

Nexteditor2

Der Zweck des JS Projektes Nexteditor 2 ist der Bau eines CKeditors5 aus den Quellen

Schritt 1 (Aufsetzen eines Editors)

Ich verfare nach den Anleitungen in „Getting started“:

<https://ckeditor.com/docs/ckeditor5/latest/installation/advanced/using-two-editors.html>

So verbaue ich mir für die Zukunft nicht die Möglichkeit einen ClassicEditor und InlineEditor gleichzeitig zu benutzen. Das dort aufgeführte „git clone“ gefolgt von „yarn install“ Installiert eine ganze Menge, ist aber nur zu Referenz Zwecken nützlich. Das Gleiche kann man direkt in gitHub nachschauen und es ist alle in der ts Version, so dass es für copy/paste nicht geeignet ist. Ich verzichte deshalb darauf.

Besser geeignet sind die Anleitungen in „Getting started > Customized Installation > Building the editor from source“ <https://ckeditor.com/docs/ckeditor5/latest/installation/getting-started/quick-start-other.html>. Zu beachten ist, dass alle Bestandteile die gleiche Version haben

Keep in mind, however, that all packages (excluding @ckeditor/
ckeditor5-dev-*) **must have the same version as the base editor package.**

Ich denke ein `npm update` könnte bei nachträglichem Hinzufügen von Modulen sicherstellen, dass alle zur gleichen Version gehören.

package.json von [github/ckeditor5/packages/ckeditor5-build-classic](https://github.com/ckeditor5/packages/ckeditor5-build-classic) (ctrl-c im Browser) nach nexteditor2 kopiert und mit `npm install` die Module installiert. Die Quellen in `node_modules` sind nun sowohl als `js` als auch als `ts` vorhanden. Ich musste einzig die Zeile `"postbuild": "tsc -p ./tsconfig.types.json"` in „scripts“ entfernen, die wahrscheinlich `ts` spezifisch ist, damit dann der build funktioniert.

webpack.config.js aus „ispencil5_2“ kopiert. `entry` und `output.filename` auf `isCkeditor.js` geändert.

isCkeditor.js von „Getting started > Advanced concepts > Alternative setups > Integrating from Source using webpack“ kopiert. Dort heisst es `ckeditor.js`.

Mit `yarn build` kann nun das Verzeichnis `build` erstellt werden und darin befindet sich die gebaute Datei `isCkeditor.js` mit der dazugehörigen `map`.

index.php und **testdocuments** von `ispencil5_2` nach nexteditor2 kopiert. An die neue Umgebung angepasst. Für die Darstellung von Zeichnungen ausserhalb des Editors, das Skript von `ispencil5_2` angepasst.

```
$html .= '<script type="module">';  
$html .= 'import {attachIsPencil} from "../ispencil5_2/ispencil/ispen/ispenengine.js";';  
$html .= 'attachIsPencil(\''.$jsonParams.'\');';  
$html .= '</script>';
```

index.css von `ispencil5_2` nach nexteditor2 kopiert.

Nun funktioniert ein Editor, aber er hat noch wenige Möglichkeiten, was auch mit `isCkeditor.js` zusammenhängt.

Schritt 2 (Plugins installieren)

Eine Liste aller verfügbaren Plugins befindet sich in „Getting started > Plugins > Plugins and HTML output“.

Autoformat ist ein Plugin, das es erlaubt mit geschriebenen Kurzbefehlen Text zu formatieren. Z.B macht # gefolgt von space, dass alles danach geschrieben ein Titel ist. Das würde in isTest nur Verwirrung schaffen. Deshalb habe ich Autoformat entfernt.

BlockQuote erstellt einen Zitat Block. Ich denke, das sollte man in der Konfiguration weglassen, aber ich belasse es im Code.

ckeditor5-basic-styles. Zusätzlich zu Bold und Italic, auch Underline, Strikethrough, Code, Subscript, Superscript importiert.

CKBox This is a premium feature and you need a subscription to use it, Kann Bilder hochladen. Ich lasse es weg.

CKFinder his is a premium feature and you need a license for it on top of your CKEditor 5 commercial license. Es ist ein Filemanager, aber das haben wir selbst gemacht.

Cloud Services Weggelassen

Code blocks Kann HTML und CSS Blocks einfügen. Könnte eine Option sein für den Programmierunterricht. Weggelassen.

Comments Kann individuelle Kommentar zu Text und Block Elementen hinzufügen. Collaboration Eigenschaft. Weggelassen.

Document outline This premium feature is a part of the [Productivity Pack](#) available only for customers with a commercial license. Weggelassen

Easy image This is a premium feature and you need a license for it on top of your CKEditor 5 commercial license. Weggelassen.

Essentials. Ist schon in der ersten Vorlage dabei.

Export to PDF This is a premium feature and you need a subscription to use it. Nicht notwendig, da alle Editor Produkte ausgedruckt werden können und man zu PDF drucken kann.

Export to Word This is a premium feature and you need a subscription to use it. Weggelassen.

Find and Replace. Hinzugefügt. Mit folgendem Prozeder fuktionstüchtig gemacht

- "@ckeditor/ckeditor5-find-and-replace": „39.0.2“ zu package.json hinzugefügt
- mit npm update in node_modules geladen
- Zu isCkeditor.js in builtinPlugins und defaultConfig hinzugefügt
- mit yarn build gebaut

Font family Erlaubt Grösse, Schrift-Familie, Farbe und Hintergrundfarbe zu wählen. Muss noch speziell konfiguriert werden.

- "@ckeditor/ckeditor5-font": "39.0.2"
- mit npm update in node_modules geladen
- In isCkeditor.js zu builtinPlugins hinzugefügt und in der Toolbar eine neue Sektion mit fontSize, fontFamily, FontColor, fontBackgroundColor gemacht. Weitere Konfigurationen sind möglich, aber nicht notwendig.
- mit yarn build gebaut und dann getestet.

Format painter This premium feature is a part of the Productivity Pack available only for customers with a commercial license. The format painter feature lets you copy text formatting (such as bold, italic, font size, color, etc.) and apply it in a different place in the edited document. It helps keep the formatting consistent and speeds up the creation of rich content.
Weggelassen

Heading Kann HTML Elemente h1, h2, etc. produzieren. War schon installiert.

Highlight Weggelassen. Es ändert die Textfarbe, ist aber weniger mächtig als das schon installierte Font.

Horizontal line Fügt via Button or im Text via drei Minus eine horizontale Linie ein. Ich wollte die übliche Installationsmethode anwenden, aber es gab Schwierigkeiten. package.json wurde wieder mit dem ursprünglichen aus gitHub kopierten package.json ersetzt. Ich konnte mit undo zu meinem laufenden zurückkehren, habe dabei aber bemerkt, dass es module hat, die man nicht importieren sollte.

Bemerkung:

Zuerst dachte ich, sie müssen aus package.json entfernt werden, dann aber bin ich zum Schluss gekommen, dass es nur wichtig ist, dass sie in isCkeditor.js nicht vorkommen. Was alles in node_modules steht ist, solange es nicht gebraucht wird, irrelevant. Zusätzlich habe ich bemerkt, dass viele Plugins schon importiert werden, aber nicht sichtbar sind, weil die Toolbar Items nicht in defaultConfig aufgeführt werden.

Horizontal line war nicht so ein Fall. Das Modul musste neu importiert werden und das Plugin zu builtinPlugins zugefügt werden und ‚horizontalLine‘ zu defaultConfig.toolbar.items.

HTML embed Ist ein Sicherheitsrisiko. Man müsste einen sanitizer selbst einbauen. Einige LK haben gewünscht HTML schreiben zu können, das nur angezeigt wird, aber die Problematik muss noch studiert werden. Weggelassen

General HTML Support Weggelassen.

HTML comment Weggelassen.

Image Das Modul ist schon in package.json. Die einzelnen Bestandteile müssen noch in isCkeditor.js geladen werden. Die einzelnen Plugins sind:

AutolImage Erkennt Bilder in URL's und fügt diese ein.

Image selbst bedient ImageBlock und ImageInline, so dass diese Plugins nicht geladen werden müssen.

Installation:

- In package.json ist nichts mehr notwendig, ckeditor5-image ist schon vorhanden und ckeditor5-upload auch. ckeditor5-image ist in package.json aufgeführt, ckeditor5-upload aber nicht. Das muss als Teil eines anderen Moduls hochgeladen worden sein.
- In isCkeditor.js import { SimpleUploadAdapter } from ‚@ckeditor/ckeditor5-upload‘; und import { Image, ImageCaption, ImageResize, ImageStyle, ImageToolbar, ImageUpload } from ‚@ckeditor/ckeditor5-image‘; importiert. Ohne ImageUpload fehlt das Toolbar Icon zum Hochladen.
- Bei den builtinPlugins SimpleUploadAdapter, ImageUpload, Image, ImageCaption, ImageResize, ImageStyle, ImageToolbar hinzugefügt
- In defaultConfig.toolbar.items uploadImage hinzugefügt. Das ist das Icon mit dem man zur Auswahl an Dateien, die hochgeladen werden sollen, gelangt. Es stammt aus dem Plugin ImageUploadEditing, das vom Plugin UploadImage geladen wird.
- Die Konfiguration von Image als defaultConfig.image.toolbar hinzugefügt. Selbst ist sie ein array ['imageStyle:inline', 'imageStyle:block', 'imageStyle:side', '|', 'toggleImageCaption', 'imageTextAlternative'] Das ist die Konfiguration der Baloon Toolbar, die beim Anklicken eines hochgeladenen Bildes erscheint.
- Damit funktionieren Bilder noch nicht. SimpleUploadAdapter liefert nur den Client Teil. Den Server Teil habe ich selbst programmiert. Von nexteditor her isUpload.php kopiert, das für image und audio gut sein sollte.

- Die Konfiguration von simpleUpload.php mit defaultConfiguration.simpleUpload: { uploadUrl: ,https://myeclipse/nexteditor2/isUpload.php } definiert. Die Dateien landen in eclipseRoot/appImages/ bzw. eclipseRoot/appAudios.
- Getestet. Es funktioniert auch mit drag/drop.

Import from Word This is a premium feature and you need a subscription to use it. Weggelassen

Indent Einzug Das Modul ist schon in package.json. Modul importiert import { Indent, IndentBlock } from '@ckeditor/ckeditor5-indent'; Beide Plugins importiert durch hinzufügen zu builtinPlugins. outdent, indent, zu toolbar.items hinzugefügt. Es ist zu beachten, dass indent und outdent markierte Blöcke verschiebt und den Grad der Schachtelung in den Listen mit Punkt oder Zahl.

Text part language. Markiert einen Teil eines Textes als einer Sprache zugeordnet. Nützlich für screen readers. Weggelassen

Link Auto link und Link sind schon implementiert. Man beachte, dass externe Links mit https:// beginnen müssen, sonst wird intern gesucht.

List ist schon implementiert.

Markdown Erlaubt markdown als Alternative zu HTML. Weggelassen

Media embed ckeditor5-media-embed ist schon in package.json. Ich habe es als „MediaEmbed“ zu builtinPlugins und ohne weitere Konfiguration als „mediaEmbed“ zu toolbar.items hinzugefügt. Tests waren nicht erfolgreich. Ein YouTube Video kann zwar im Editor hinzugefügt werden und es wird auch angezeigt, kann aber nicht abgespielt werden. Ladet man das gespeicherte Dokument, ist nichts sichtbar. Die URL wird unter dem Tag <oembed> gespeichert. Im Editor ist ein YouTube Video nicht ausführbar, aber mit der zusätzlichen Konfiguration defaultConfib.mediaEmbed = { ,previewInData': ,true' } werden sie im HTML output ausführbar. Beachte ,true' muss als string eingegeben werden (ohne Anführungszeichen geht es nicht). Es wird ein iframe eingefügt

```

▼ <figure class="media">
  ▼ <div data-oembed-url="https://www.youtube.com/watch?v=H08tGjXNH04">
    ▼ <div style="position: relative; padding-bottom: 100%; height: 0; padding-bottom: 56.2493%;">
      ▶ <iframe src="https://www.youtube.com/embed/H08tGjXNH04" style="position: absolute; width: 100%; height: 100%; top: 0; left: 0;" frameborder="0" allow="autoplay; encrypted-media" allowfullscreen>...</iframe>
    </div>
  </div>
</figure>

```

Mention Erlaubt Autocompletion an Hand von Abkürzungen. Weggelassen

Minimap The content minimap feature shows a miniature overview of your content. Weggelassen

Page break. Zu package.json hinzugefügt. Mit npm update das Paket @ckeditor/ckeditor5-page-break zu node_modules hinzugefügt. In isCkeditor.js mit import { PageBreak } from ,@ckeditor/ckeditor5-page-break'; importiert, zu builtinPlugins PageBreak hinzugefügt und in toolbar pageBreak.

Pagination Zahlungspflichtige premium feature. Erlaubt einzusehen, wo es in PDF oder Word einen Seitenumbruch geben würde, falls man zu diesen Programmen exportiert. Weggelassen.

Paste from Office Das Modul @ckeditor/ckeditor5-paste-from-office ist in package.json schon vorhanden. Ich habe es nur importiert und zu builtinPlugins hinzugefügt. Es hat kein Icon in der Toolbar, da es mit ctrl-v funktioniert.

Paste from Office Enhanced. Verbesserte premium Version. Weggelassen.

real-time-collaboration Weggelassen

Remove format Man kann von ganzen markierten Sektionen alle Formatierungen wie Farbe Schrift etc. auf einmal entfernen. Das Modul `@ckeditor/ckeditor5-remove-format` in `package.json` aufgenommen, mit `npm update` in `node_modules` installiert, in `isCkeditor.js` importiert und dann in `builtinPlugins` und `toolbar` installiert.

Restricted editing mode The restricted editing feature introduces two modes: the standard editing mode and the restricted editing mode. Users working in the restricted editing mode cannot change the content, except for parts marked as editable. Weggelassen, aber vielleicht könnte es gerade in `isTest` nützlich sein.

Revision History Premium feature weggelassen

Select all Erlaubt `ctrl-a`. Ist schon mit Essentials installiert

Show Blocks Zeigt alle Blocks im Sinn von HTML durch rahmen und Tag an. Weggelassen

Slash command. Premium feature. Erlaubt die Ausführung von Befehlen wie z.B. File manager öffnen. Weggelassen

Source editing Erlaubt es zwischen der gewohnten Darstellung und HTML hin und her zu wechseln. Könnte bei der Fehlersuche nützlich sein. Paket zu `package.json` hinzugefügt, `npm update`, importiert, zu `builtinPlugins` und `toolbar` hinzugefügt. Getestet in beiden Richtungen.

Special Characters. Sonderzeichen, kann erweitert werden. Wenn neue Sonderzeichen erwünscht sind, kann man das eventuell machen. Nach der üblichen Prozedur war das Icon vorhanden, aber es gab keine Sonderzeichen. Erst durch zusätzliches Laden von `SpecialCharactersEssentials` hat es wie in der Demo funktioniert. Man könnte einzelne Kategorien von Sonderzeichen laden und diese namentlich erwähnen oder alle vordefinierten Kategorien und das sind die Essentials. Das Verfahren war das Package `@ckeditor/ckeditor5-special-characters` in `package.json` aufzunehmen und mit `npm update` in `node_modules` zu laden. Danach mit `import { SpecialCharacters, SpecialCharactersEssentials } from '@ckeditor/ckeditor5-special-characters'`; das Grundmodul und alle vordefinierten Kategorien importiert und beide Plugins in `builtinPlugins` geladen. In der Toolbar genügt `specialCharacters`.

Style Wendet eine oder mehrere CSS Klassen an. Im Wesentlichen schon mit `basic-styles` installiert-

Table Die Grundinstallation mit den Plugins `Table` und `TableToolbar`
`import { Table, TableToolbar, TableProperties, TableCellProperties } from '@ckeditor/ckeditor5-table'`; In `toolbar` nur `insertTable`, aber keine Konfiguration
`table: {`
 `'contentToolbar' : ['tableColumn', 'tableRow', 'mergeTableCells', 'tableProperties',`
 `, 'tableCellProperties']`
`},`

Template The template feature allows you to insert predefined content structures into the document. Premium feature. Weggelassen

Track changes Overview Premium feature, Weggelassen.

Text transformation The text transformation feature enables autocorrection. It automatically changes predefined text fragments into their improved forms. Das wäre für manche `isTest` Anwendung schädlich, z.B. die Transformation von $3/4$ in das typographische Äquivalent würde Mathematik fast unmöglich machen. Weggelassen

Block Toolbar The block toolbar plugin provides an additional [configurable](#) toolbar on the left-hand side of the content area. The block toolbar comes in handy when you cannot access the main editor toolbar. Weggelassen

Base64 upload adapter. Weggelassen. isTest arbeitet mit dem SimpleUploadAdapter in Verbindung mit einem Server Skript

Watchdog The watchdog utility protects you from data loss in case the editor crashes. It saves your content just before the crash and creates a new instance of the editor with your content intact. Kann nur mit einem Build aus den Quellen erstellt werden und verkompliziert den ganzen Prozess enorm, da es einen Mantel um Create erstellt It also means that any code that should be executed for any new editor instance should be either loaded as an editor plugin or executed in the callbacks defined using EditorWatchdog#setCreator() and EditorWatchdog#setDestructor(). Read more about controlling the editor creation and destruction in the next section. Könnte etwas Gutes sein, aber ich lasse es vorerst weg.

Word count Das package @ckeditor/ckeditor5-word-count zu package.json hinzugefügt und mit npm update in node_modules importiert. In isCkeditor.js import { WordCount } from '@ckeditor/ckeditor5-word-count'; das plugin importiert und WordCount zu builtinPlugins hinzugefügt. Es gibt keinen Toolbar Eintrag, denn die Zählung wird extern angezeigt. Dazu braucht es im HTML einen div mit id=„word-count“. Beim Instantieren des Editors, kann man dann WordCount in diesem div darstellen.

```
$html .= '<textarea id="editor" name="content">'.$this->currentHtml.'</textarea>';  
$html .= '<div class="smallspacer"></div>';  
$html .= '<div id="word-count"></div>';  
$html .= '<script>';  
$html .= $this->createEditorScript();  
$html .= '</script>';
```

Es ist wichtig, dass das div mit word-count vor dem Skript steht, denn es wird im Skript benutzt

```
ClassicEditor  
  .create( document.querySelector( '#editor' ), {  
  
  } )  
  .then( editor => {  
    console.log('editor ready', editor);  
    const wordCountPlugin = editor.plugins.get( 'WordCount' );  
    const wordCountWrapper = document.getElementById( 'word-count' );  
    wordCountWrapper.appendChild( wordCountPlugin.wordCountContainer );  
  } )  
  .catch( error => {  
    console.error( error );  
  } );
```

Die Zählung wird erst bei der Instantierung des Editors extern angehängt. Das erlaubt eine grosse Flexibilität in der Darstellung.

MathType Die Installation gelingt, wie in der CKeditor5 Dokumentation beschrieben, aber mein lokaler Apache scheint noch nicht ganz auf der Höhe, denn es funktioniert, gibt aber die Meldung: `WebAssembly.instantiateStreaming` failed because your server does not serve wasm with `application/wasm` MIME type. Falling back to `WebAssembly.instantiate` which is slower. Original error:

Installation:

- `npm install --save @wiris/mathtype-ckeditor5`
- `import MathType from '@wiris/mathtype-ckeditor5'`; Beachte, dass es keine geschweifte Klammer um MathType geben darf. Eventuell hängt dies damit zusammen, dass es keine Quelle, sondern nur ein Package gibt. Ich habe es mit Klammer versucht und es hat nicht funktioniert.
- MathType zu `builtinPlugins` hinzufügen
- MathType und ChemType zu `defaultConfig.toolbar` hinzufügen

Nun funktioniert es soweit mit MathType, ist aber langsam, was erklärbar ist mit dem Hinweis `our server does not serve wasm with `application/wasm` MIME type`, Mit ChemType gibt es Probleme, weil anscheinend LaTeX fehlt. Erst mit

```
// Version 2 von mathjax mit cdn laden.  
// Version 3 hat noch nicht alle Funktionen  
$html .= '<script async src=„https://cdn.jsdelivr.net/npm/mathjax@2/  
MathJax.js?config=TeX-AMS-MML_CHTML"></script>';
```

im `<head>` Teil von `index.php` funktioniert fast alles korrekt. Es gibt noch einen Fehler

Refused to execute `https://cdn.jsdelivr.net/npm/mathjax@2/config/TeX-AMS-MML_CHTML.js?V=2.7.9` as script because "X-Content-Type-Options: nosniff" was given and its Content-Type is not a script MIME type.

Dieser Fehler muss im HTTP Server behoben werden, indem man den Mime Typ `wasm` hinzufügt. Vereinfacht geht da in `.htaccess`, aber man sollte es in der Konfiguration von `apache` machen, weil das weniger ausbremst. Der Eintrag ist:

```
AddType application/wasm wasm
```

Ich habe alternativ zum obigen Version 2 Skript MathJax 3 versucht und das ist schneller und besser.

```
// Version 3  
$html .= '<script src=„https://polyfill.io/v3/polyfill.min.js?  
features=es6"></script>';  
$html .= '<script id="MathJax-script"  
async src=„https://cdn.jsdelivr.net/npm/mathjax@3/es5/  
tex-mml-cthtml.js"></script>';
```

WProofreader Ein zahlungspflichtiges Korrektur Modul. Weggelassen.

Die Version mit allen fremden Plugins auf `gitHub` hochgeladen. Zuerst in `gitHub` ein neues Repository `nextedito2` erzeugt und dann mit

```
git remote add origin git@github.com:anbrusi/nexteditor2.git  
git branch -M main  
git push -u origin main
```

auf `gitHub` als Sicherung hochgeladen.

Schritt 2 (Vorläufige Version auf Cyon installieren)

Die Subdomain ‚test.misas.ch‘ enthält ein HTML Dokument index.html und die beiden Ordner ‚appliedimages‘ und ‚appAudios‘ in denen Bilder und Audios gespeichert werden.

Der Ordner ‚nexteditor‘ enthält den Editor isEditor.js, die Dateien für die Demo Anwendung und den Ordner ‚testdocuments‘ für die von der Demo erzeugten Dokumente.

Zu beachten gilt:

- isCkeditor muss im `—mod production` gebaut werden, denn die Dateien aus `node_modules` in `nexteditor2` stehen nicht mehr zur Verfügung.
- Die Pfade in `index.php` und `isUpload.php` müssen an die neue Umgebung angepasst werden
- Die `.htaccess` Datei mit „AddType application/wasm .wasm“ funktioniert zwar mit dem lokalen apache, nicht aber auf cyon

Schritt 3 (Installation von SimpleAudio)

Auf nexteditor2 eine branch „audio“ erstellt

```
git checkout -b „audio“
```

SimpleAudio installiert

```
npm install https://github.com/anbrusi/simple-audio
```

In isCkeditor.js SimpleAudio importiert

```
import SimpleAudio from '@anbrusi/ckeditor5-simple-audio';
```

Es hat keine geschweifte Klammer im import, weil SimpleAudio der default export ist ‚SimpleAudio‘ zu den builtinPlugins hinzugefügt, ‚simpleAudioButton‘ zur Toolbar hinzugefügt. Es heisst wirklich simpleAudioButton, weil ich es im Plugein leider so genannt habe.

Mit yarn build isCkeditor i, production mode gebaut. Die Dateien aus build nach test/nexteditor in Cyon hochgeladen und getestet.

Defekt (sollte man bei Gelegenheit überarbeiten)

Es hat noch den gleichen Defekt, wie MediaEmbed. Wenn SimpleAudio als letztes Element im Editor eingefügt wird, kann man es weder löschen noch etwas danach einfügen. Wenn man es hingegen weiter oben einfügt, ist alles ok. Bei Bilder hat es dazu spezielle handles über und unter dem Bild. Vielleicht sollte man aus SimpleAudio ein Widget machen.

Mit folgenden Befehlen auf gitHub gespeichert

```
git add .
git commit -m "added audio"
git merge audio
git checkout main
git merge audio
git push
```

Das gebaute isCkeditor von nextedito2 nach test kopiert und von dort auf Cyon hochgeladen. Nun funktioniert es auch auf test.misas.ch