

A workaround for replace text issues in Slate.js



After release 5.6.0 on June 4, 2020 the described **workaround is no longer needed**. It has been implemented in the application core and now WProofreader works properly in Slate.js editor.

Slate.js editor has certain implementation specifics that don't work smoothly with external spellchecker tools and WProofreader is not an exception.

- Slate.js editor uses **beforeinput** event to listen to DOM changes.
- It listens to **selectionchange** event with 100ms delay.

It is possible to listen to our **customInsertText** event to prevent our text replacement and use internal Slate.js replace. Below an example how to archive that.

```
var el; // Get the main editable Slate element into this variable.

el.addEventListener('customInsertText', function(event) {
  event.preventDefault();

  setTimeout(function() {
    if (// For Safari browser) {
      replaceSafari(event.detail.text);

      return;
    }

    replaceForAll(el, event.detail.text);
  }, 100);
});

var replaceForAll = function(element, text) {
  var clipboardEvent = new ClipboardEvent('paste', {
    clipboardData: new DataTransfer(),
    bubbles: !0
  });

  clipboardEvent.clipboardData.setData('text/plain', text);

  element.dispatchEvent(clipboardEvent);
};

var replaceSafari = function(text) {
  document.execCommand('insertText', false, text);
};
```