

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.

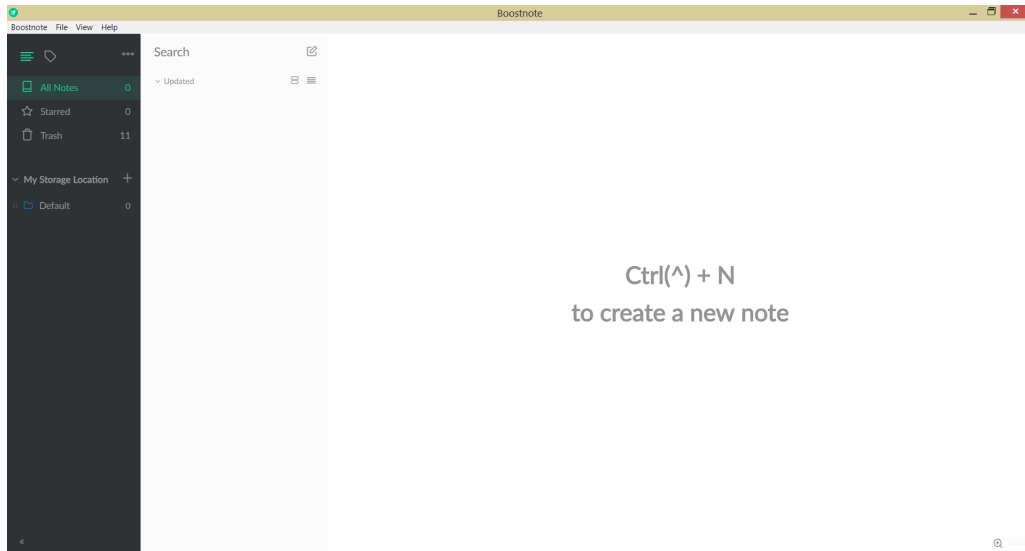


Figure 1: Normal Zoom

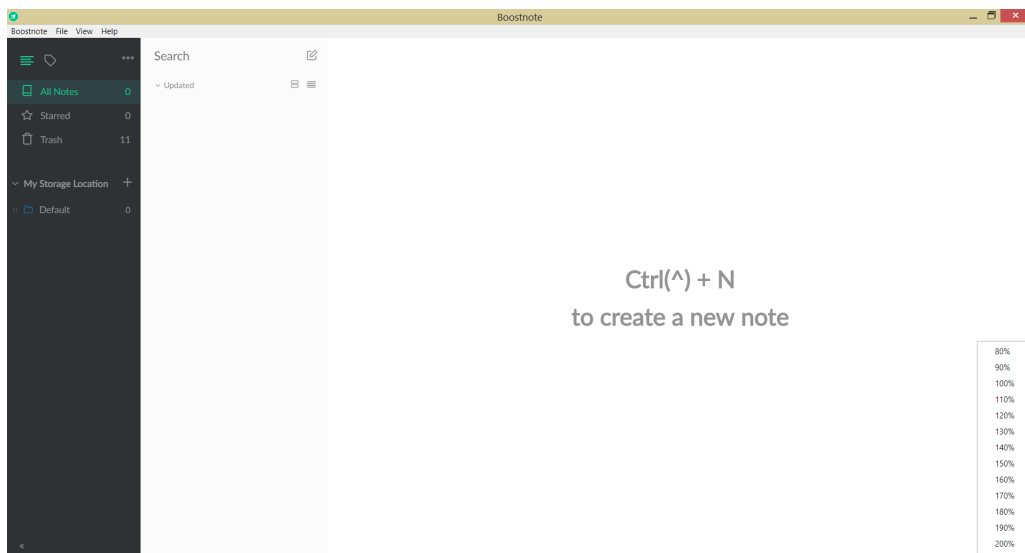


Figure 2: Zoom Menu

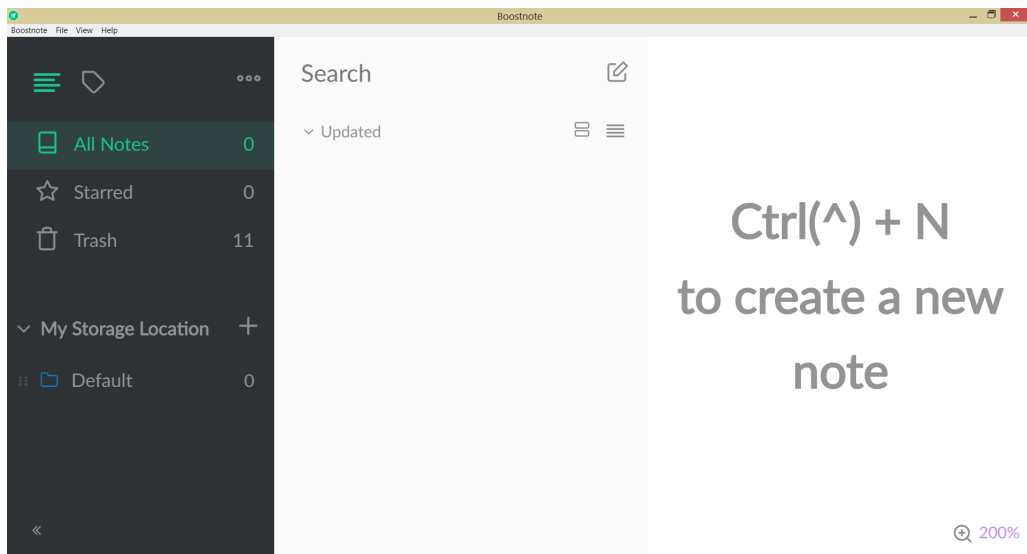


Figure 3: Zoom level at 200% and label update

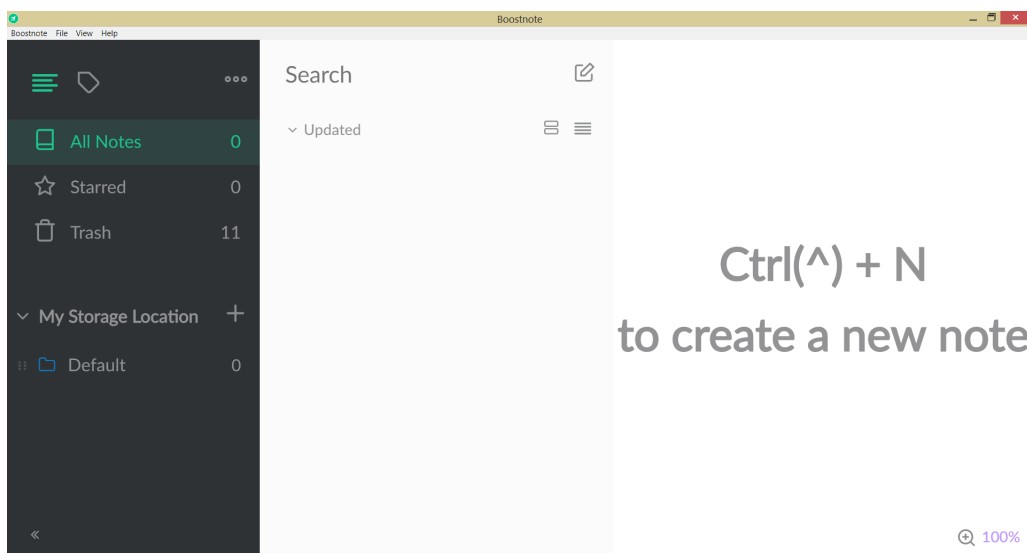


Figure 4: Zoom level increased with key shortcut and label not updated

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 items 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 inherits 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 gets the 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', 'ALPHABETICAL'
  },
  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 ? 'Sageo UI' : 'Menaco 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.

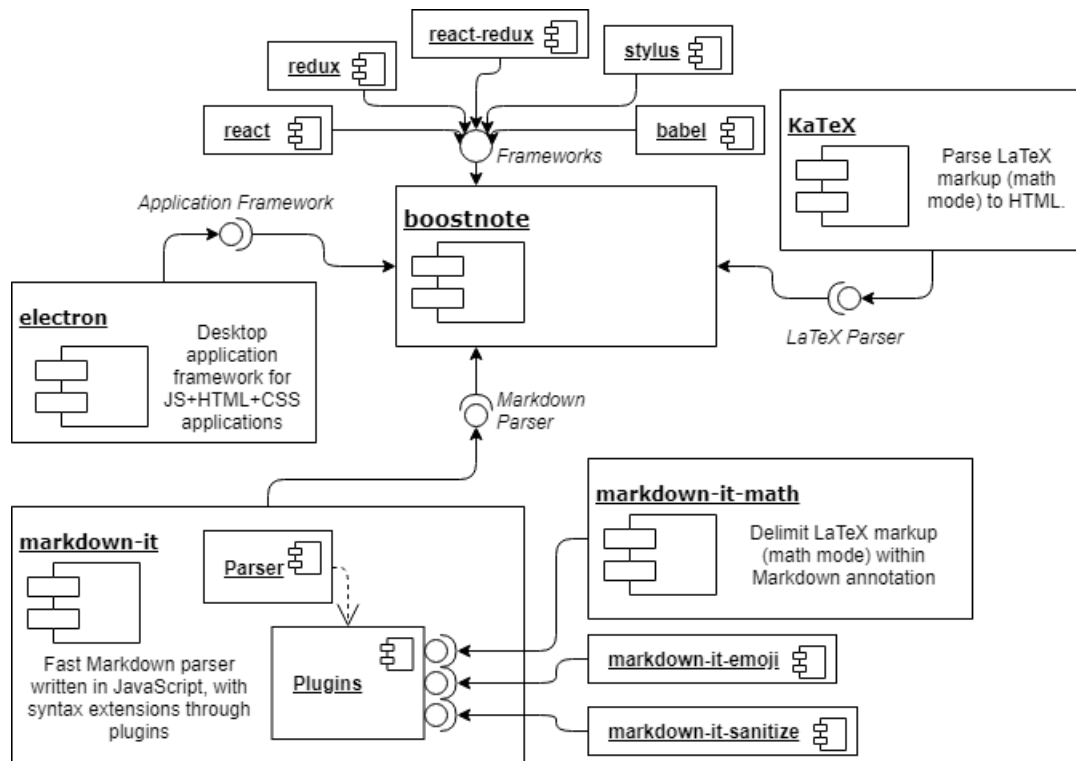


Figure 10: Boostnote's main dependencies

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.

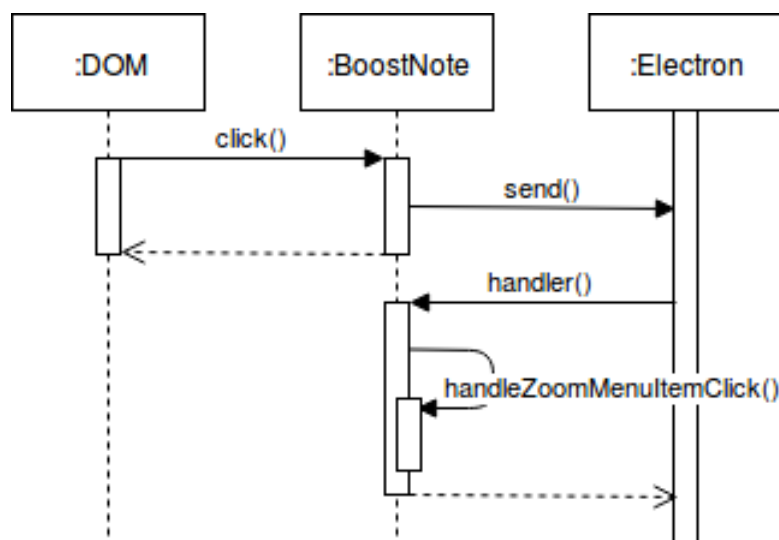


Figure 11: UML diagrams that illustrate the plan to fix the issue

1.6 Fix Source Code

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

Figure 12: main-menu.js

In this file, the menu itens for zoom out and zoom in were changed from roles to labels in order for a custom listener to be associated with click(), disabling the built in listener associated to the role zoomin and zoomout. An accelerator was associated with each of the itens, CommandOrControl, meaning the hotkey will work on mac as well, CommandOrControl+= meaning Ctrl+(or Cmd+) and CommandOrControl+- meaning Ctrl-(or Cmd-).

On the listeners, an electron request is sent which will be redirected to the respective handler.

```
componentDidMount () {
  this.handleZoomInBound = this.handleZoomIn.bind(this)
  this.handleZoomOutBound = this.handleZoomOut.bind(this)
  EventEmitter.on('statusbar:zoomin', this.handleZoomInBound)
  EventEmitter.on('statusbar:zoomout', this.handleZoomOutBound)
}

componentWillUnmount () {
  EventEmitter.off('statusbar:zoomin', this.handleZoomInBound)
  EventEmitter.off('statusbar:zoomout', this.handleZoomOutBound)
}
```

Figure 13: StatusBar/index.js

In this snippet, the React.js function componentDidMount is used to bind the handlers to the StatusBar component as soon as it is mounted (added to the tree) and the EventEmitter.js functions are used to associate the requests to the respective bound handlers. By binding the handlers to the component, they are able to call class functions using the 'this' keyword.

The function componentWillUnmount is also used to dissociate the requests from the respective handlers just before the component is unmounted and destroyed.

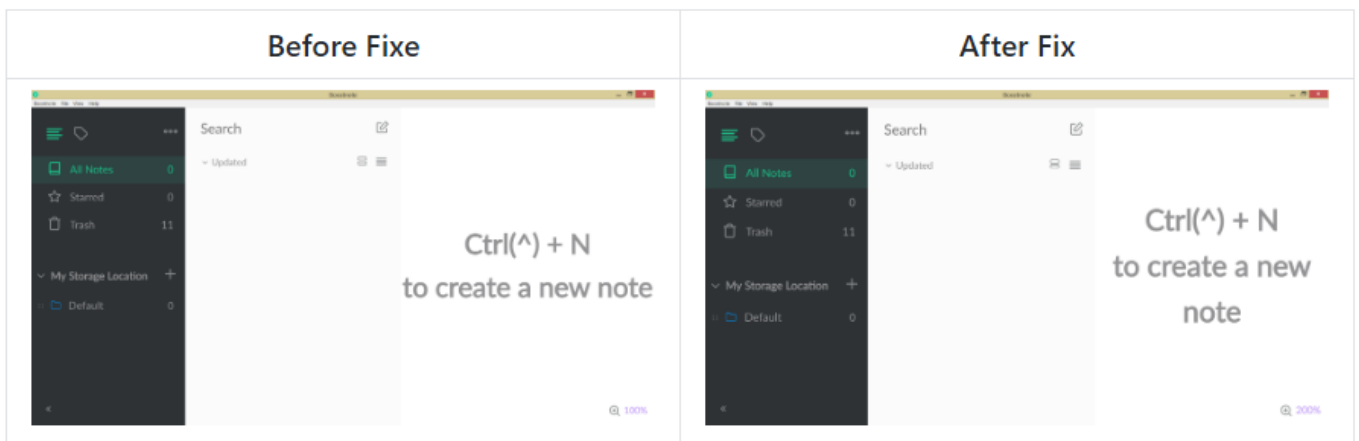
```
handleZoomIn () {
  this.handleZoomMenuItemClick(ZoomManager.getZoom() + 0.1)
}

handleZoomOut () {
  this.handleZoomMenuItemClick(ZoomManager.getZoom() - 0.1)
}
```

Figure 14: Handle Zoom

In this snippet, the handler functions are defined calling the already implemented function `handleZoomMenuItemClick` that changes the zoom value on the config and also on the browser window, receiving as parameter the new zoom which is the current zoom incremented or decremented by 0,1.

1.7 Validate the fix



Both images have a zoom value of 200%, changed by the key shortcut. Only after the fix, the value of the label in the lower right corner was changed.

- ☒ Peer review
- ☒ Easy to read and understand
- ☒ Efficient
- ☒ Lint Verified
- ☒ Passed all tests

1.8 Submission of the fix

Pull Request

1. The Pull Request has been submitted and approved;
2. The issue was indeed solved by a third party before us;
3. The issue is the same as in the first delivery;
4. Although the pull request was denied, we believe our solution to the issue is a bit more efficient and clean than the one previously accepted.