React

dinsdag 8 oktober 2019 13:17

29

Return React.createElement('div',null,'h1','Hi, I\'m a React App')
Configuration for the div = second element
Childeren in div = third elemnt h1 zal al tekst tonen voor een tag H1 moet je een nieuwe element createn

Voor css kan je het in null => {className: 'App'}
This is the reason that we use JSX (does like html)

30
In html is het class maar bij jsx is het className
Het moet 1 root element hebben in return dus <h1></h1> zal niet werken

src > JS App.js > ...

```
1   import React, { Component } from 'react';
2   import './App.css';
3   import Person from './Person/Person.js';
4
5
6   class App extends Component {
7     render() {
8       return (
9         <div className="App">
10          <h1>Hi, I'am a React App</h1>
11          <button>Verander naam</button>
12          <Person name="Cansu" age="21">Test childeren</Person>
13        </div>
14      );
15    }
16  }
17
18  export default App;
```

```
18   export default App;
```

```
import React from 'react';


//function
const person = (props) => {
    //return <p>I'm a person and I am {Math.floor(Math.random()*30)} y
    return (
        <div>
            <p>I'm {props.name} and I am {props.age} years old!</p>
            <p>{props.children}</p>
        </div>
    )

};

export default person;
```

2)

```
src > JS App.js > App > render
1   import React, { Component } from 'react';
2   import './App.css';
3   import Person from './Person/Person.js';
4
5   class App extends Component {
6     state = {
7       persons : [
8         { name:'Cansu', age:21 },
9         { name : 'Test', age:23 },
10      ],
11    }
```

```
ears old!</p>
```

```
```

```
  switchName = () => {
    //console.log('test')
    this.setState({
      persons : [
        { name:'Cansu', age:21 },
        { name : 'Test', age:30 },
      ],
    })
  }
  render() {
    return (
      <div className="App">
        <h1>Hi, I'am a React App</h1>

        {/* only pas the reference this.switchName not this.swi
        <button onClick={this.switchName}>Verander naam</button
        <Person name={this.state.persons[0].name} age={this.sta
        <Person name={this.state.persons[1].name} age={this.sta

      </div>
    );
  }
}
export default App;
```

src > Person > JS Person.js > ...

```
import React from 'react';


//function
const person = (props) => {
    //return <p>I'm a person and I am {Math.floor(Math.random()*
    return (
        <div>
            <p>I'm {props.name} and I am {props.age} years old!<
            <p>{props.children}</p>
        </div>
    )
```

```
tchName() */}
>
te.persons[0].age}>Test childeren</Person>
te.persons[1].age}>Test childeren</Person>
```

```
30)} years old!</p>

/p>
```

```
13
14    };
15
16    export default person;
```

3) 43 dia is een moeilijker sectie 3

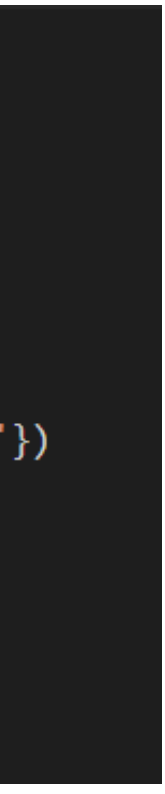
44: Statefull component is app (smart component or container comp)
More functional

Stateless, presentational, dom component is person because he has not interenal management

45: Bind oef

46: Tekstvakken inputvelden doorgeven

```
const app =(props) => {
  const [personsState, setPersonsState ] = useState( {
    persons : [
      { name:'Cansu', age:21 },
      { name : 'Test', age:23 },
    ],
  });

  const [otherState, setOtherState]= useState({otherState: 'OtherState

  console.log(personsState, otherState);

  const changeName = (event) => {
    setPersonsState({
      persons : [
```



```
        { name: 'Cansu', age:21 },
        { name : event.target.value, age:30 },
      ],
    })
  }


  return (
    <div className="App">
      <h1>Hi, I'am a React App</h1>
      <Person
        name={personsState.persons[0].name}
        age={personsState.persons[0].age}>Test childeren</Person>
      <Person
      name={personsState.persons[1].name}
       age={personsState.persons[1].age}
       changed={changeName}>Test childeren</Person>
    </div>
  );
}
export default app;
```

erson > JS Person.js > ...

```
import React from 'react';


//function
const person = (props) => {
    //return <p>I'm a person and I am {Math.floor(Math.random()*30)}
    return (
        <div>
            <p onClick={props.click}>I'm {props.name} and I am {props
            <p>{props.children}</p>
            <input type="text" onChange={props.changed} value={props.
        </div>
    )

};
```

```
years old!</p>


.age} years old!</p>

name}/>
```

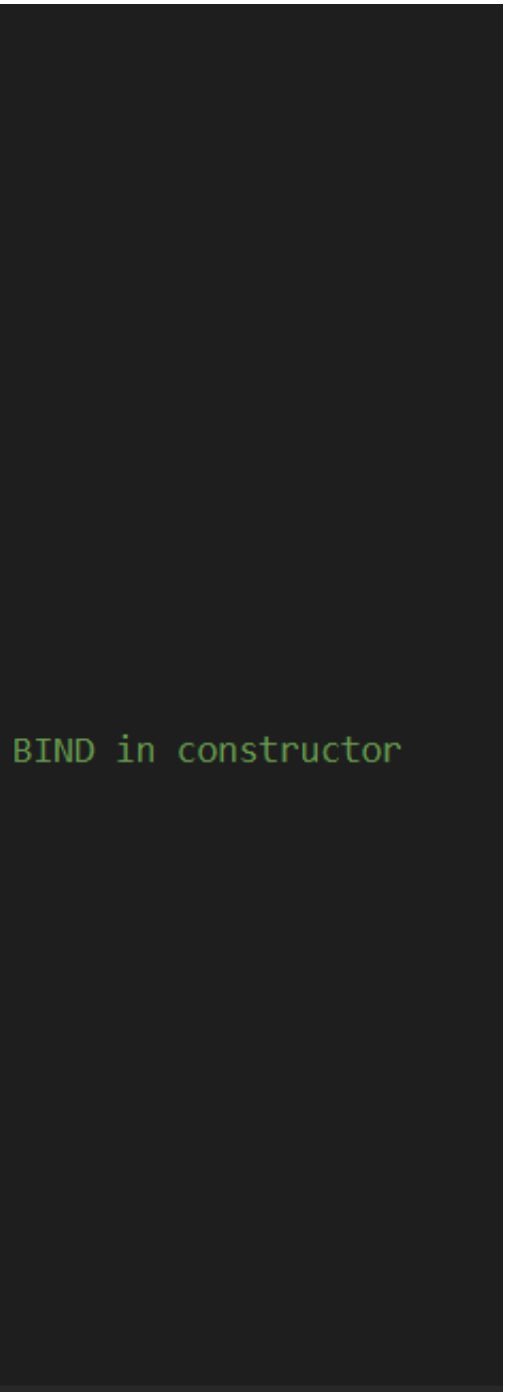
```
export default person;
```

babel src/app.js --out-file=public/scripts/app.js --presets=env,react --watch

Live-server public

Sectie6: laatste stap E6 class prop

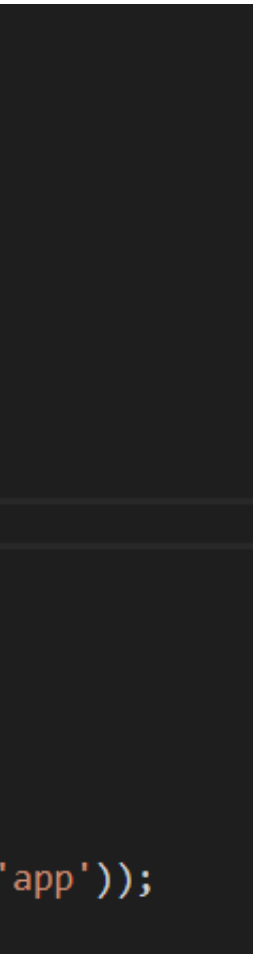
```
class OldSytax{
    constructor(){
        this.name = "Mike";
        //DIT IS DE OPLOSSING:
         this.getGreeting = this.getGreeting.bind(this);
    }

    getGreeting(){
        return `Hi my name is ${this.name}`
    }
}

const oldSytax = new OldSytax();
const getGreeting = oldSytax.getGreeting;
console.log(getGreeting()); // GEEFT EEN FOUT WE LOSSEN DIT OP MET EEN


class NewSyntax{
    name='Jen';
    getGreeting = () =>{
        return `Hi my name is ${this.name}`
    }
}

const newSyntax = new NewSyntax();
const newGetGreeting = newSyntax.getGreeting;
console.log(newGetGreeting());
```


BIND in constructor

Sectie7

```
const Layout = (props) =>{
    return(
        <div>
            <p>Header</p>
            {props.content}
            <p>Footer</p>
        </div>
    );
};

const template = (
    <div>
        <p>Page Title</p>
        <p>This is my page</p>
    </div>
);
ReactDOM.render(<Layout content={template}/>,document.getElementById(
```

```
const Layout = (props) =>{
    return(
        <div>
            <p>Header</p>
            {props.children}
            <p>Footer</p>
        </div>
    );
};

const template = (
    <div>
        <p>Page Title</p>
        <p>This is my page</p>
    </div>
```


'app'));

```
);

ReactDOM.render((
    <Layout>
    <p>Test inline</p>
    </Layout>
),document.getElementById('app'));
```

Sectie9

```
const NotFoundPage = () =>(
    <div>
        404! <Link to="/">Go home</Link>
    </div>
)

const Header = () => (
    <header>
        <h1>Expensify</h1>
        <NavLink to="/" activeClassName="is-active" exact>Home</NavLi
        <NavLink to="/create" activeClassName="is-active">Create</Nav
        <NavLink to="/edit" activeClassName="is-active">Edit</NavLink
        <NavLink to="/help" activeClassName="is-active">Help</NavLink
    </header>
)



const routes = (
    <BrowserRouter>
    <div>
        <Header/>
        <Switch>
            <Route path="/" component={ExpenseDashboardPage} exact />
            <Route path="/create" component={Create}   />
            <Route path="/edit" component={Edit}  />
            <Route path="/help" component={Help}  />
```

nk>
link>

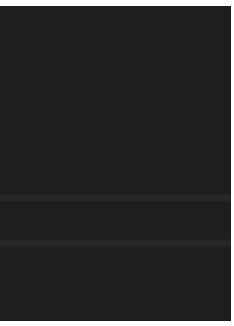
```
            <Route component={NotFoundPage}/>
        </Switch>
    </div>
    </BrowserRouter>
);
ReactDOM.render(routes ,document.getElementById('app'));
```

Setcie 10 redux

```
import { createStore} from 'redux';

const store = createStore((state = { count: 0}, action)=>{
    if(action.type ==="INCREMENT"){
        return {
            count: state.count +1
        };
    } else {
        return state;
    }
});

console.log(store.getState());


//dispatch zal store twee keer uitvoeren
store.dispatch({
    type: 'INCREMENT'
});

store.dispatch({
    type: 'INCREMENT'
});

console.log(store.getState());
```

Object { count: 0 }

Object { count: 2 }



```
import { createStore} from 'redux';

const store = createStore((state = { count: 0}, action)=>{
    switch(action.type){
        case 'INCREMENT':
            return{
                count:state.count +1
            };
        case 'DECREMENT':
            return{
                count: state.count -1
            };
        default:
            return state;
    }
});

console.log(store.getState());


//dispatch zal store twee keer uitvoeren
store.dispatch({
    type: 'INCREMENT'
});

store.dispatch({
    type: 'DECREMENT'
});

store.dispatch({
    type: 'DECREMENT'
});

console.log(store.getState());
```

Object { count: 0 }
redux-101.js:18
Object { count: -1 }

Object { count: -1 }

```
//wordt altijd geroepen als het state veranderd
const unsubscribe = store.subscribe(() => {
    console.log(store.getState());

});


//ACTIONS
//dispatch zal store twee keer uitvoeren
store.dispatch({
    type: 'INCREMENT'
});

unsubscribe();

store.dispatch({
    type: 'DECREMENT'
});

store.dispatch({
    type: 'RESET'
});

store.dispatch({
    type: 'DECREMENT'
});
```

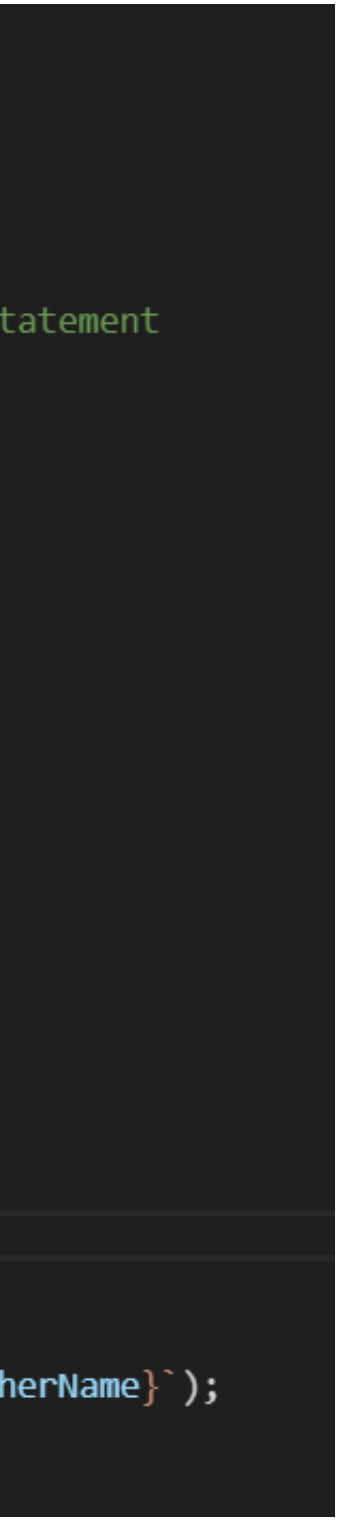
Zal enkel eerste tonen

Descrutering

```
const person = {
    name: 'Cansu',
    age: 21,
```

```
    location:{
        city:'Gent',
        temp:2
    }
};

const { name: firstname = 'Anonymous', age} = person; //destructered s
console.log(`${firstname} is ${age}`);

const { city, temp: temperature} = person.location
if(city && temperature){
    console.log(`${city} is ${temperature} graden`);
}


const book ={
    title: 'Boek1',
    author:'Een auteur',
    publisher:{
        name: 'Maan'
    }
}

const {title, author, publisher} = book;
const {name: publisherName='Zon'} = book.publisher;

if(publisherName){
    console.log(`Title: ${title} Author: ${author} Publisher: ${publis
}
```

```
//ARRAY DESTRUCTER//
const address = ['desmetstraat','Gent','9000'];
const [street, , zip= '0000'] = address;
console.log(`Odisee: ${street} ${zip}`);
```


tatement
herName}`);

Action genarators

```
//action generator

const incrementCount = ({ incrementBy = 1} = {}) =>({
    type:'INCREMENT',
    incrementBy: incrementBy
})

const decrementCount = ({decrementBy = 1} = {} ) => {
    return{
        type: 'DECREMENT',
        decrementBy: decrementBy
    }
}

const resetCount = () =>({
    type: 'RESET',
})

const setCount = ({set} = {}) =>({
    type: 'SET',
    count: set
})
```

```
const store = createStore((state = { count: 0}, action)=>{
   switch(action.type){
      case 'INCREMENT':
         return{
            count:state.count + action.incrementBy
         };
      case 'DECREMENT':
         return{
            count: state.count - action.decrementBy
         };
      case 'SET':
          return{
             count: action.count
          }
      case 'RESET':
```

```
            return{
                count: 0
            };
        default:
            return state;
    }
});

//wordt altijd geroepen als het state veranderd
const unsubscribe = store.subscribe(() => {
    console.log(store.getState());

});


store.dispatch(incrementCount({ incrementBy: 5 }));
store.dispatch(incrementCount());
store.dispatch(resetCount());
store.dispatch(decrementCount());
store.dispatch(decrementCount({decrementBy:10}));
store.dispatch(setCount({set:101}));
```

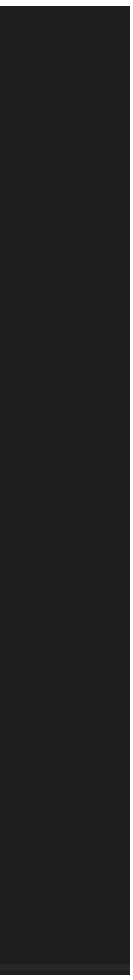
Reducers

```
//Reducers
//1. pure functions: output wordt alleen bepaald door de output
//2. Never change state or action

const countReducer = ((state = { count: 0}, action)=>{
    switch(action.type){
        case 'INCREMENT':
            return{
                count:state.count + action.incrementBy
            };
```

```
        case 'DECREMENT':
            return{
                count: state.count - action.decrementBy
            };
        case 'SET':
            return{
                count: action.count
            }
        case 'RESET':
            return{
                count: 0
            };
        default:
            return state;
    }
});


const store = createStore(countReducer);
```

100.

```
//render hijacking
//prop manipulation
//abstract state
import React from 'react';
import ReactDOM from 'react-dom';

const Info = (props) => (
    <div>
        <h1>The info</h1>
        <p>The info is {props.info}</p>
    </div>
)

const withAdminWarning = (WrappedComponent) => {
    return (props) =>(
        <div>
            {props.isAdmin && <p>This is private info, please don't sh
            <WrappedComponent {...props}/>
```



nare</p>}

```
        </div>
    )
};

const requireAuthentication = (WrappedComponent) =>{
    return (props) =>(
        <div>
            {props.IsAuthenticated ?  <WrappedComponent {...props}/>
        </div>
    )
}

const AdminInfo = withAdminWarning(Info);
const AuthInfo = requireAuthentication(Info);
ReactDOM.render(<AuthInfo IsAuthenticated={true} />, document.getEleme
```

```
FAIL  src\tests\components\ExpenseListFilters.test.js
● hould handle date focus changes

  Method “props” is only meant to be run on a single node. 0 found instead.

    at ShallowWrapper.single (node_modules/enzyme/build/ShallowWrapper.js:1
    at ShallowWrapper.props (node_modules/enzyme/build/ShallowWrapper.js:94
    at ShallowWrapper.prop (node_modules/enzyme/build/ShallowWrapper.js:115
    at Object.<anonymous> (src/tests/components/ExpenseListFilters.test.js:
        at new Promise (<anonymous>)
    at node_modules/p-map/index.js:46:16
    at processTicksAndRejections (internal/process/task_queues.js:93:5)

√ should render ExpenseListFilters with alt data correctly (2ms)
√ should handle text change (2ms)
√ should sort by date (1ms)
√ should sort by amount (1ms)
√ should handle date changes (1ms)
× hould handle date focus changes
```

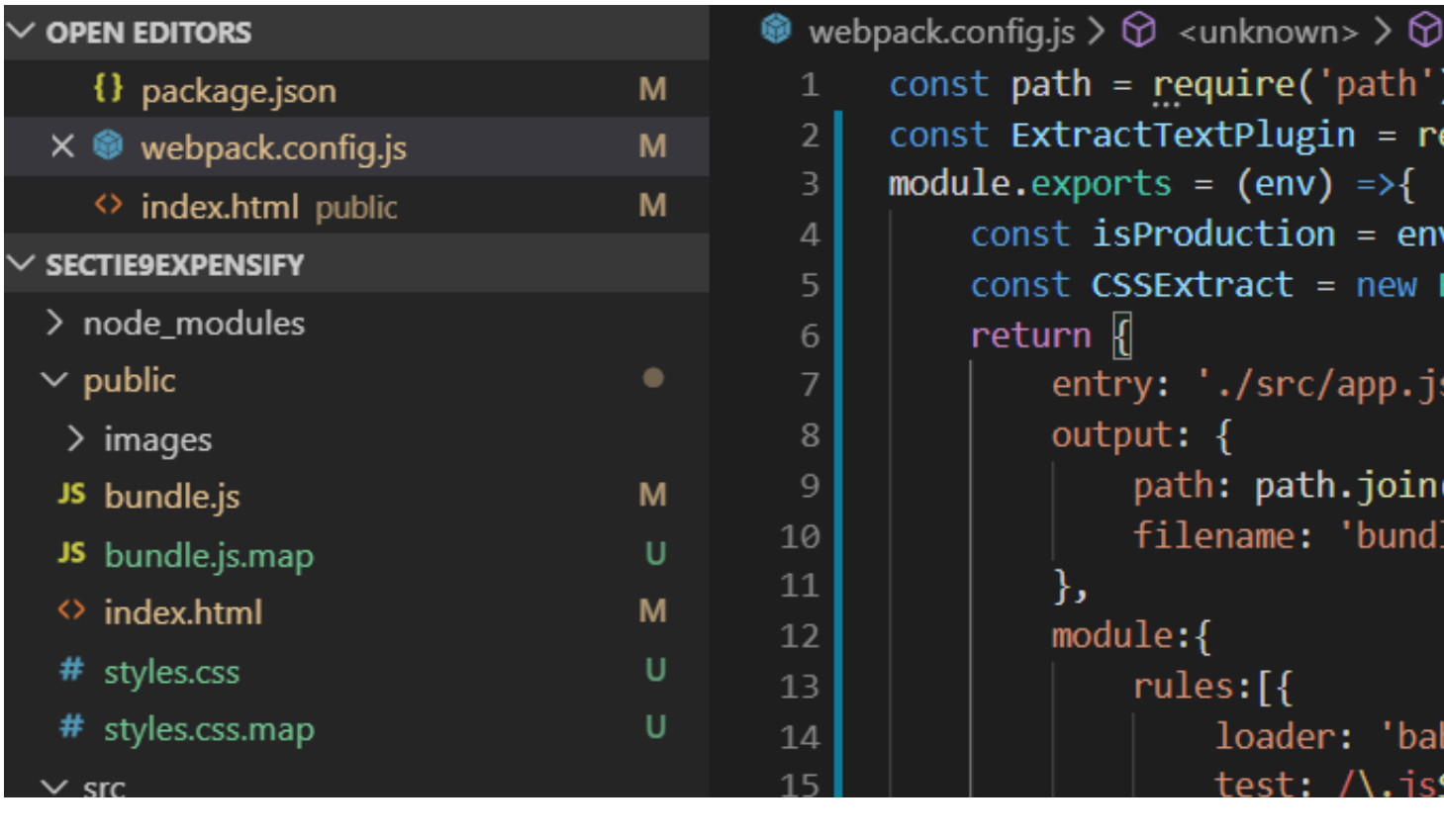
```
: <p>Please login to view the info</p>}
```

```
entById('app'))
```

```
.
1614:17)
48:21)
59:21)
:69:35)
```

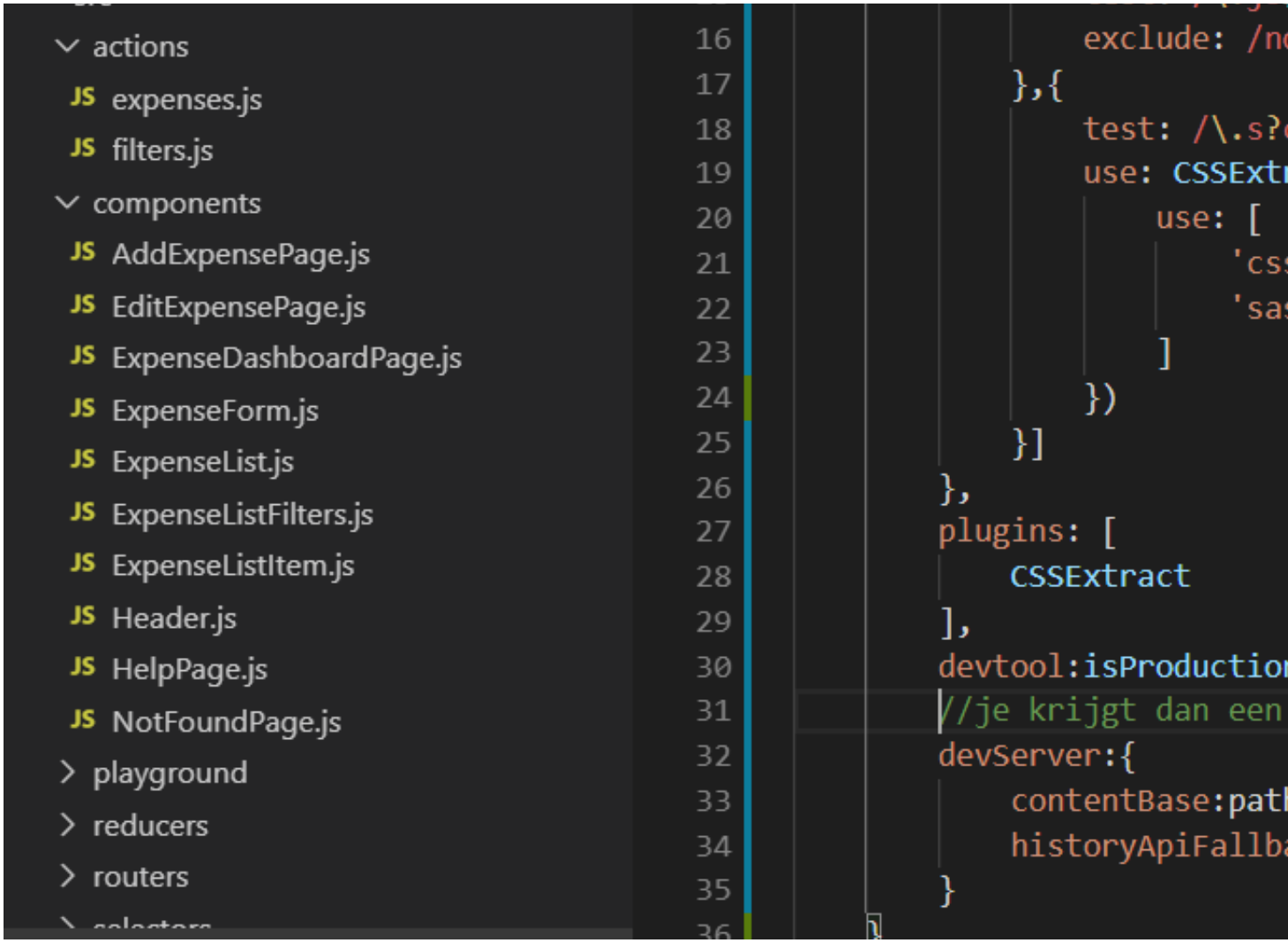
```
    wrapper.find('withStyles(DateRangePicker)').prop('onFocusCha
nge')(calendarFocused); =new
    wrapper.find('DateRangePicker').prop('onFocusChange')
(calendarFocused); = oud
```

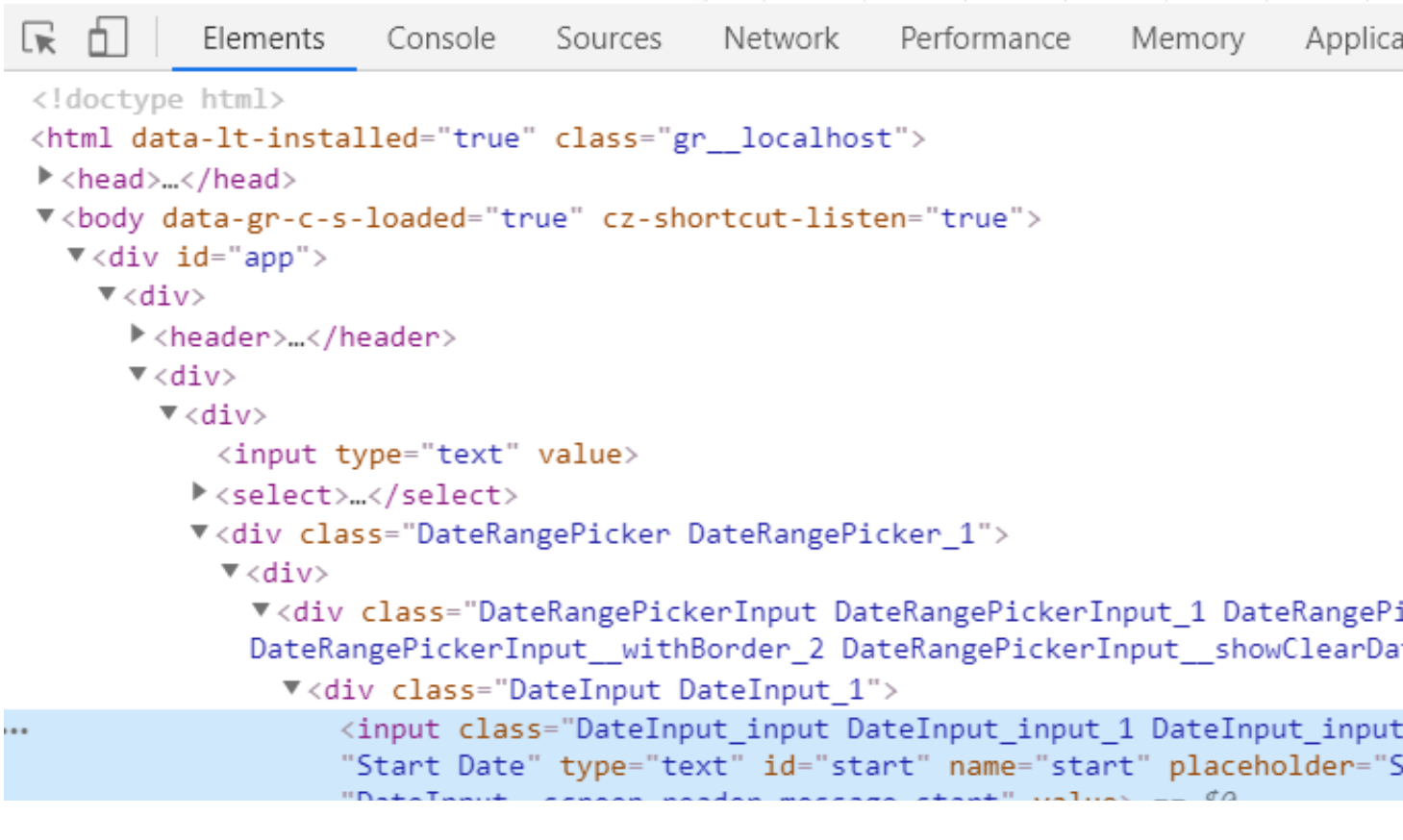
Warning: Please use `require("history").createBrowserHistory` instead of `require("history/createBrowserHistory")`. Support for the latter will be removed in the next major release. IN CONSOLE ==>> npm install react-router-dom@latest



[redacted]

```
module.exports
);
equire('extract-text-webpack-plugin');

v === 'production';
ExtractTextPlugin('styles.css');

s',

(__dirname , 'public'),
le.js'



bel-loader',
$/,  //eindigt met js zodat jsx werkt babel
```


actions
expenses.js
filters.js
components
AddExpensePage.js
EditExpensePage.js
ExpenseDashboardPage.js
ExpenseForm.js
ExpenseList.js
ExpenseListFilters.js
ExpenseListItem.js
Header.js
HelpPage.js
NotFoundPage.js
playground
reducers
routers
exclude: /n
},{
test: /\.s?
use: CSSExt
use: [
'cs
'sa
]
})
}]
},
plugins: [
CSSExtract
],
devtool:isProductio
//je krijgt dan een
devServer:{
contentBase:pat
historyApiFallb
}


Elements
Console
Sources
Network
Performance
Memory
Applica
<!doctype html>
<html data-lt-installed="true" class="gr__localhost">
<head>…</head>
<body data-gr-c-s-loaded="true" cz-shortcut-listen="true">

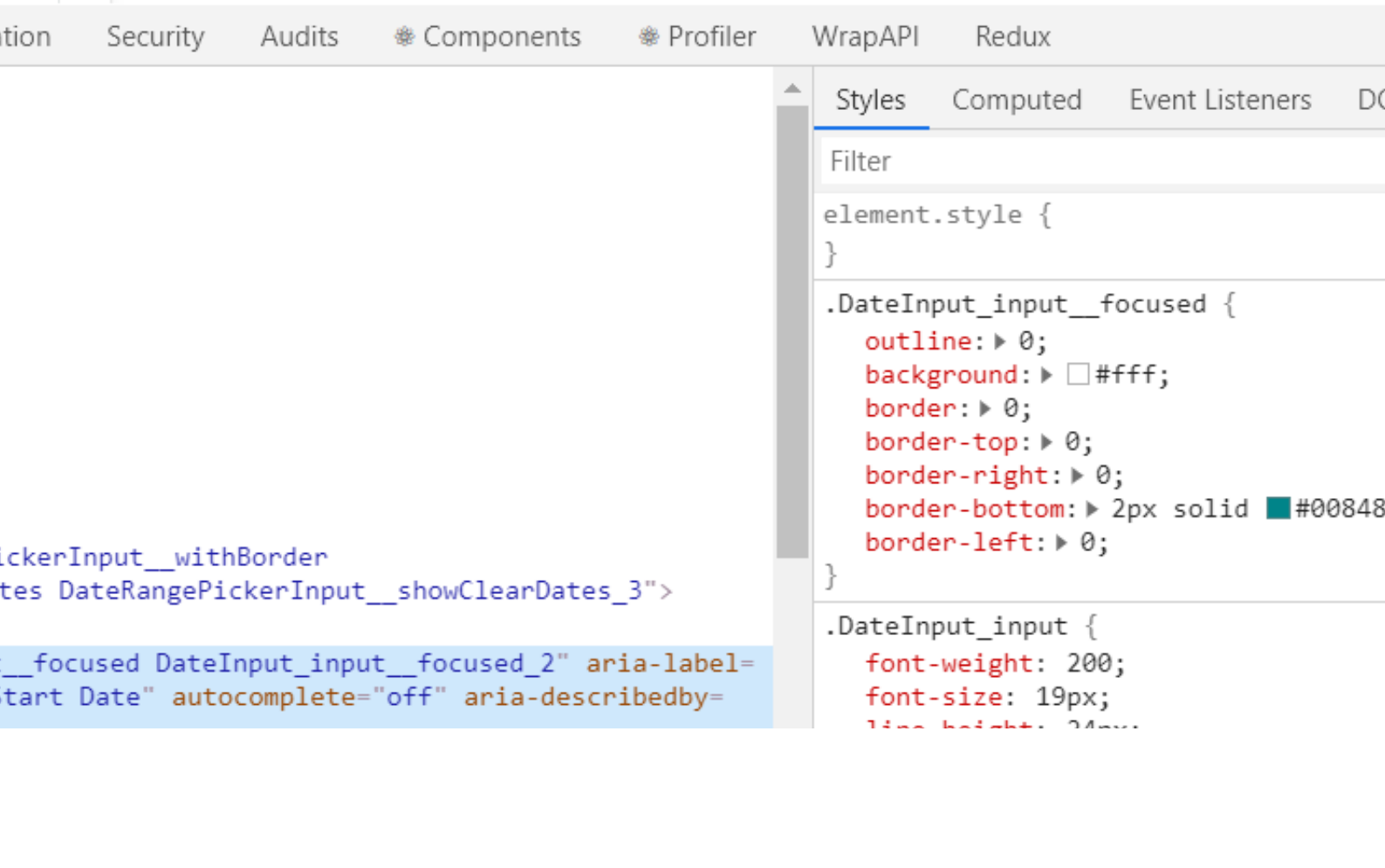
<header>…</header>

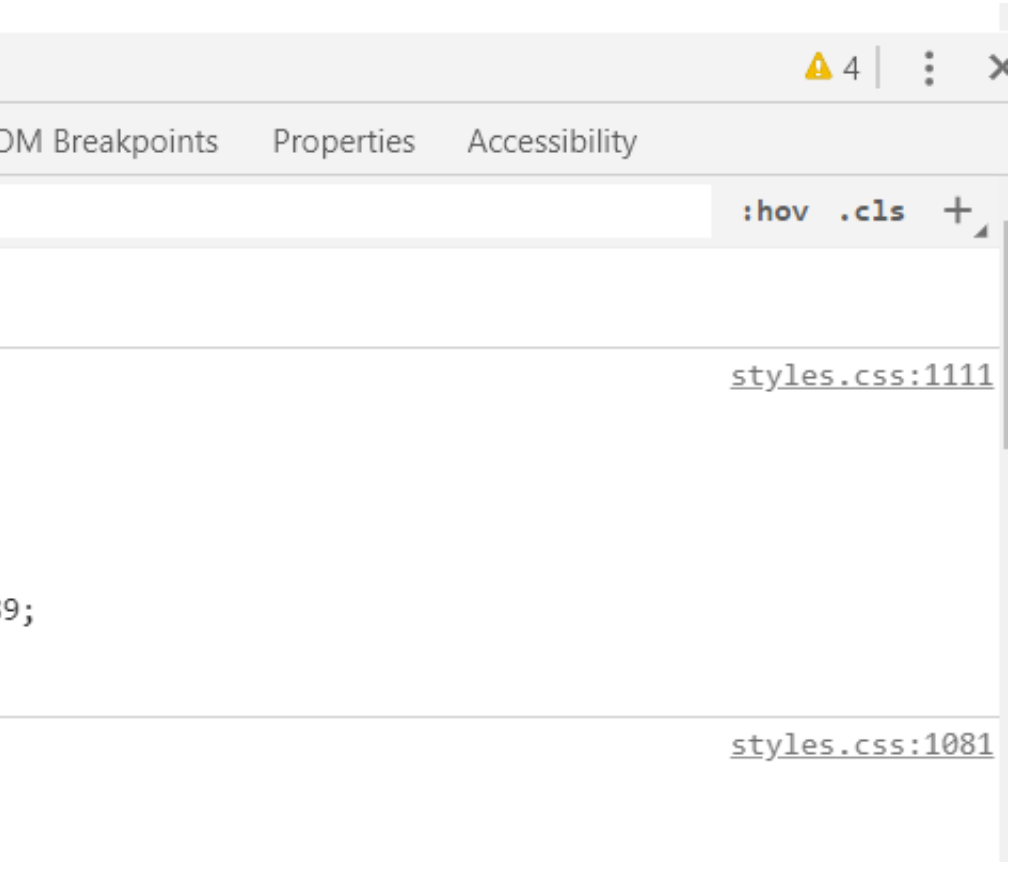
<input type="text" value>
<select>…</select>

<div class="DateRangePickerInput DateRangePickerInput_1 DateRangePi
DateRangePickerInput__withBorder_2 DateRangePickerInput__showClearDa

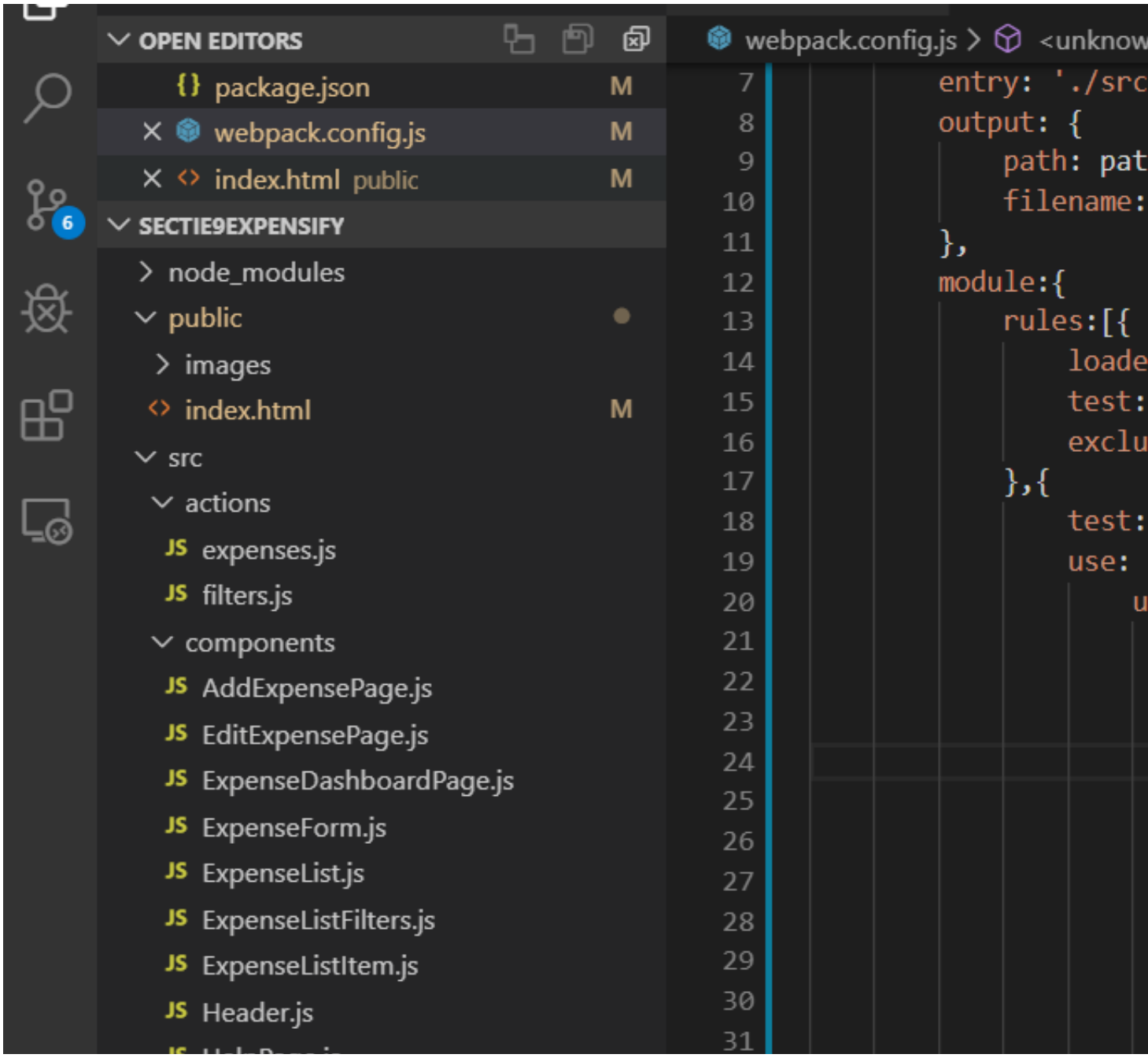
<input class="DateInput_input DateInput_input_1 DateInput_input
"Start Date" type="text" id="start" name="start" placeholder="S

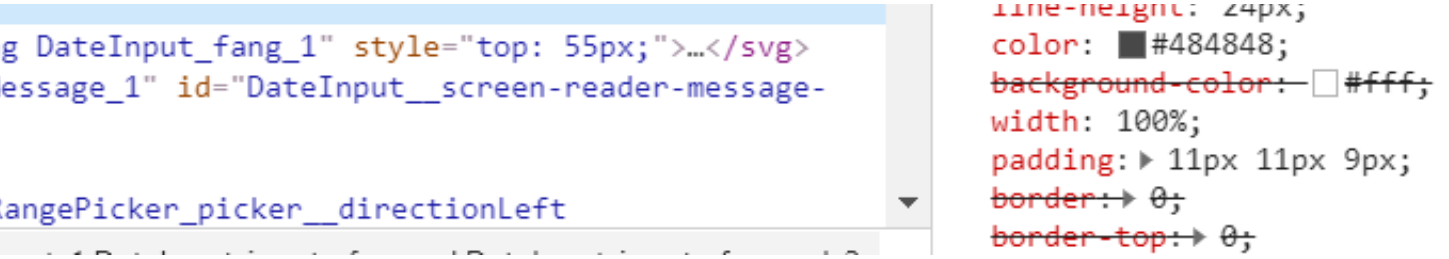
```
ode_modules/ //we don't want that babel is running in nodemodules

css$/, //css en sccs
ract.extract({

s-loader',
ss-loader'




n ? 'source-map' : 'cheap-module-eval-source-map',
 link naar foutmelding, zonder dit krijg je naar babelrc link

h.join(__dirname,'public'),
ack: true //handeling routing via client-side route
```

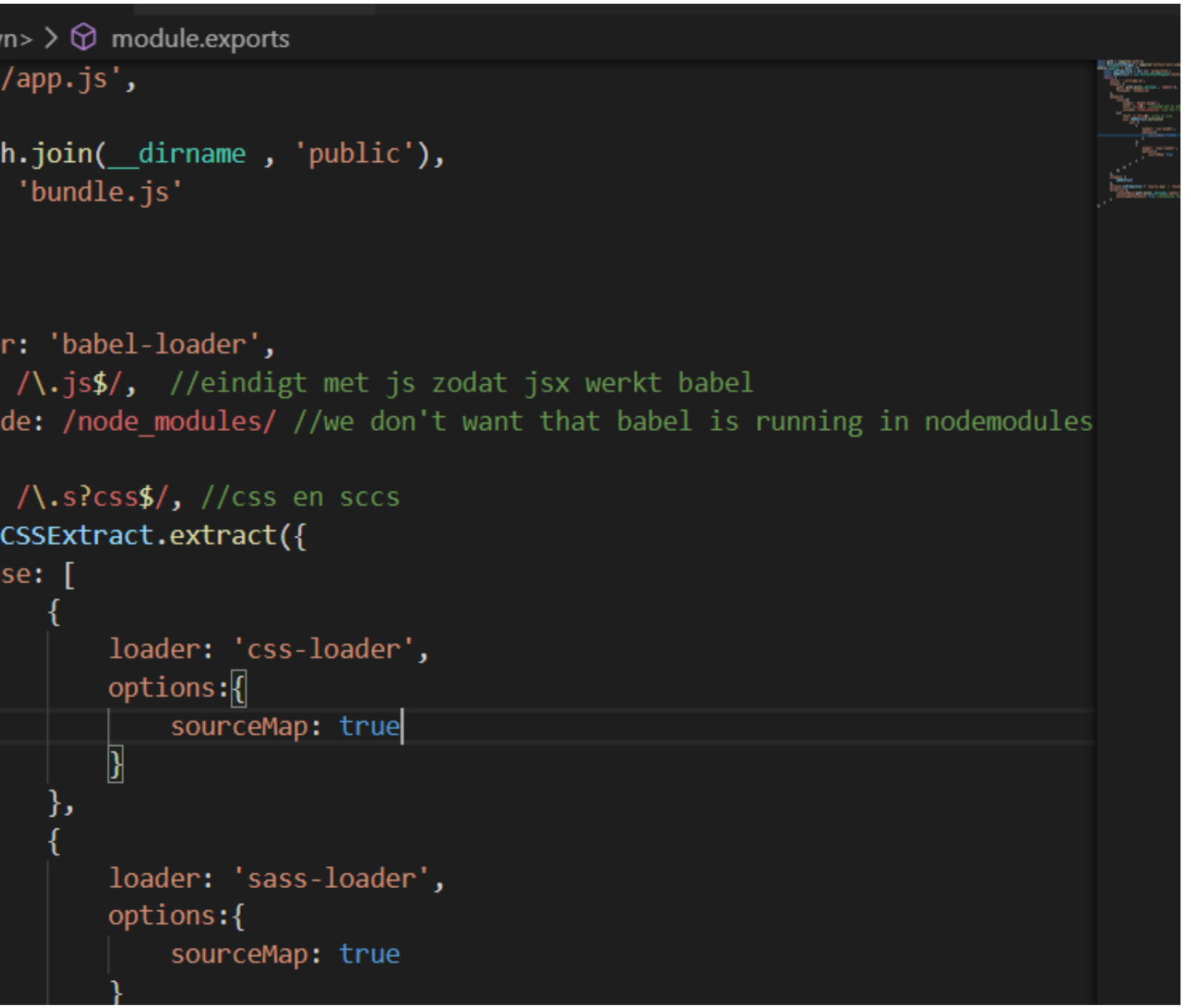


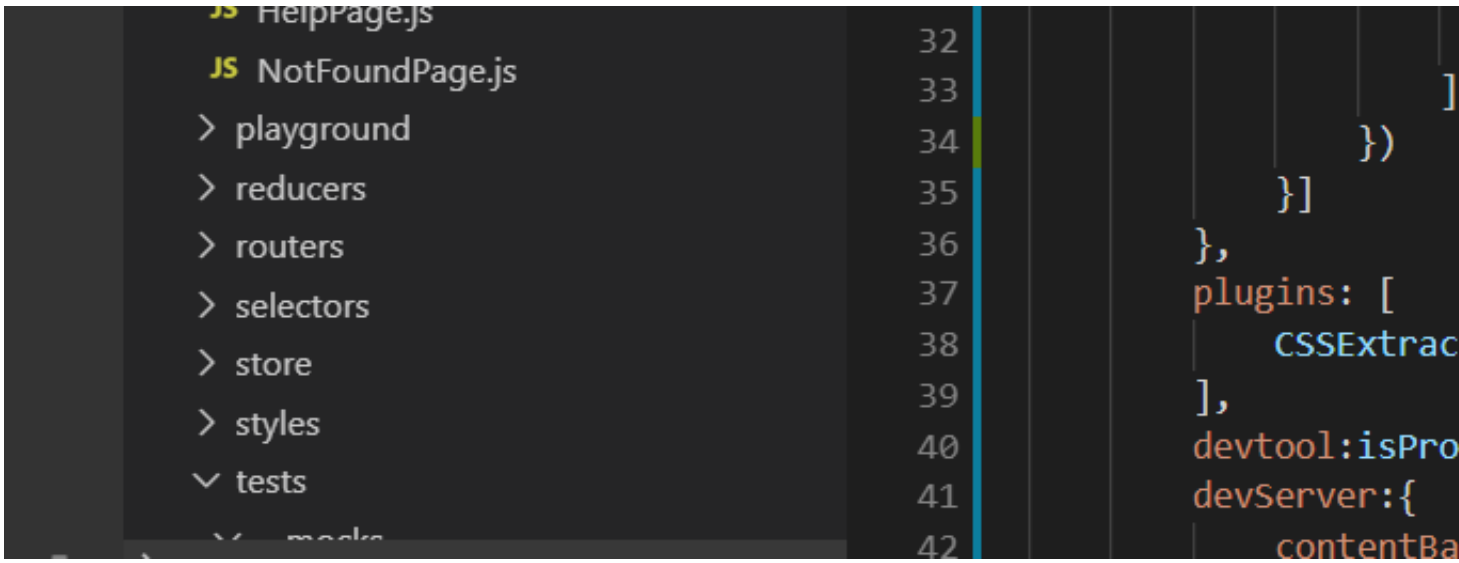

4
DM Breakpoints
Properties
Accessibility
:hov .cls
styles.css:1111
9;
styles.css:1081

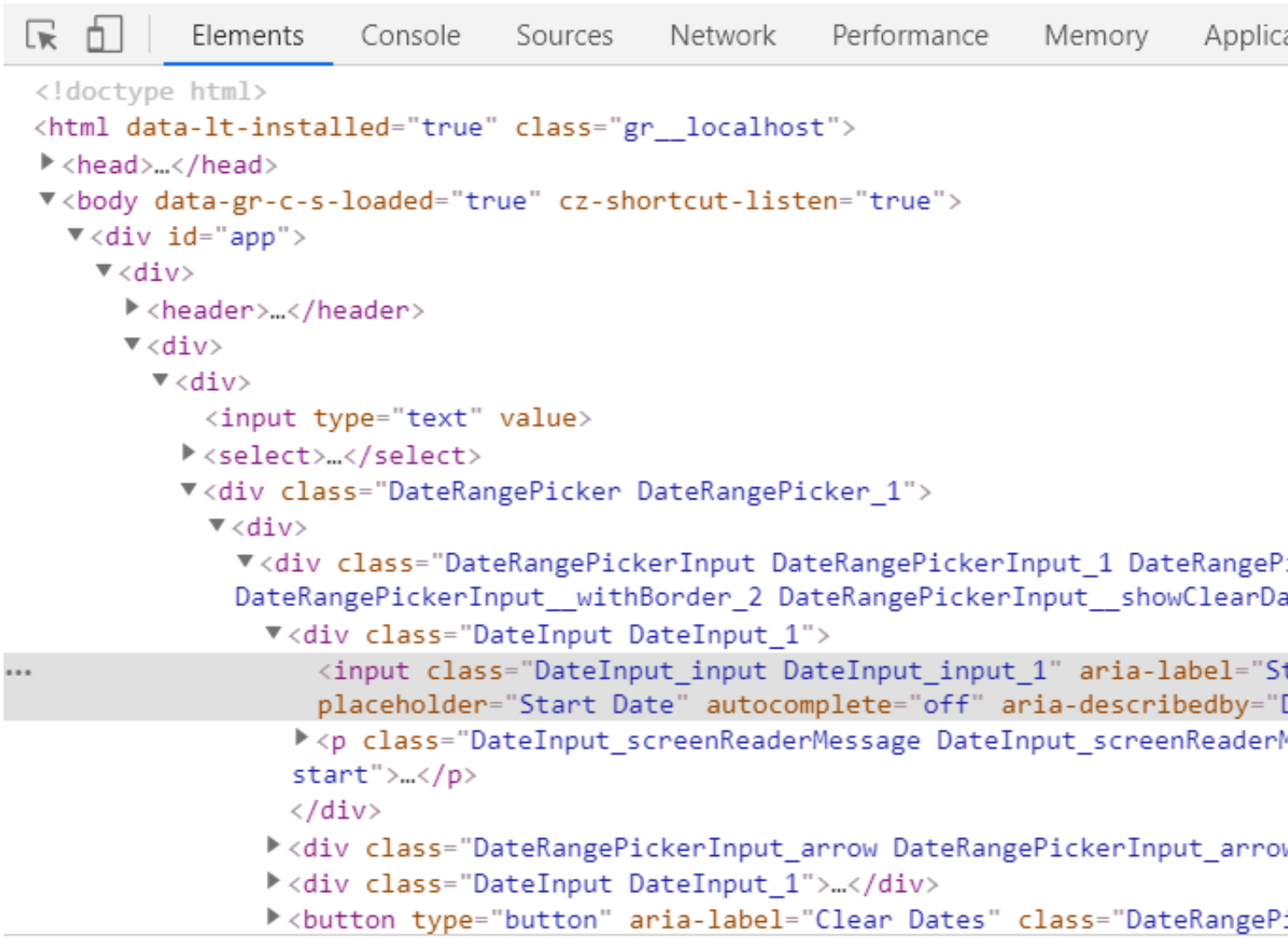
```
DateInput__screen-reader-message-start" value> == $0
▶<svg role="presentation" focusable="false" class="DateInput_fan
▶<p class="DateInput_screenReaderMessage DateInput_screenReaderM
start">…</p>
</div>
▶<div class="DateRangePicker_picker DateRangePicker_picker_1 DateR
```




g DateInput_fang_1" style="top: 55px;">…</svg>
essage_1" id="DateInput__screen-reader-message-
angePicker_picker__directionLeft
line-height: 24px;
color: #484848;
background-color: #fff;
width: 100%;
padding: 11px 11px 9px;
border: 0;
border-top: 0;


module.exports
/app.js',
h.join(__dirname , 'public'),
'bundle.js'
r: 'babel-loader',
/\.js$/, //eindigt met js zodat jsx werkt babel
de: /node_modules/ //we don't want that babel is running in nodemodules
/\.s?css$/, //css en sccs
CSSExtract.extract({
se: [
{
loader: 'css-loader',
options:{
sourceMap: true
}
},
{
loader: 'sass-loader',
options:{
sourceMap: true
}


HelpPage.js
NotFoundPage.js
playground
reducers
routers
selectors
store
styles
tests
32
33
34
35
36
37
38
39
40
41
42
}]
},
plugins: [
CSSExtrac
],
devtool:isPro
devServer:{
contentBa


Elements
Console
Sources
Network
Performance
Memory
Applica
<!doctype html>
<html data-lt-installed="true" class="gr__localhost">
<head>…</head>
<body data-gr-c-s-loaded="true" cz-shortcut-listen="true">

<header>…</header>

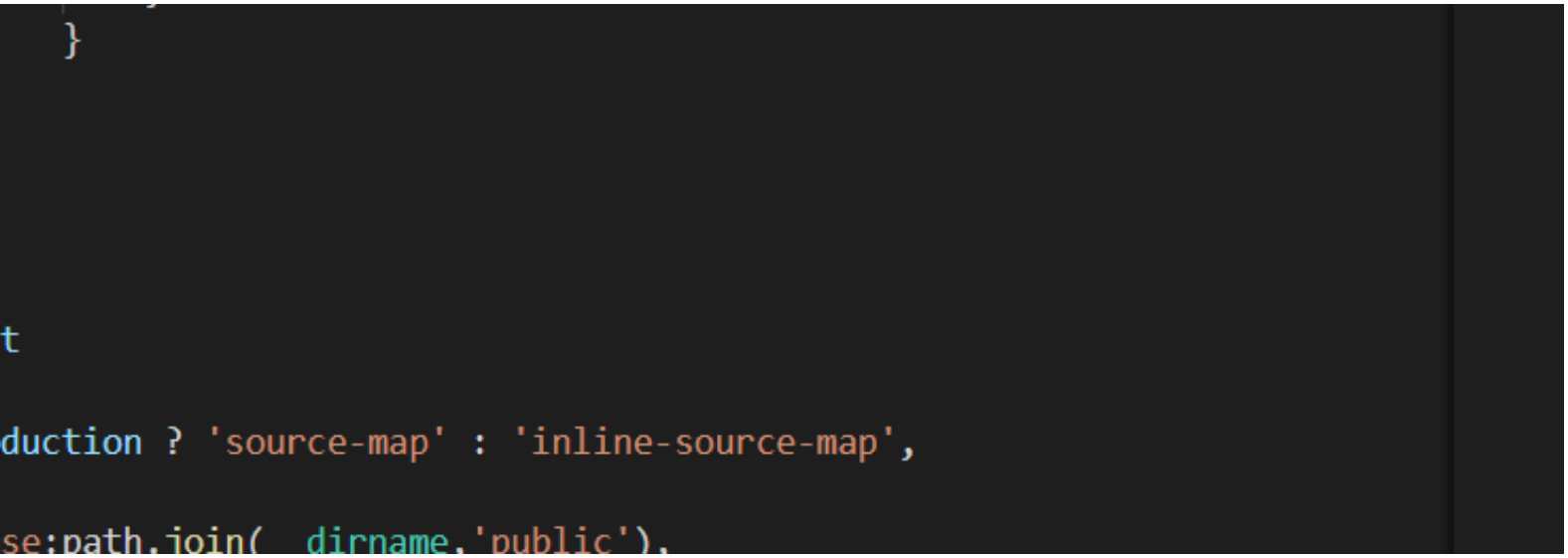
<input type="text" value>
<select>…</select>

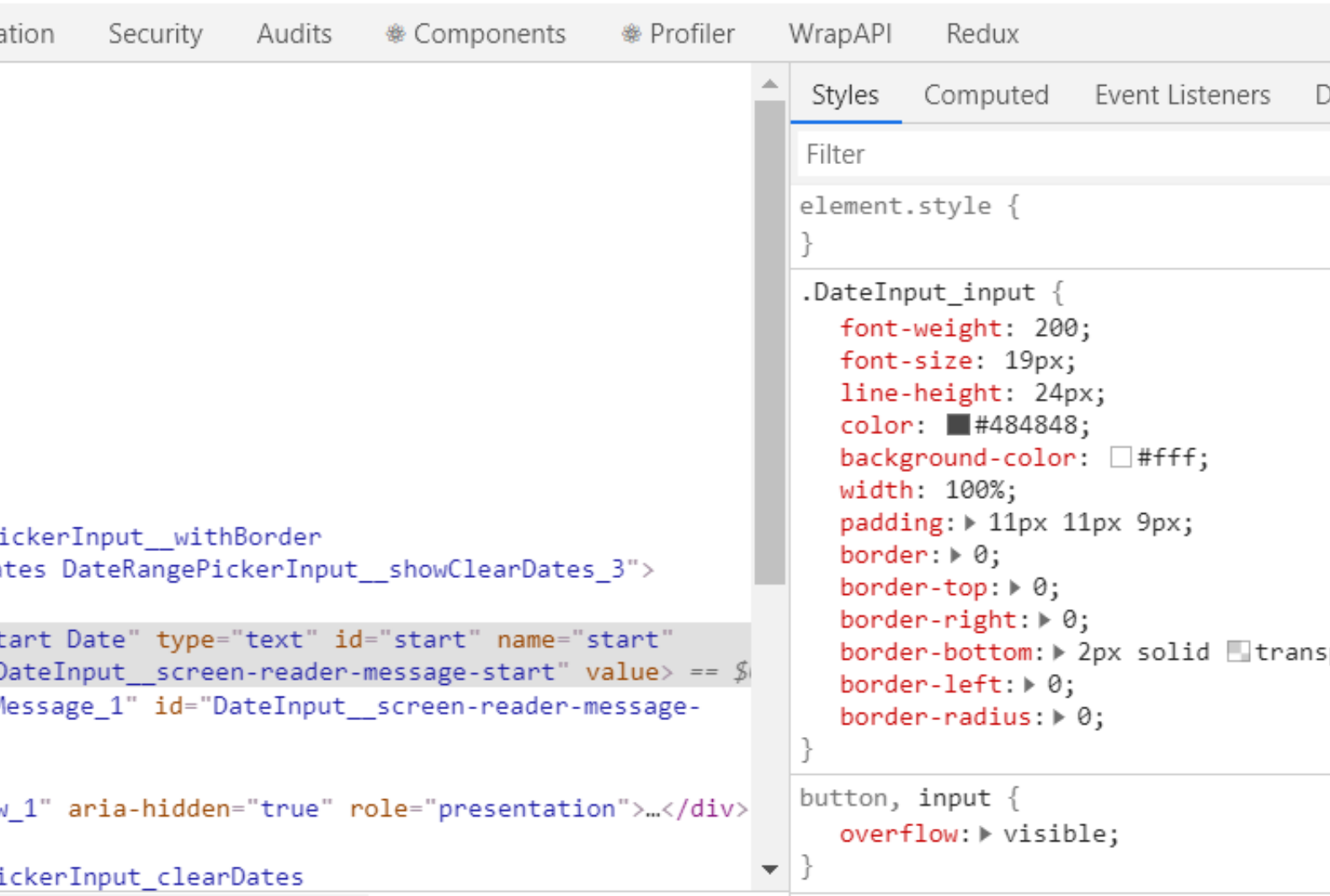
<div class="DateRangePickerInput DateRangePickerInput_1 DateRangeP
DateRangePickerInput__withBorder_2 DateRangePickerInput__showClearDa

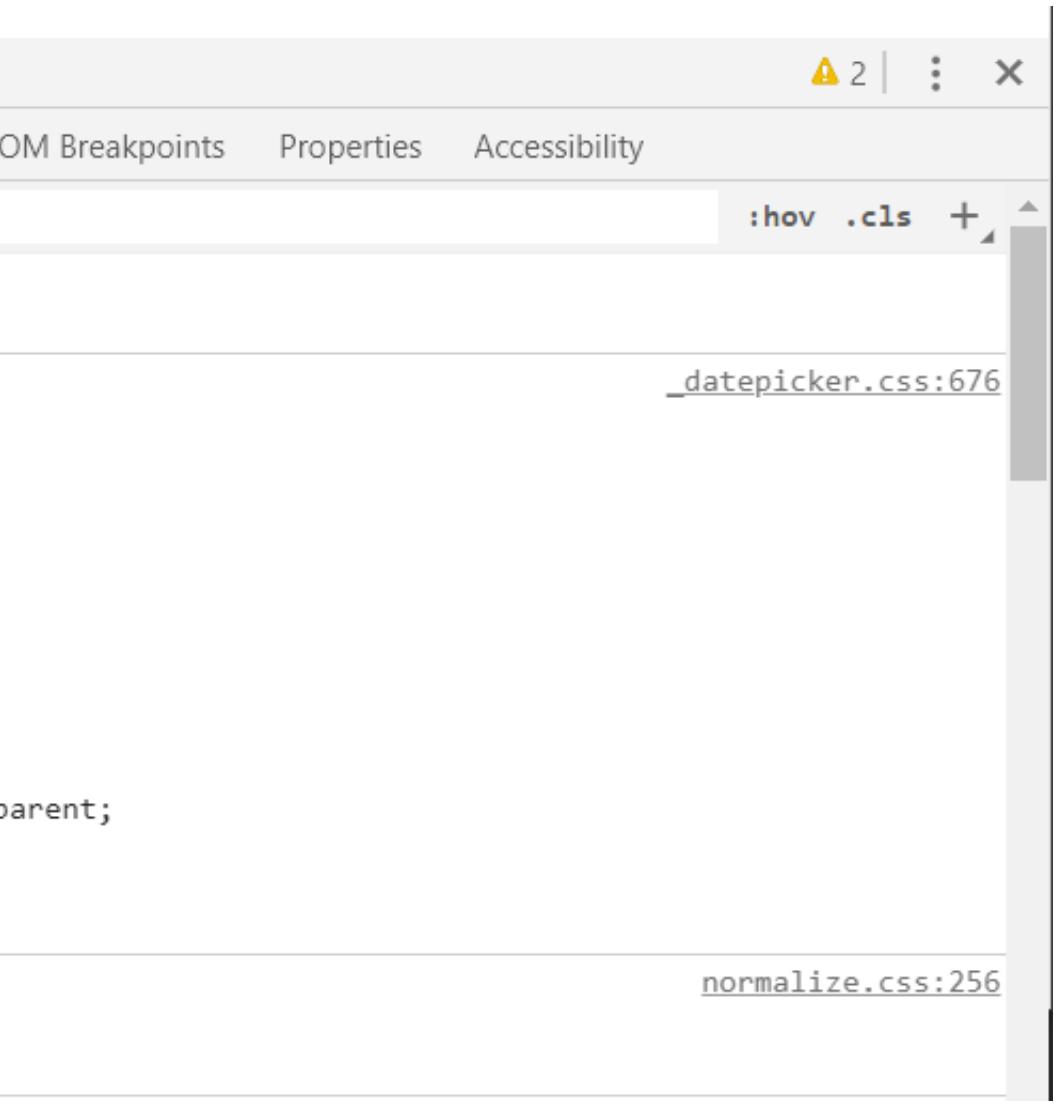
<input class="DateInput_input DateInput_input_1" aria-label="St
placeholder="Start Date" autocomplete="off" aria-describedby="D
<p class="DateInput_screenReaderMessage DateInput_screenReaderM
start">…</p>

<div class="DateRangePickerInput_arrow DateRangePickerInput_arrow
…
<button type="button" aria-label="Clear Dates" class="DateRangeP

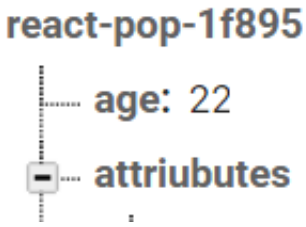
FIREBASE

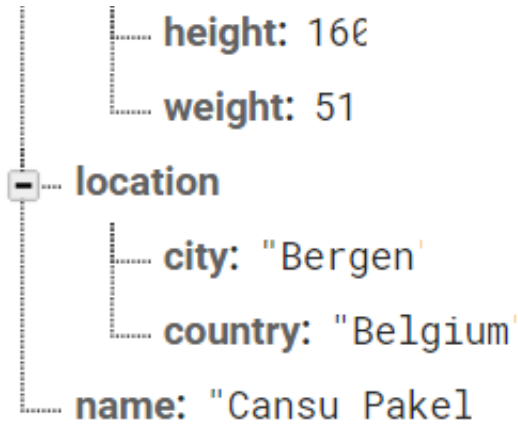

}
t
duction ? 'source-map' : 'inline-source-map',
se:path.join(dirname,'public'),


ation
Security
Audits
Components
Profiler
WrapAPI
Redux
ickerInput__withBorder
ates DateRangePickerInput__showClearDates_3">
tart Date" type="text" id="start" name="start"
DateInput__screen-reader-message-start" value> == $
Message_1" id="DateInput__screen-reader-message-
w_1" aria-hidden="true" role="presentation">…
ickerInput_clearDates
Styles
Computed
Event Listeners
Filter
element.style {
}
.DateInput_input {
font-weight: 200;
font-size: 19px;
line-height: 24px;
color: #484848;
background-color: #fff;
width: 100%;
padding: 11px 11px 9px;
border: 0;
border-top: 0;
border-right: 0;
border-bottom: 2px solid transp
border-left: 0;
border-radius: 0;
}
button, input {
overflow: visible;
}


2
OM Breakpoints
Properties
Accessibility
:hov .cls
_datepicker.css:676
parent;
normalize.css:256

```
import  *  as firebase from 'firebase';

const firebaseConfig = {
    apiKey: "AIzaSyBp-0i4bx_mDEnsbiT2o5HG7VZNFHhOUts",
    authDomain: "react-pop-1f895.firebaseapp.com",
    databaseURL: "https://react-pop-1f895.firebaseio.com",
    projectId: "react-pop-1f895",
    storageBucket: "react-pop-1f895.appspot.com",
    messagingSenderId: "668529991201",
    appId: "1:668529991201:web:99f589c07c488320740d60",
    measurementId: "G-BLZ6C3Z83T"
  };

firebase.initializeApp(firebaseConfig);
const database = firebase.database();

database.ref().set({
    name:'Cansu Pakel',
    age:21,
    location:{
        city:'Gent',
        country:'Belgium'
    }
});

database.ref('age').set(22);
database.ref('location/city').set('Bergen');

database.ref('attriubutes').set({
    height:160,
    weight:51
})
```






```
database.ref().set({
    name:'Cansu Pakel',
    age:21,
    location:{
        city:'Gent',
        country:'Belgium'
    }
}).then(()=>{
    console.log('Data is saved');
}).catch((error)=>{
    console.log('This failed', error)
});

//database.ref('age').set(22);
//database.ref('location/city').set('Bergen');
```

```
database.ref('age')
    .remove()
    .then(()=>{
        console.log('data was removed')
    }).catch(()=>{
        console.log('data was not removed')
    })

database.ref('age').set(null) //word ook verwijderd
```