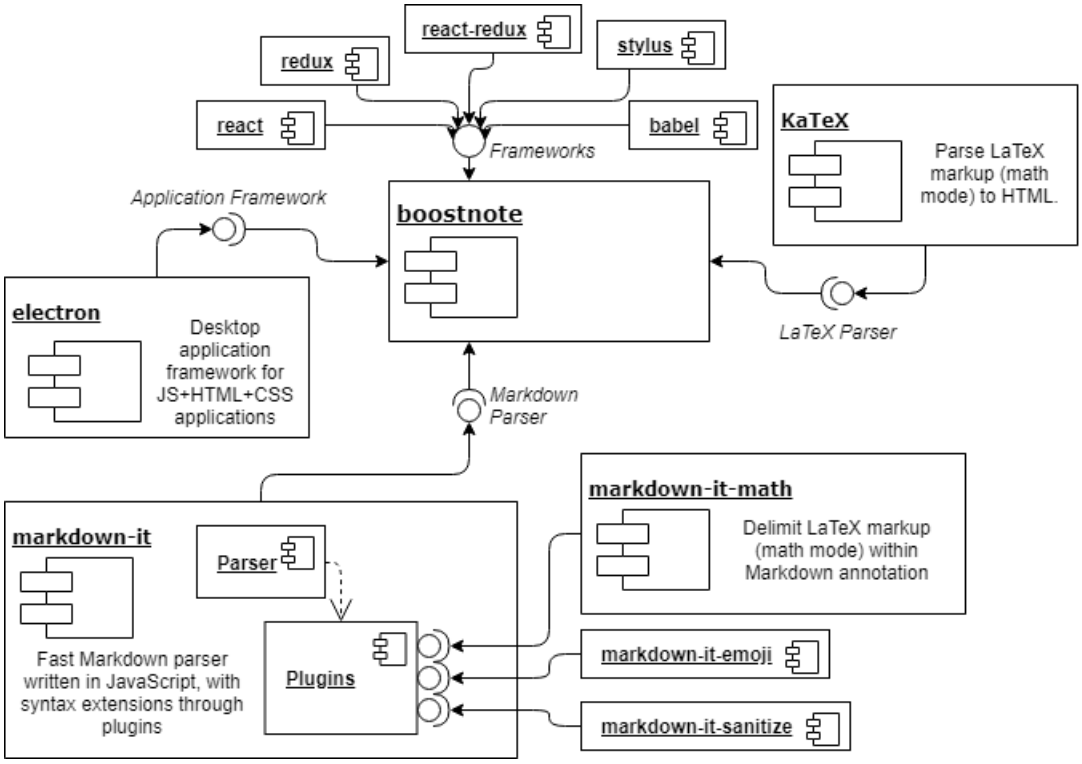
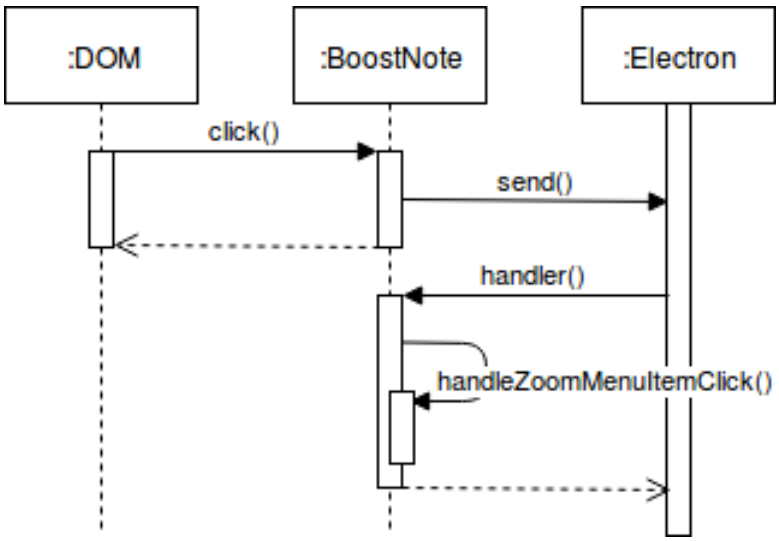


Figure 10: Boostnote's main dependencies

1.5 Design of the fix

The way to fix this issue is by either, getting the zoomFactor from the browser window instead of from the config or changing the menu item in the menu to not using the zoomin and zoomout role action, using instead another listener that also calls the config update.

In the following sequence diagram, it is explained how the zoomin/zoomout menu option click (which can be done with the hotkeys as well) will be handled. Boostnote will send a message to electron which will have a callback associated to it and will call the said function to handle it. The handler function will call handleZoomMenuItemClick function which is used by the zoom button to change the menu, it updates the zoom value on the config and on the browser window.



2 Issue 2557

When typing display math LaTeX in a markdown note, the written LaTeX is parsed correctly but displayed incorrectly, with certain constructs having line breaks between tokens.

2.1 Steps

1. Launch Boostnote
2. Open a new Markdown Note. Press *Ctrl N* or click on the icon next to the search bar; then select Markdown Note.
3. Toggle mode to two panels in the top right corner (should be the default). Now you should have two panels open, the edit and view panels.
4. Reproduce the error. Type the following in the edit panel:

```
$$
\begin{aligned}
   &P(X=x)=\binom{n}{k}P^x(1-p)^{(n-x)}\\
   &Var(X)=np(1-p)
\end{aligned}
$$
```

2.2 Requirements

As of release v0.11.9 of Sep 9, and upstream master branch as of Nov 14, we get this result:

$$\underset{(n-x)}{P(X=x)} = \binom{n}{k} P^x (1-p)$$

$$Var(X) = np(1-p)$$

However, the expected result is naturally:

$$P(X=x) = \binom{n}{k} P^x (1-p)^{(n-x)}$$
$$Var(X) = np(1-p)$$

The problem is that there is a line break where there should not be one: line breaks in display math mode are not allowed outside environments, and inside environment aligned line breaks are inserted with '\\', enumerating equations.

2.3 Source Code Files

The bug is the file:

- browser/components/markdown.styl

```
body
  font-size 16px
  padding 15px
  font-family helvetica, arial, sans-serif
  line-height 1.6
  overflow-x hidden
  background-color $ui-noteDetail-backgroundColor
  .flowchart-error, .sequence-error .chart-error
    background-color errorBackgroundColor
    color errorTextColor
    padding 5px
    border-radius 5px
    justify-content left
  .katex
    font 400 1.2em 'KaTeX_Main'
    line-height 1.2em
    white-space initial
    text-indent 0
  .katex .mfrac>.vlist>span:nth-child(2)
    top 0 !important
  .katex-error
    background-color errorBackgroundColor
    color errorTextColor
    padding 5px
    margin -5px
    border-radius 5px
li
  label.taskListItem
    margin-left -1.8em
    &.checked
      text-decoration line-through
      opacity 0.5
  &.taskListItem.checked
    text-decoration line-through
    opacity 0.5
div.math-rendered
  text-align center
.math-failed
  background-color alpha(red, 0.1)
  color darken(red, 15%)
  padding 5px
  margin 5px 0
  border-radius 5px
```

2.4 Source of the problem

The problem is not in the dependency (**KaTeX**). The **KaTeX** dependency includes its own major stylesheet, which should enforce this restriction on its own display math elements. So the problem should be in some of Boostnote's own style sheets, conflicting somehow with those of **KaTeX**.

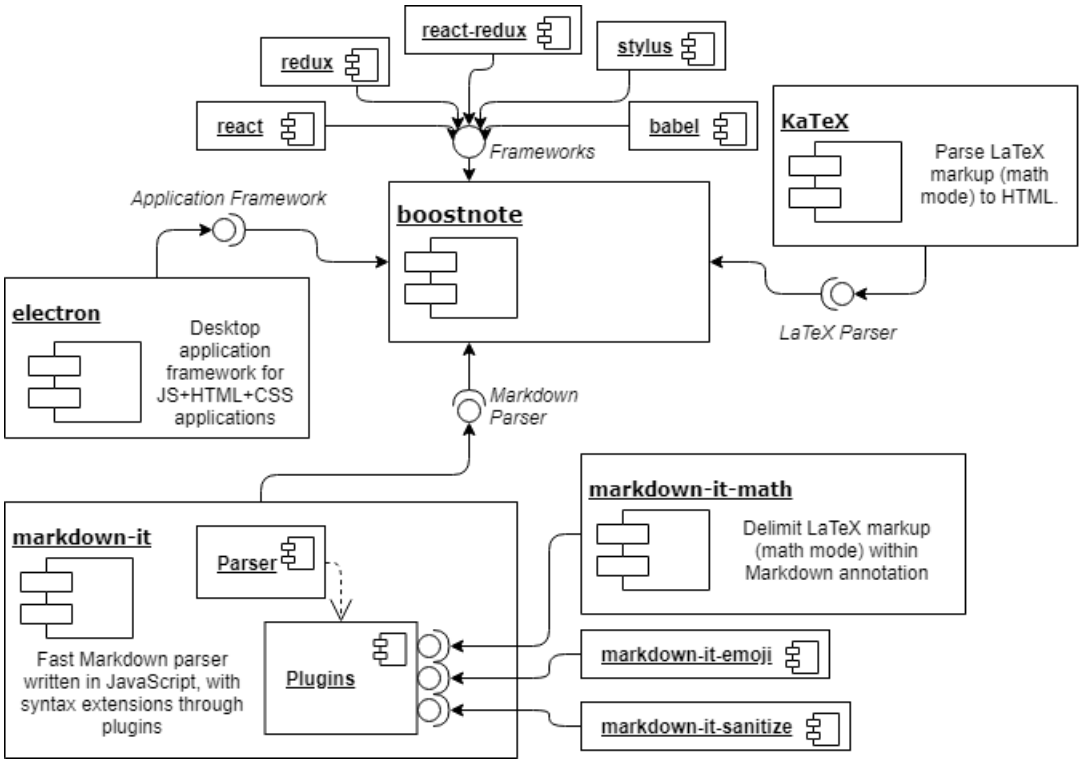
Inspecting the elements inside Boostnote (*Ctrl Shift I*) suggests looking at class names *.katex* and *.katex-display*, as the rendered LaTeX parent element has these class names. A quick

```
grep -rnE "\\.katex" browser/
```

immediately pinpoints file markdown.styl.

2.5 System Architecture

A simple rundown of Boostnote's main dependencies, with emphasis on markdown-it and katex:



2.6 Design of the fix

There are two ways to fix this issue. At katex.org we compare the computed styles of the LaTeX shown right on the front page to those of Boostnote applied to the same LaTeX elements. In particular we looked at attributes like display, textalign, and overflow-wrap. We concluded quickly that the problem was in the attribute white-space, whose value was reset to initial in Boostnote (meaning normal), instead of nowrap.

Changing this attribute to nowrap will have unintended side effects (no wrapping in inline math mode), so the problem, albeit simple, will require further testing. The most likely solution will involve demoting selector body .katex to .katex-inline or removing the attribute, or even removing the .katex rules altogether.