



February 2019

Building a design system with (p)react

~ Bart Waardenburg – Front-End Lead @ ANWB



Topics

1. What is a design system?
2. The process we have in place.
3. The technical choices we had to make.

Design system?

“

A design system offers a library of visual style, components, and other concerns documented and released by an individual, team or community as code and design tools so that adopting products can be more efficient and cohesive.

~ Nathan Curtis – Founder & Designer @ Eightshapes

Why use a Design System?



Better consistency

All products use the same design & development standards.



Ship faster

Teams can design & develop products more quickly.



Higher quality

Hard design & development challenges are solved together and do not need to be repeated.



The Great Divide

The list of skills required for building web applications is getting longer and longer.

Our design system

Organization structure

14

Teams

30

Front-end developers

121

Front-end applications

Poncho



What does Poncho include?



Documentation

Everything needed to explain the visual language (colors, typography, tone and voice, etc.).



Sketch Symbols

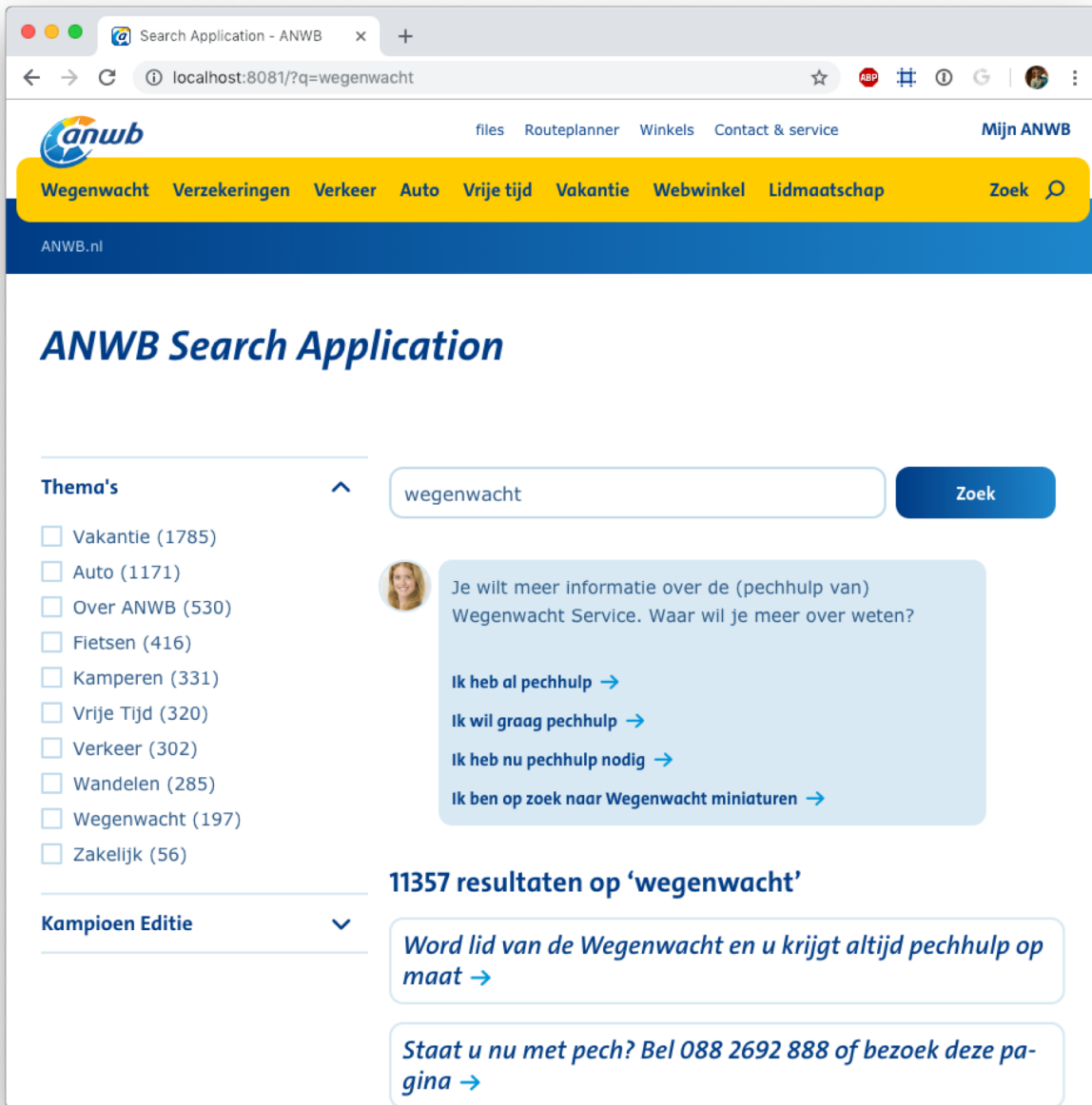
Symbols for User Experience designers to mock up applications and the rationales for using them.



Preact Components

Components build using Preact which developers can use to assemble their products.

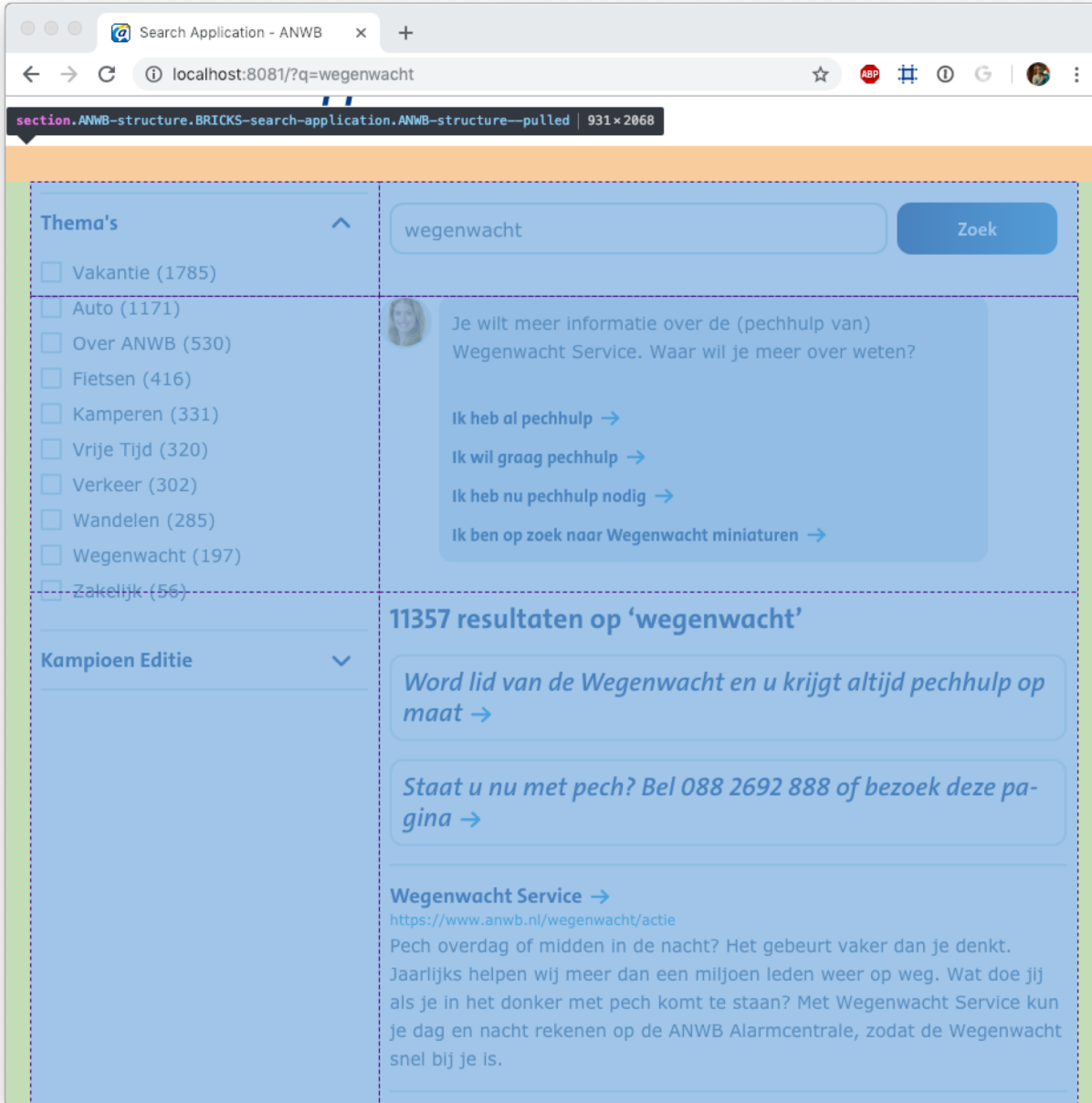
Shipping products



Search on anwb.nl

Build within only a single day, because all UI components were already there.

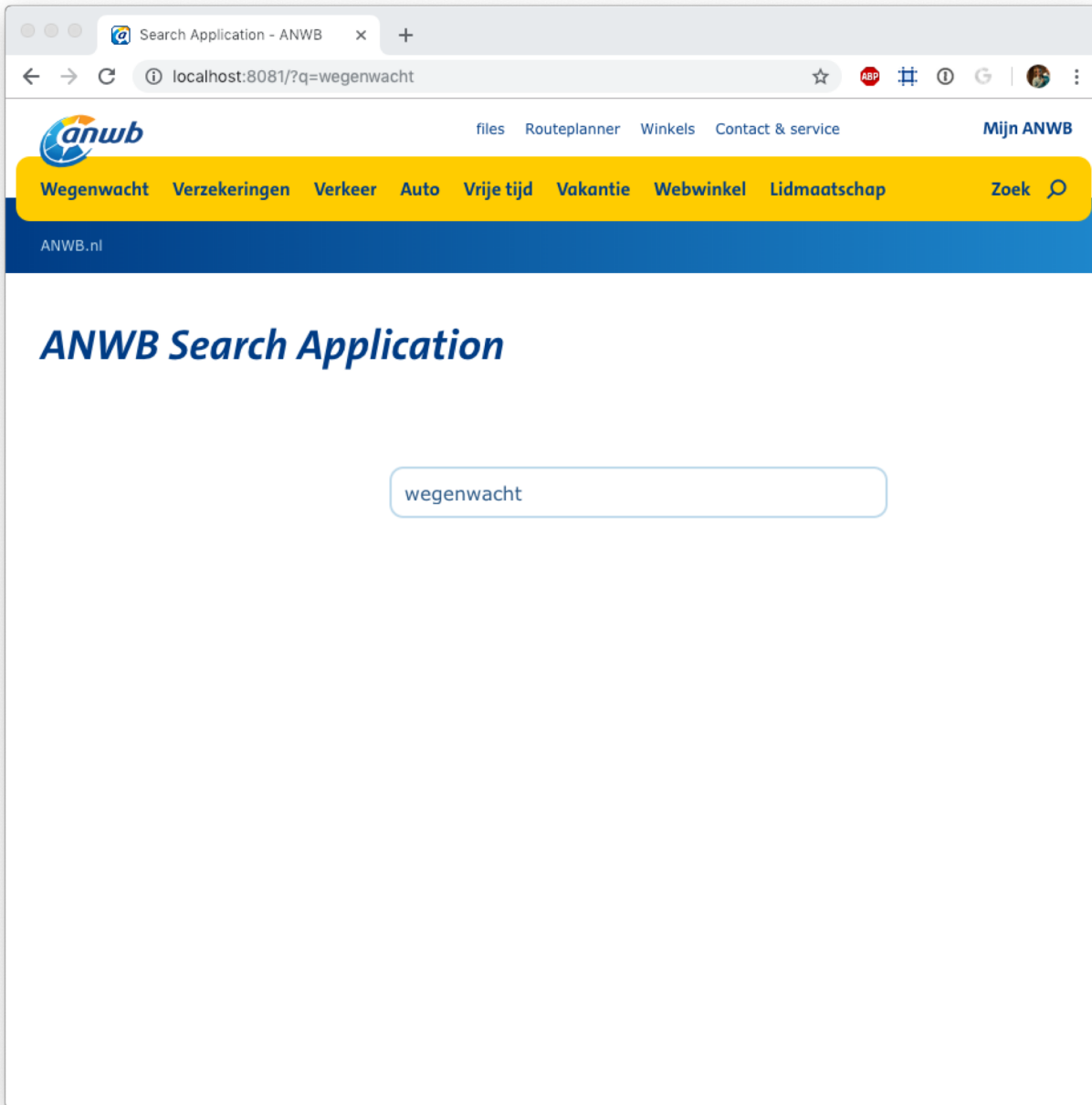




Search on anwb.nl

Build within only a single day, because all UI components were already there.





Search on anwb.nl

Build within only a single day, because all UI components were already there.



Search on anwb.nl

Build within only a single day, because all UI components were already there.

Autosuggest (optioneel)

Story Source

```
<Form/>
```

Prop Types

"FormFieldAutosuggest" Component

| property | propType | required | default | description |
|-----------|------------------------|----------|---------------|---|
| size | 's' 'm' 'l' 'xl' | - | undefined | The size of the component |
| reference | string | - | 'autosuggest' | A unique reference to the input element |
| name | string | - | 'autosuggest' | The name of the element |
| label | string | - | 'Autosuggest' | The label for the element |
| value | string | - | " | The value of the element |
| required | boolean | - | false | Whether the element is required |

Knobs | README | STORY | ACCESSIBILITY

clientId: MvVrJUf34J8A4gDJF2TgXkL6FUSv9SJ

url: /v1/content-search/suggest

value:

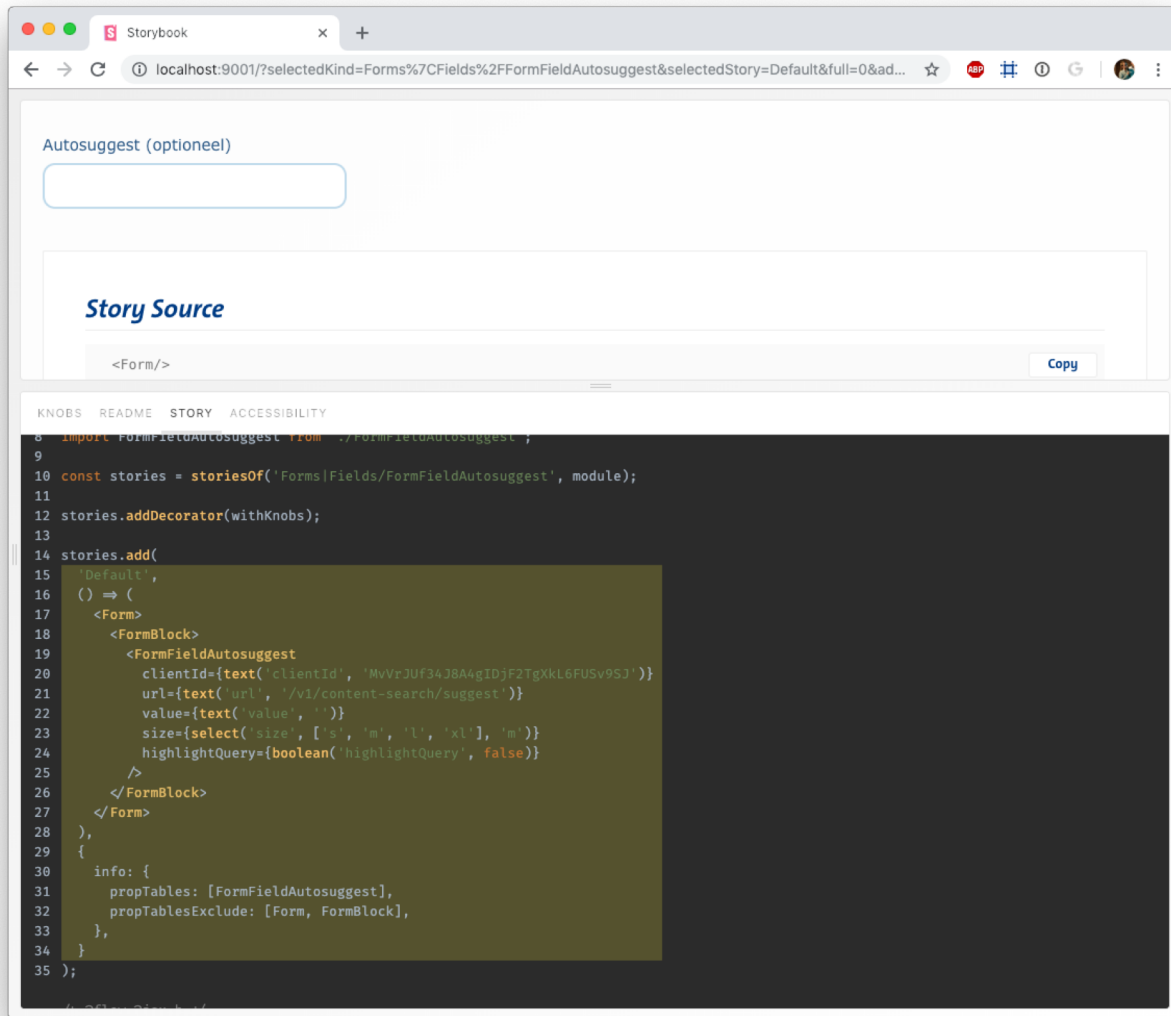
size: m

highlightQuery:

COPY | RESET

Search on *anwb.nl*

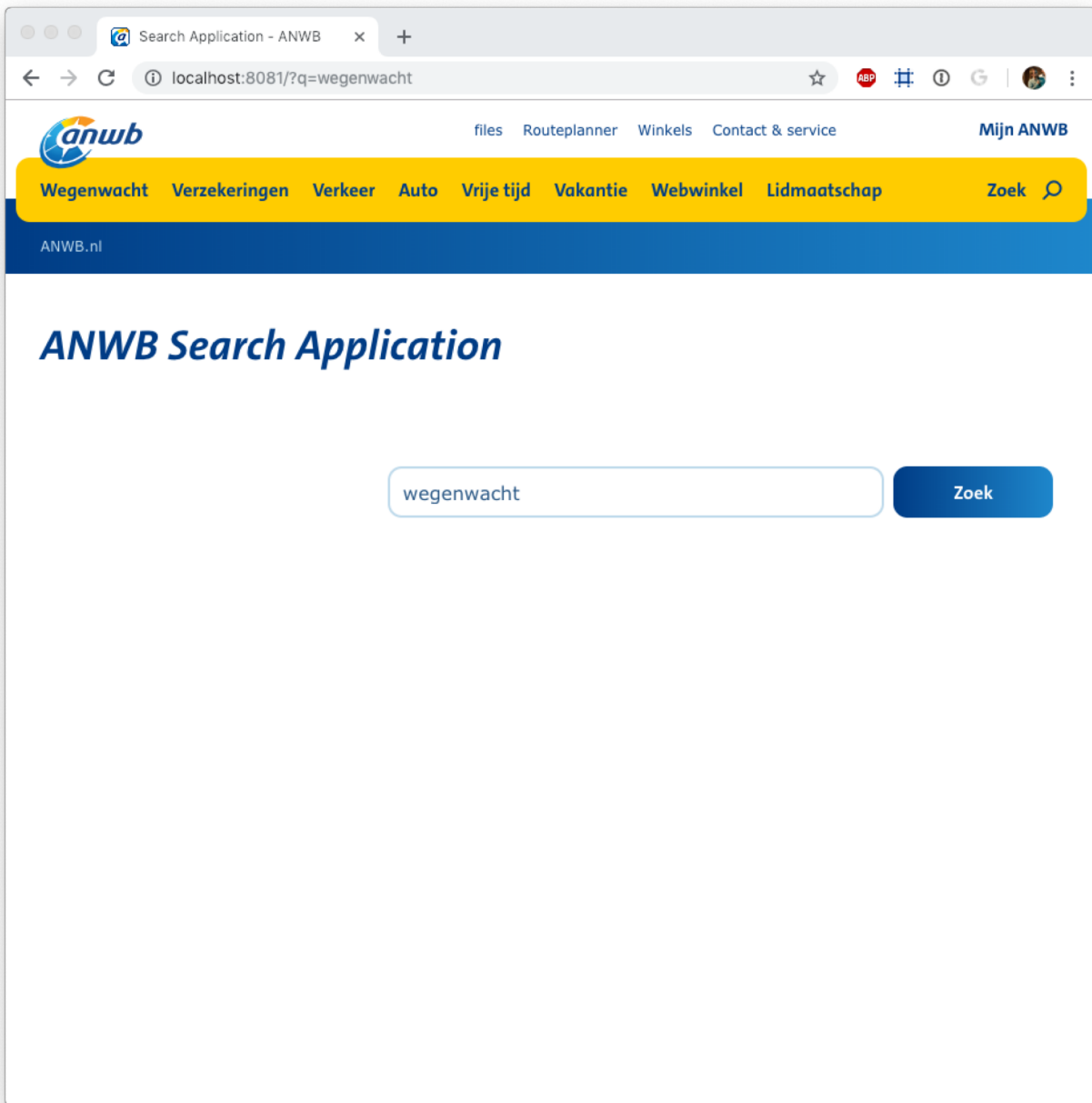
Build within only a single day, because all UI components were already there.



The screenshot shows a Storybook browser window. The top part displays a preview of the 'Autosuggest (optioneel)' component, which is a simple text input field. Below the preview is the 'Story Source' section, which contains the following code:

```
<Form/>
```

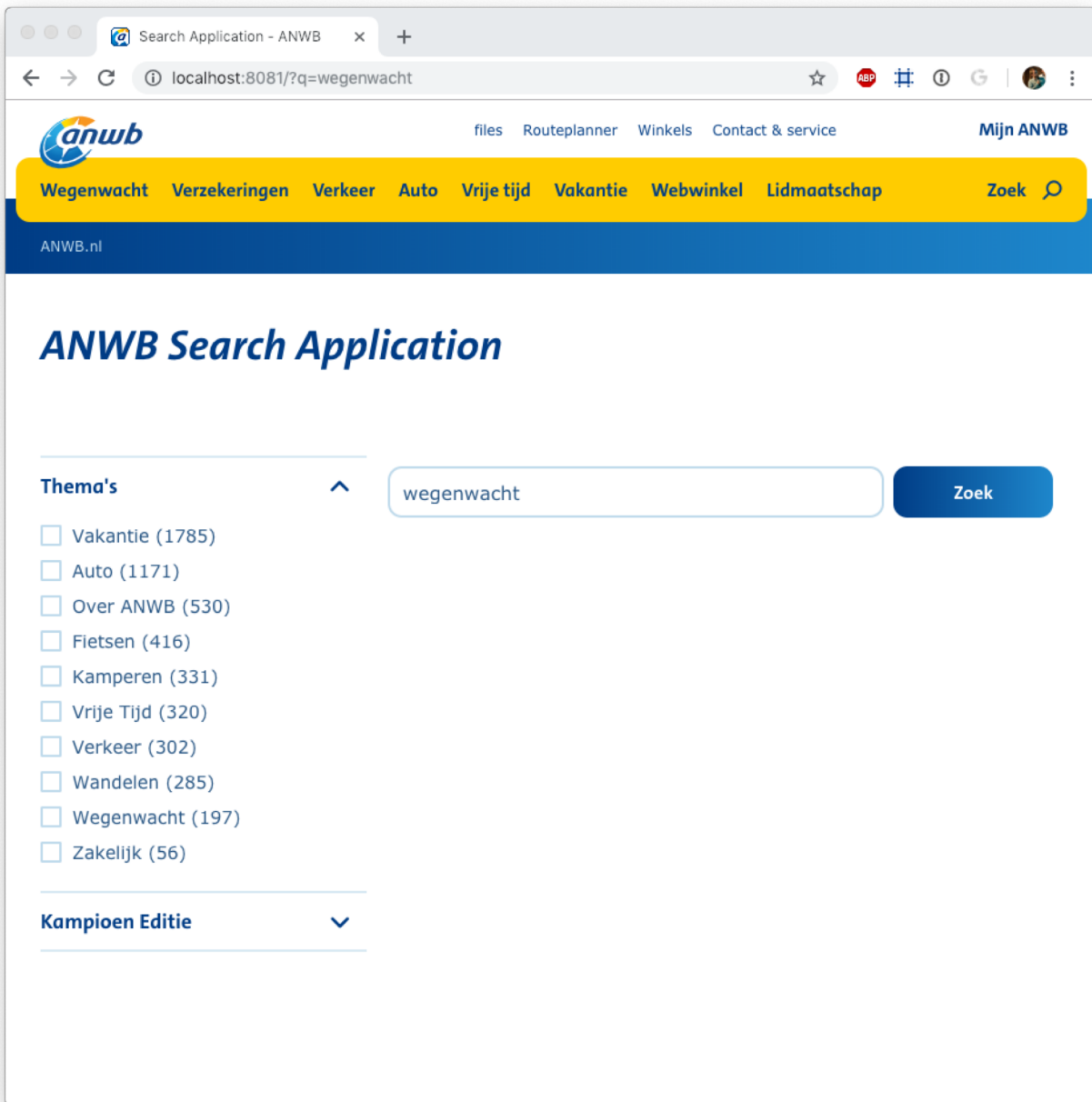
```
8 import FormFieldAutosuggest from './FormFieldAutosuggest';
9
10 const stories = storiesOf('Forms|Fields/FormFieldAutosuggest', module);
11
12 stories.addDecorator(withKnobs);
13
14 stories.add(
15   'Default',
16   () => (
17     <Form>
18       <FormBlock>
19         <FormFieldAutosuggest
20           clientId={text('clientId', 'MvVxJUF34J8A4gIDjF2TpXkL6FUSv95J')}
21           url={text('url', '/v1/content-search/suggest')}
22           value={text('value', '')}
23           size={select('size', ['s', 'm', 'l', 'xl', 'm'])}
24           highlightQuery={boolean('highlightQuery', false)}
25         />
26       </FormBlock>
27     </Form>
28   ),
29   {
30     info: {
31       propTables: [FormFieldAutosuggest],
32       propTablesExclude: [Form, FormBlock],
33     },
34   }
35 );
```

Search on anwb.nl

Build within only a single day, because all UI components were already there.

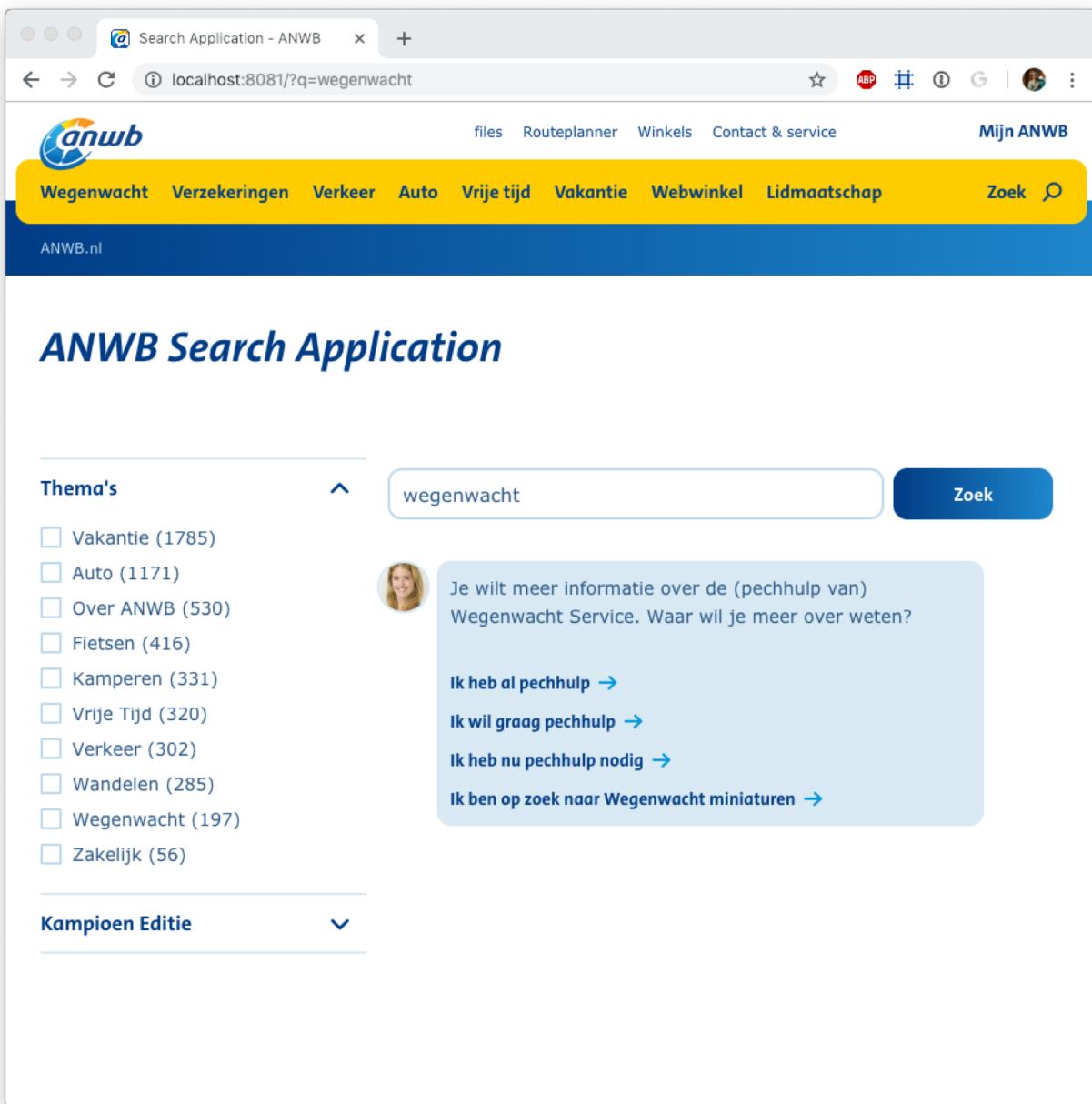




Search on anwb.nl

Build within only a single day, because all UI components were already there.





Search on anwb.nl

Build within only a single day, because all UI components were already there.



Search Application - ANWB

localhost:8081?q=wegenwacht

files Routeplanner Winkels Contact & service Mijn ANWB

Wegenwacht Verzekeringen Verkeer Auto Vrije tijd Vakantie Webwinkel Lidmaatschap Zoek

ANWB.nl

ANWB Search Application

Thema's

- Vakantie (1785)
- Auto (1171)
- Over ANWB (530)
- Fietsen (416)
- Kamperen (331)
- Vrije Tijd (320)
- Verkeer (302)
- Wandelen (285)
- Wegenwacht (197)
- Zakelijk (56)

Je wilt meer informatie over de (pechhulp van) Wegenwacht Service. Waar wil je meer over weten?

- [Ik heb al pechhulp →](#)
- [Ik wil graag pechhulp →](#)
- [Ik heb nu pechhulp nodig →](#)
- [Ik ben op zoek naar Wegenwacht miniaturen →](#)

11357 resultaten op 'wegenwacht'

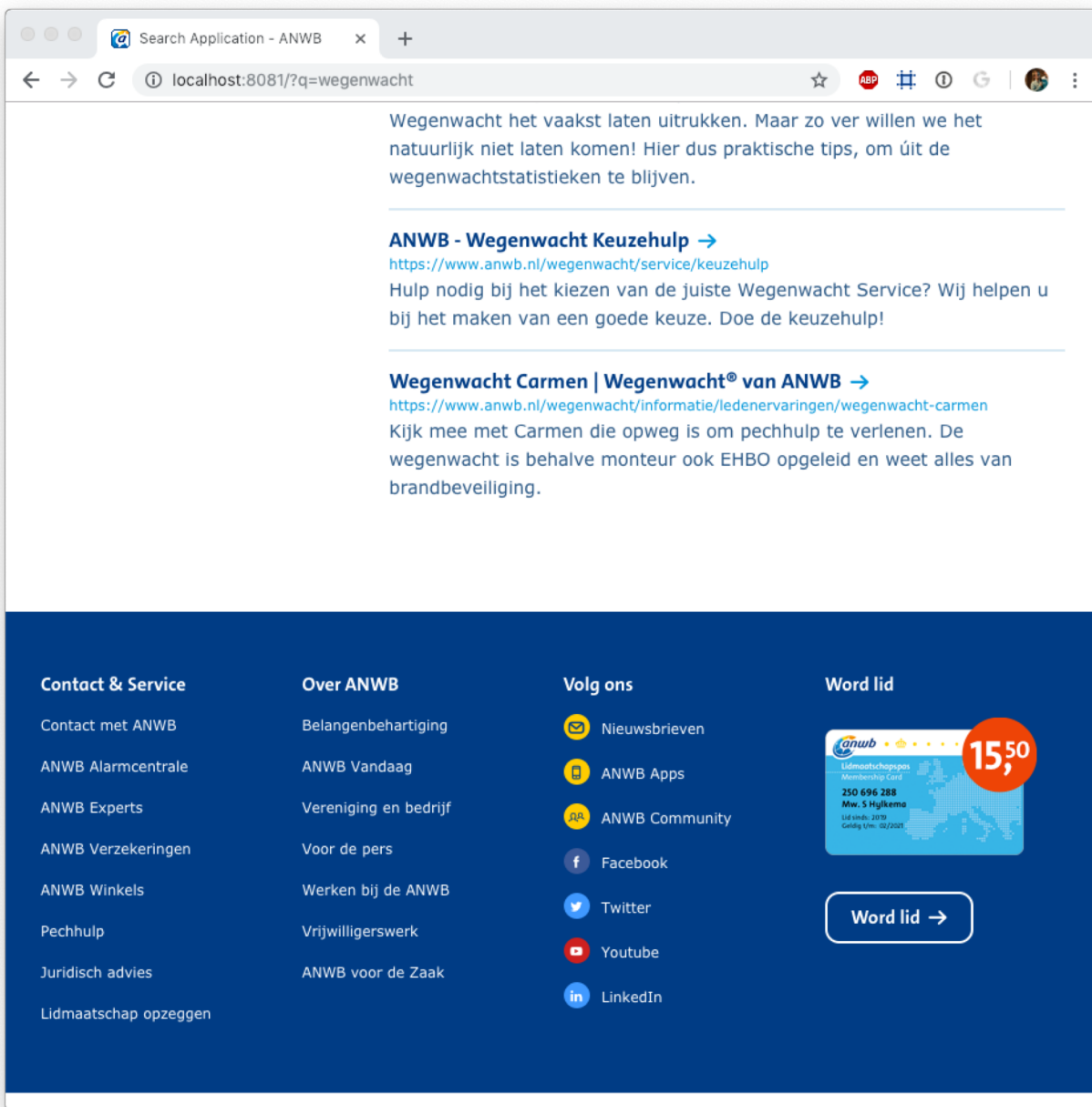
Word lid van de Wegenwacht en u krijgt altijd pechhulp op maat →

Staat u nu met pech? Bel 088 2692 888 of bezoek deze pagina →

Search on *anwb.nl*

Build within only a single day, because all UI components were already there.

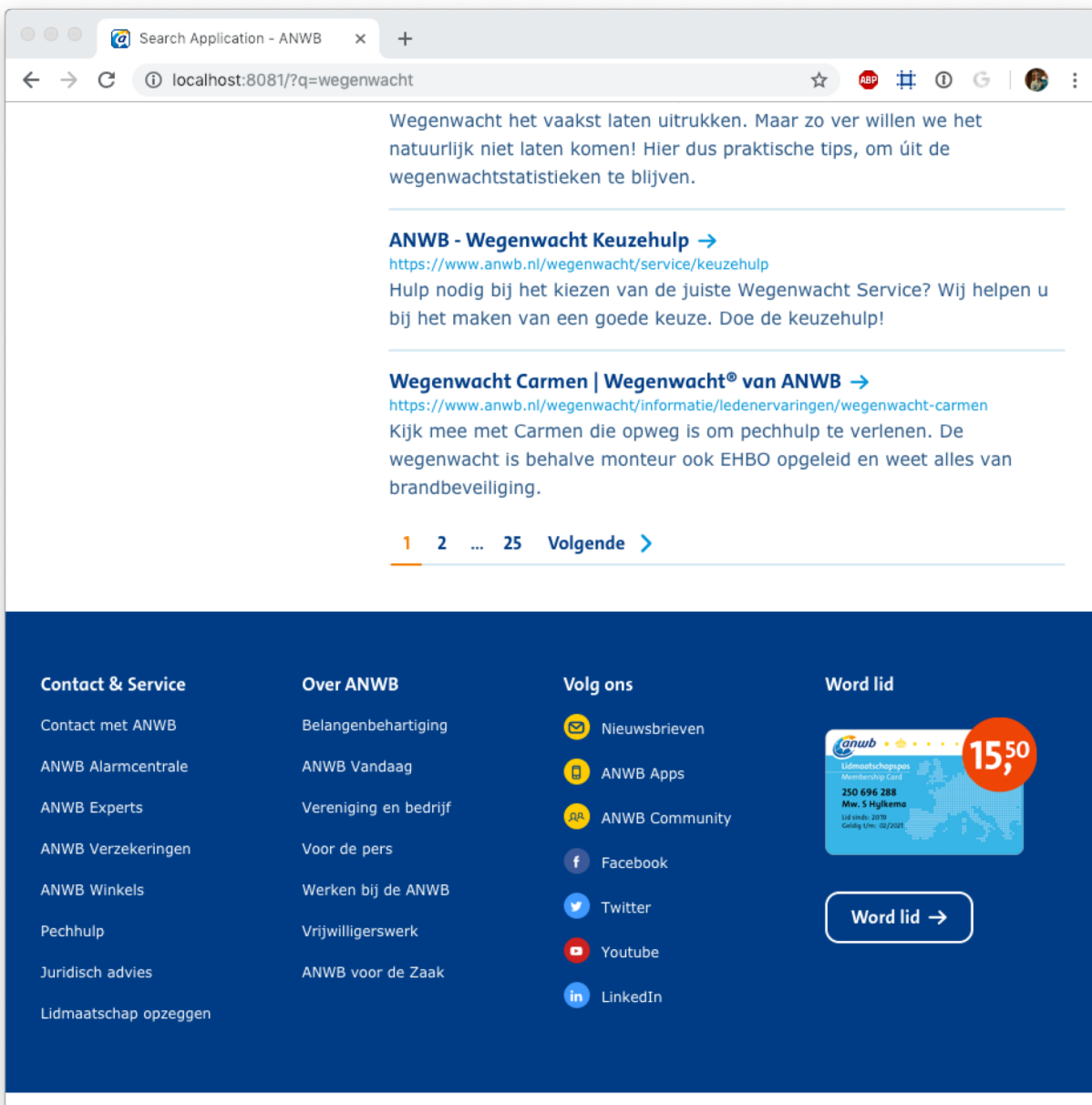




Search on *anwb.nl*

Build within only a single day, because all UI components were already there.





Search on *anwb.nl*

Build within only a single day, because all UI components were already there.



Poncho components

"@anwb/api": "^2.0.5",
"@anwb/browser": "^0.4.0",
"@anwb/button": "^2.2.0",
"@anwb/checkbox": "^1.6.0",
"@anwb/form": "^3.7.3",
"@anwb/form-block": "^1.4.2",
"@anwb/form-field-autosuggest": "^0.3.8",
"@anwb/form-field-button": "^0.1.1",

"@anwb/iris": "^2.0.7",
"@anwb/pagination": "^0.5.0",
"@anwb/panel": "^2.4.0",
"@anwb/results": "^0.6.0",
"@anwb/spinner": "^0.7.0",
"@anwb/structure": "^2.3.0",
"@anwb/tabbox": "^4.4.0",

Our process

Ownership

How do we handle ownership of Poncho



Platform team

A central team holds ownership over the platform and tooling.



Design together

All design changes are reviewed and decided on together.



Code together

Every new feature or refactor goes through a code review process.



Shared responsibility

Everyone is allowed to make changes and take ownership over components they create.



Source of truth

The design for a component is the source of truth. Initial changes are always made at the design level and synced to the component.

A design system is a product

Adaption

How did we motivate people to use Poncho



Ease of use

Using Poncho should have been easier than not using Poncho.



Dogfooding

Every team using components should also create them where needed and take ownership.



Show & Tell

Designers and developers have to show each other changed or new components every week.

Guidelines

How do we create or change Poncho components



1. Proposal

A team creates a proposal which is pitched by developers and designers within their respective peer groups.



2. Validation

The proposal is validated by doing some form of user testing either qualitative or quantitative.



3. Development

The developers of the requesting team create a merge request and start developing the component.



4. Review

The merge request is being reviewed visually and technically by the Poncho team.



5. Release

The component is being released and available for use by developers and UX designers.



6. Show & Tell

The developers and designers pitch the component within their respective peer groups.

Poncho usage

78

Components

35

Applications

Technical choices

Preact



Familiar

The React api is the most popular component model out there.



Small

Preact is smaller in bundle size compared to React and the differences were barely noticeable for developers.



Future

The future is slightly foggy when comparing the firepower of the React team with that of Preact.

Individual versioning



Flexible upgrade path

Applications should be able to mix and match their components and versions to make working with them as easy as possible.



Component freedom

Individual versioning and gradual adoption allows for more freedom when developing (breaking changes have less of an impact).



Updating

Making sure all applications are up-to-date has been proven hard.

Quality Assurance



Snapshot testing

Every component is rendered in every state so deterministic html output is ensured.



Analytics & error monitoring

We have analytics set up for component usage statistics and runtime error monitoring to detect bugs.



Screenshots

We are going to start storing visual screenshots of components to ensure visual consistency.

Specifics



CSS-in-JS

We have to keep the styling for our CMS output in sync with the output of our other applications.



Hooks

Not yet, stability and ease-of-use are highest priority.



Typescript

We are using Flow but we might switch to Flow based on overall developer usage statistics.

Thanks!

Questions?

 @bartwaardenburg