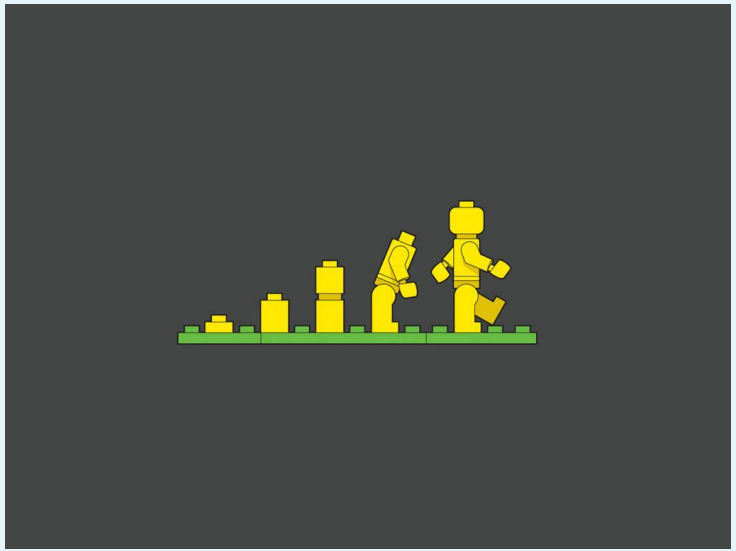
hometogo

Migrating website to react.
Feature by feature.
Zero downtime.

Irmantas Zenkus | Frontend Team Lead | Kaunas

Evolution of our frontend app



Our first Frontend app

jQuery
jQuery-ui
twig

- All content is rendered server side
- Page is reloaded on each filter change
- Some small parts are fetched via ajax and html is replaced
- Classic approach to frontend

hometogo

Why change that?

- Cannot update parts of page gradually without replacing large chunks
- One part of site cannot influence other part
- Hard to reuse some data from server side on different widgets
- Many unnecessary reloads

hometogo

MVC architecture

jQuery
rivets
twig

- Separation of concerns
- Reusable M or V or C
- A lot of different ways for developer to write code
- We have less than 30 components

hometogo

What did not work?

- A lot of different ways for developer to write code
- No control of how one component must call another
- Construction and destruction is difficult
- No lifecycle control of rivets.js view library
- Rivets.js library often forced us to choose less optimal solutions

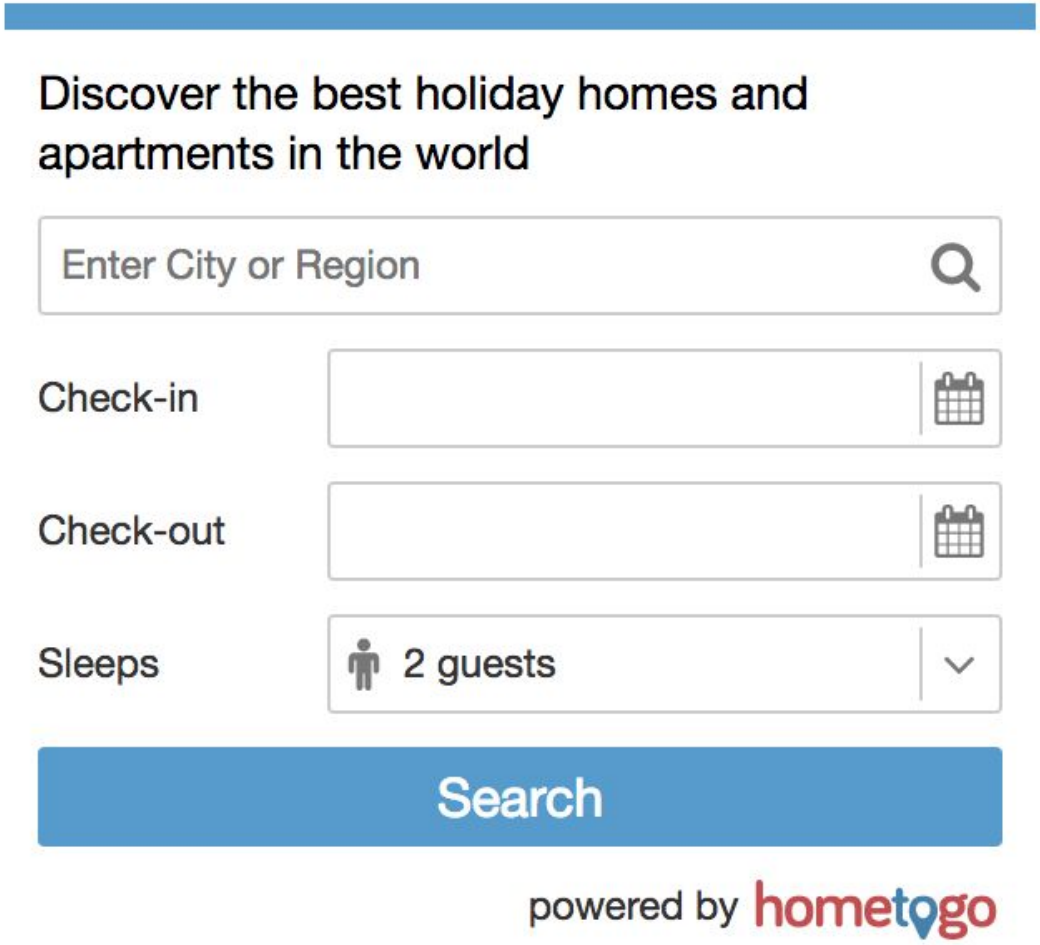
hometogo

Next architecture pilot project

- No matter how much you read and play around, you need to bring your code live at some point
- Small scale project with several difficult to do features
- Autocompleter
- Date picker
- Radio selector

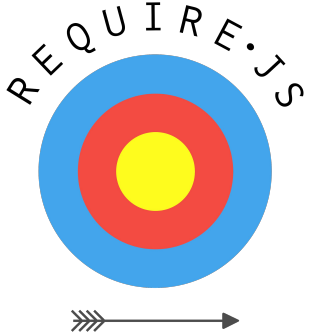
hometogo

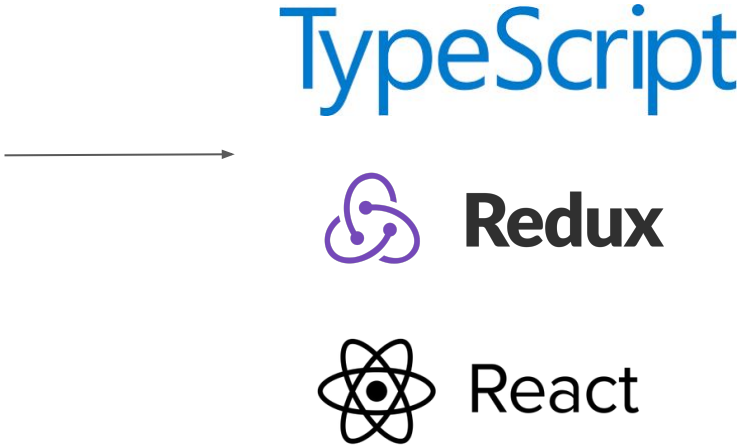
Next architecture pilot project



hometogo

Next architect ure pilot project





hometogo

Why redux?

- Popular yet do-one-thing-only library
- Easy to follow rules that have solid arguments
- Covers all cases we could think of
- Less different ways to write code

hometogo

Why react?

- Does only one thing and does it really well
- Has an API that is not likely to change
- Is used live by large enterprises
- Is backed by large enterprice

hometogo

Why typescript?

- No code comments are needed
- Can be maintained by a large group of developers
- Is backed by large enterprice

hometogo

Road to React

- Start with modules that are self sufficient
- Pick the one that have the least dependencies
- If contract with rest of app is needed - use event manager
- Install performance and key business measurment code

hometogo

Road to React

Islands of React

```
import { render } from 'react-dom';
import { Provider } from 'react-redux';
import store from './store';

render((
   <Provider store={store} >
       <BreadCrumb />
   </Provide>
), document.getElementById('breadcrumb'));

render((
   <Provider store={store} >
       <CurrencyChooser />
   </Provide>
), document.getElementById('currency-chooser'));

render((
   <Provider store={store} >
       <SearchSummary />
   </Provide>
), document.getElementById('search-summary'));
```

hometogo

Road to React

Zero downtime
==
No decrease
of velocity

- Convert a module to redux only when changes for it are planned
- Make converted code as part of new feature being tested
- Run code to gather enought data
- Clean old code

hometogo

Road to React

Minimal cost

- Islands of converted components formed
- Those are merged to form even bigger islands
- There is already a solid foundation of redux+react code
- Price for bussiness is minimal, since changes were planned anyway

hometogo

Road to React

Islands become continents

```
import { render } from 'react-dom';
import { Provider } from 'react-redux';
import store from './store';

render((
   <Provider store={store} >
       <TopBar>
           <BreadCrumb />
           <CurrencyChooser />
           <SearchSummary />
       </TopBar>
   </Provide>
), document.getElementById('top-bar'));
```

hometogo

Road to React

Challenges

- Gained trust and support from business developers
- “Install React” became business value
- Start working on Offers section and Map

hometogo

Road to React

Wyndham Grand Clearwater Beach

★★★★ Hotel, Clearwater Beach, Clearwater, Florida

4.3 "Outstanding"

930 reviews

from 152 €

AMOMA	152 €
Hotelpower.com	156 €
bedfinder	203 €

Details ⌄ | 24 Offers ⌄ | View deal ›

Hilton Marco Island Beach Resort

★★★★ Hotel, City of Marco, Marco Island, Florida

4.2 "Outstanding"

3,064 reviews

-9% ~~179 €~~

from 162 €

Hilton.com	162 €
Hilton.com	162 €
Agoda.com	179 €

Details ⌄ | 105 Offers ⌄ | View deal ›

One Ocean Resort & Spa

★★★★ Hotel, Atlantic Beaches, Neptune Beach, …

4.3 "Outstanding"

1,157 reviews

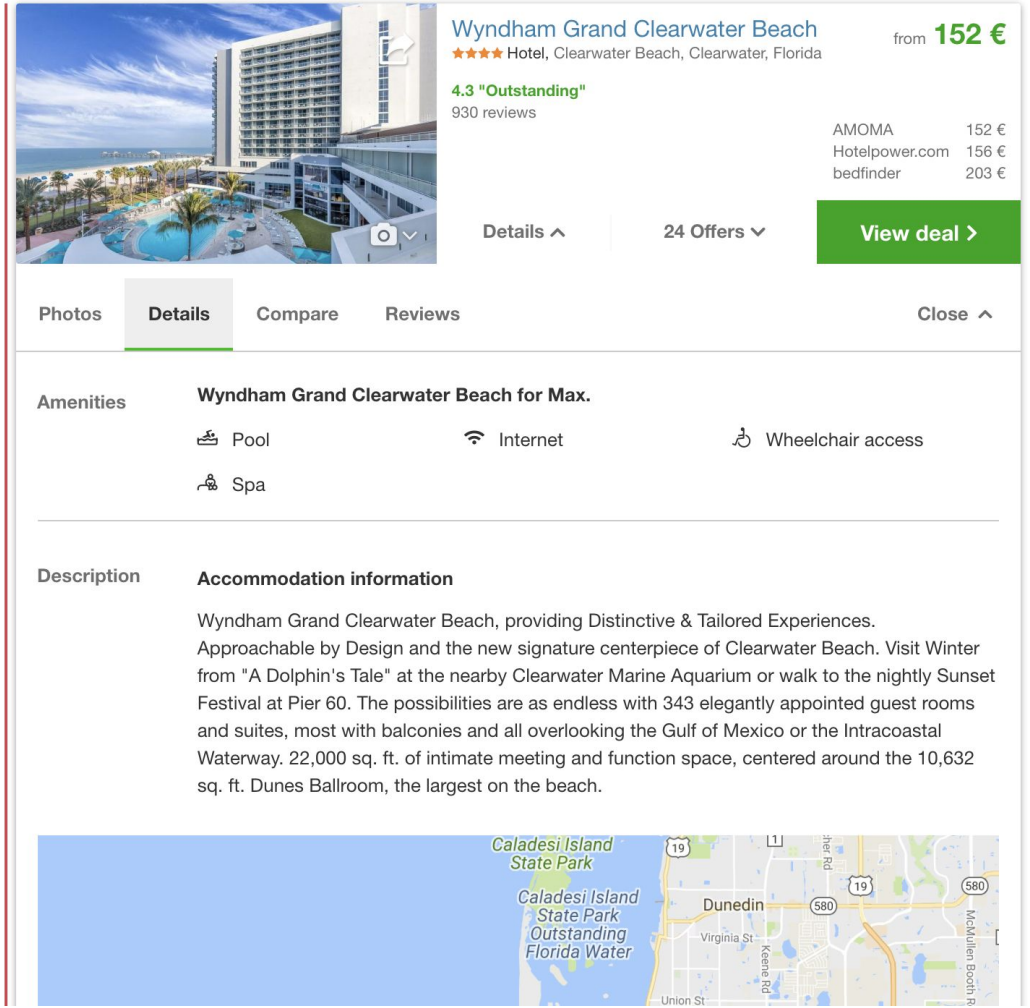
from 182 €

Hotels.com	182 €
Orbitz.com	182 €
Travelocity	182 €
Hotwire.com	182 €

hometogo

Road to React

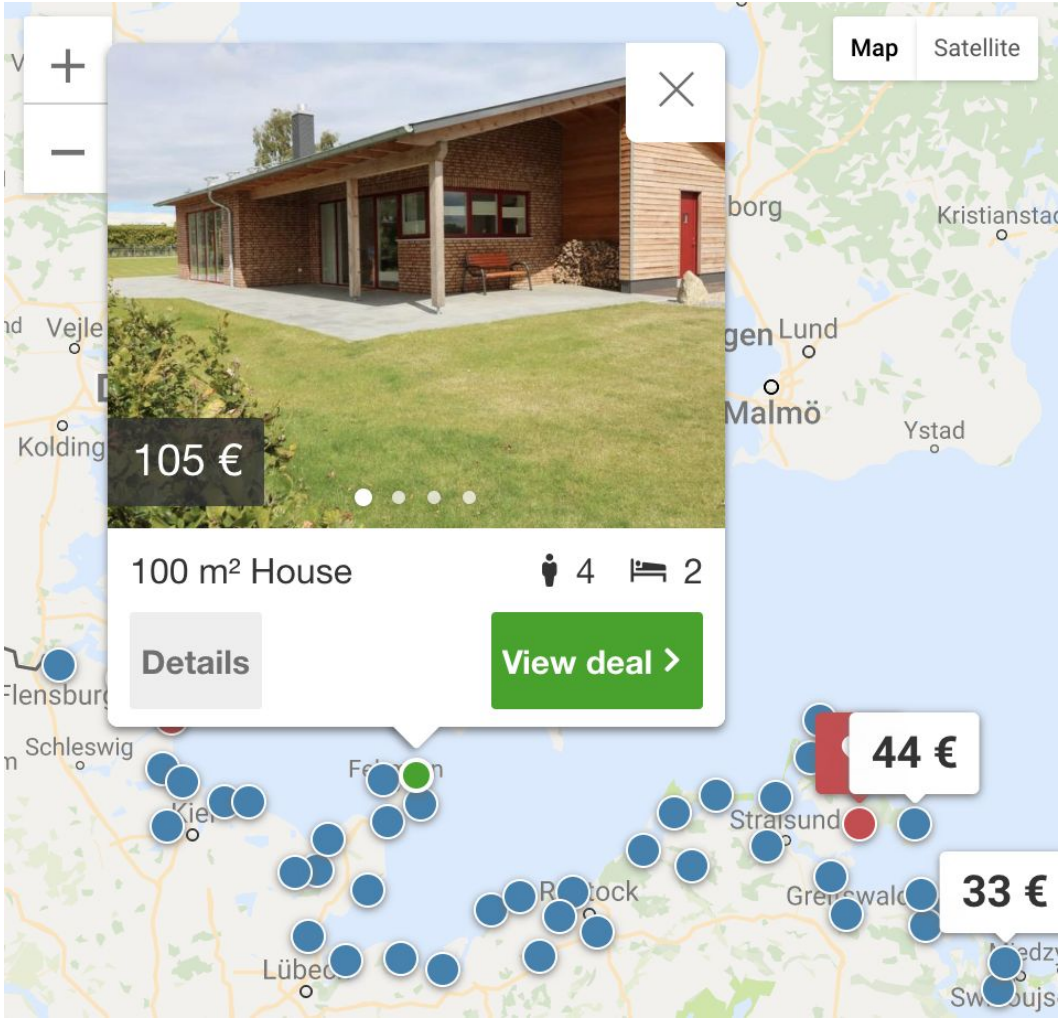
Offer cards



hometogo

Road to React

Map



hometogo

Road to React

Declarative map

```
// google maps api
onMapReady(map) {
   map.addMarker(new MarkerOptions().position(new LatLng(10, 10));
}

// react-google-maps
<Map>
   <Marker key={1} />
   <Marker key={2} />
</Map>
```

hometogo

Road to React

SPA must have a router

- You want all dynamic content to be wrapped with a Router
- In our case we needed all page to be wrapped
- Search crawlers are not to trust to see your React rendered content

hometogo

Road to React

Static content

```
class HtmlWrapper extends React.Component<Props, {}> {
   render() {
       const TagName = this.props.tagName;
       return (
           <TagName
               id={this.props.id}
               dangerouslySetInnerHTML={{ __html:
this.props.staticHTML }}
           />
       );
   }
}
```

hometogo

What is better?

- Whole application state is visible in one place
- At any given time there is just the minimal amount of DOM
- At any change of application state there is a minimal amount of computations and DOM updates
- Our RWD is mostly done with JS instead of CSS - more flexible
- Application has a solid structure and all developers know exactly where to find what

hometogo

What could be better?

- Move as much computation as possible from reducer to thunks
- Have as little middlewares as you can
- Write tests for thunks only
- Separate components from containers
- Join several entry points into one app

hometogo

Improve

Reducer with logic

```
// reducer
handleActions({
   [RECEIVE_DATA]: (state, action) => {
       const items = action.payload.response.map((item) => {
           return {
               id: item.itemId,
               name: item.title
           };
       });

       return items;
   }
}, defaultState);
```

hometogo

Improve

Logic within thunk

```
//with thunk
const receiveData = (response) => (action, getState) => {
   const items = action.payload.response.map((item) => {
       return { id: item.itemId, name: item.title };
   });

   dispatch(setItems({ items }));
}

//reducer
handleActions({
   [SET_ITEMS]: (state, action) => {
       return action.payload.items;
   }
}, defaultState);
```

hometogo

What could be better?

- Move as much computation as possible from reducer to thunks
- Have as little middlewares as you can
- Write tests for thunks only
- Separate components from containers
- Join several entry points into one app

hometogo

Thanks