

Hybrid Topic

Next-intl

- **Door:** Rob Schoovaerts
- **Demo datum:** 10/03/2025
- **Code url github:**
 - <https://github.com/ROB2005/rob-schoovaerts-hybrid-topic.git>

1 Korte uitleg topic

1.1 Korte beschrijving

next-intl is een handige tool om meertalige websites te bouwen in Next.js. Het helpt je om teksten in verschillende talen te beheren en weer te geven, zodat je website toegankelijk is voor gebruikers over de hele wereld.

1.2 Prerequisites

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

- Basiskennis van React en React Hooks
- JavaScript/TypeScript fundamentals
- HTML en CSS basiskennis
- Begrip van routing in web applicaties
- Basiskennis van npm en package management

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

- Node.js
- npm
- VS Code of een andere code editor

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.

- **Project initialisatie**

```
npx create-next-app@latest multilingual-next-app
```

- **Installeer next-intl**

```
cd multilingual-next-app
npm install next-intl@latest
```

- **Configureer next-intl**

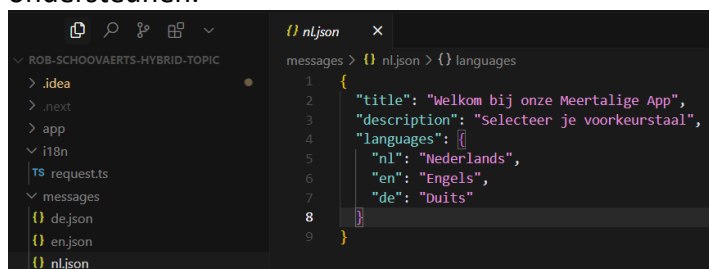
- **Maak next.config.mjs in root**

```
JS next.config.mjs > ...
1 import createNextIntlPlugin from 'next-intl/plugin';
2
3 const withNextIntl = createNextIntlPlugin();
4
5 /** @type {import('next').NextConfig} */
6 const config = {
7   reactStrictMode: true,
8 };
9
10 export default withNextIntl(config);
```

- **Maak middleware.ts in root**

```
TS middleware.ts > ...
1 import createMiddleware from 'next-intl/middleware';
2
3 export default createMiddleware({
4   // A list of all locales that are supported
5   locales: ['nl', 'en', 'de'],
6
7   // Default locale
8   defaultLocale: 'nl'
9 });
10
11 export const config = {
12   // Match only internationalized pathnames
13   matcher: ['/', '/(nl|en|de)/:path*']
14 };
```

- **Maak in root een folder “messages” en zet daar de talen in die je wilt ondersteunen.**



The screenshot shows the VS Code interface. On the left, the Explorer sidebar shows the project structure: .idea, .next, app, i18n, request.ts, messages, de.json, en.json, and nl.json. The 'messages' folder is expanded. On the right, the editor shows the content of 'nl.json' with the following JSON:

```
{
  "title": "Welkom bij onze Meertalige App",
  "description": "Selecteer je voorkeurstaal",
  "languages": {
    "nl": "Nederlands",
    "en": "Engels",
    "de": "Duits"
  }
}
```

- Maak in root een folder “i18n” en maak er een request.ts bestand in

```
TS request.ts x
i18n > TS request.ts > ...
1 import {getRequestConfig} from 'next-intl/server';
2
3 export default getRequestConfig(async ({locale}) => ({
4   messages: (await import(`../messages/${locale}.json`)).default,
5   locale: locale
6 }));
```

- Maak vervolgens in de app folder een nieuwe folder “[locale]” en zet hier de layout.tsx in en de page.tsx. Dit is nodig om de routing te laten werken.

```

v app \ [locale]
# global.css
layout.tsx
page.tsx

```

Pas de layout.tsx als volgt aan zodat de translatie op alle pagina's beschikbaar is

```
app > [locale] > layout.tsx > ...
1 import {NextIntlClientProvider} from 'next-intl';
2 import {notFound} from 'next/navigation';
3 import './global.css';
4
5 export function generateStaticParams() {
6   return [{locale: 'en'}, {locale: 'nl'}, {locale: 'de'}];
7 }
8
9 export default async function LocaleLayout({
10   children,
11   params: {locale}
12 }): {
13   children: React.ReactNode;
14   params: {locale: string};
15 } {
16   let messages;
17   try {
18     messages = (await import(`../../messages/${locale}.json`)).default;
19   } catch (error) {
20     notFound();
21   }
22
23   return (
24     <html lang={locale}>
25       <body>
26         <NextIntlClientProvider locale={locale} messages={messages}>
27           {children}
28         </NextIntlClientProvider>
29       </body>
30     </html>
31   );
32 }
```

Maak dan een simpele pagina om de talen uit te testen

```
'use client';

import {useTranslations} from 'next-intl';
import {usePathname} from 'next/navigation';
import NextLink from 'next/link';

export default function Home() {
  const t = useTranslations();
  const pathname = usePathname();

  return (
    <main className="min-h-screen flex flex-col items-center">
      <div className="max-w-md w-full bg-white rounded-2xl p-8">
        <h1 className="text-3xl font-bold text-center">
          {t('title')}
        </h1>
        <p className="text-gray-600 text-center mb-8">
          {t('description')}
        </p>
        <div className="flex flex-col space-y-2">
          {[ 'nl', 'en', 'de' ].map((locale) => (
            <NextLink
              key={locale}
              href={`/${locale}`}
              className="w-full py-2 px-4 text-center">
                {t(`languages.${locale}`)}
            </NextLink>
          ))}
        </div>
      </div>
    </main>
  );
}
```

- Start de applicatie met “npm run dev” en ga naar <http://localhost:3000>. Hier zal de taal nederlands zijn omdat we het als standaardtaal hebben ingesteld in middleware.ts. als je dan naar <http://localhost:3000/en> gaat, ga je zien dat de taal in het engels is. Als je een taal kiest die niet is gemaakt door ons wordt er terug geswitched naar de standaardtaal

1.4 Referenties

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

- <https://next-intl.dev/>
- <https://www.npmjs.com/package/next-intl>
- <https://www.youtube.com/watch?v=uZQ5d2bRMO4>
- <https://www.youtube.com/watch?v=I9kiZvBBIs4&t=581s>

2 Conclusie

Ik vond dit interessant, vooral omdat ik mijn eigen apps meertalig wil maken. In het begin was het wel verwarrend. Ik begon met de next-i18next package, maar dat werkte niet goed en bleek verouderd. Elke YouTube-video die ik keek voor de i18next package, gaf een andere aanpak, wat verwarrend was. Uiteindelijk ben ik overgestapt naar next-intl, en dat werkte veel beter en was ook meer up to date.