

Hybrid Topic

Tiptap

- **Door:** Michiel Truijen
- **Demo datum:** 10
- **Code url github:**
 - <https://github.com/TruijenM/michiel-truijen-hybrid-topic>

1 Korte uitleg topic

*We gebruiken tiptap om een tekstveld met editor functionaliteiten te maken
Voor de demo willen we een editor die ons toegang geeft tot:*

- 2 types hoodfing
- Bold tekst
- Italic tekst

1.1 Korte beschrijving

Tiptap is een toolkit die je toestaat om verschillende extensions te gebruiken om je eigen texteditor te maken.

Je kan gebruik maken van de basis starterkit of uitbreiden met extra voorgemaakte extensions.

1.2 Prerequisites

Wat moet al gekend zijn voor je aan dit topic begint:

- React/tailwind/shadcn (Button)

Welke tools moeten geïnstalleerd zijn voor je aan dit topic kan beginnen:

- Een nextjs/react project met tailwind en shadcn

Info om deze tools te installeren (als het gaat om iets dat we **niet** in de lessen gezien hebben):

-

1.3 Stappen

Beschrijf in kleine stapjes hoe je medestudent dit kan maken in zijn eigen code voorbeeld.

Voorkennis van je medestudent: alles wat we in de lessen geleerd hebben.

Je doet je demo aan de hand van dit stappenplan.

1. Installatie

Installeer de benodigde packages met het volgende commando:

```
npm install @tiptap/react @tiptap/pm @tiptap/starter-kit
```

Dit installeert de basisfunctionaliteiten van Tiptap en de StarterKit, die enkele standaardextensies bevat.

2. Projectstructuur

Maak een nieuwe map components, waarin we de editor- en toolbarcomponent zullen aanmaken:

- **Tiptap.tsx:** Bevat de hoofdeditor.
- **Toolbar.tsx:** Bevat de knoppen voor het aanpassen van de tekst.

3. De Editor-component

In de **Tiptap.tsx**-component gebruiken we de useEditor-hook om een editorobject aan te maken met de volgende eigenschappen:

- **extensions:** `[StarterKit.configure({heading: {levels: [1, 2]}})]`,
We laden de StarterKit en configureren de heading-extensie om alleen niveau 1 en 2 toe te staan.
- **attributes:** `{class: 'prose prose-sm sm:prose-base lg:prose-lg xl:prose-2xl m-5 focus:outline-none'}`
- We voegen Tailwind-klassen toe voor de styling van de editor.
- **Content:** We stellen een standaardinhoud in met een heading en een paragraaf.

De editor wordt gerenderd met:

- `<Toolbar editor={editor} />` voor de knoppenbalk.
- `<EditorContent editor={editor} />` voor de teksteditor zelf.

4. De Toolbar-component

In de **Toolbar.tsx**-component voegen we knoppen toe voor het bewerken van de tekst. Dit omvat:

- Heading 1 en 2
- Vetgedrukte en cursieve tekst

- Extra knoppen om de JSON-structuur en HTML-output van de editor te tonen

Elke knop krijgt een onClick-event dat een actie uitvoert op de editor, zoals het togglen van een heading of het activeren van bold/italic. De actieve status van een knop wordt weergegeven met de variant-eigenschap.

```
<Button
  onClick={ (e) => {
    e.preventDefault();
    editor.chain().focus().toggleHeading({ level: 1 }).run();
  }}
  size="sm"
  variant={editor.isActive('heading', { level: 1 }) ? 'default' :
'ghost'}
>
  <Heading1Icon />
</Button>
```

5. Toevoegen van Tailwind Typografie

Om de styling van de editor correct weer te geven, voeg je in **tailwindconfig.ts** de volgende regel toe aan de plugins:

```
require('@tailwindcss/typography')
```

1.4 Referenties

Geef hier links, boeken, etc van plaatsen waar je info opgezocht hebt.

- <https://tiptap.dev/docs>

2 Conclusie

Geef hier de conclusie: Vond je dit zelf interessant? Vond je dit zelf moeilijk?

Ik vond origineel het idee van de tiptap editor interessant voor mijn persoonlijke project. Maar achteraf gezien ben ik geen fan. Voor mij project is een simpel text input veld alles dat ik nodig heb.

Voor een app waar je bevoorbeeld artikels met styling etc... moet schrijven kan dit echter wel de moeite waard zijn.

Ik vond het moeilijk om sommige dingen aan de praat te krijgen.

Origineel probeerde ik een color input toe te voegen maar na een volledige van trial & error en fixes besloot ik om dit te laten vallen.

Als je de tijd neemt om een goede editor te bouwen voor je project raadt ik ook aan om zelf de styling te voorzien i.p.v. de tailwind prose klasse.