REACT: HTML STILL MATTERS*

@jbmoelker

REACT: HTML STILL MATTERS*

**unless you're developing for React Native*

@jbmoelker

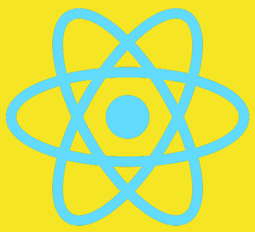
ABOUT ME

Jasper Moelker

@jbmoelker

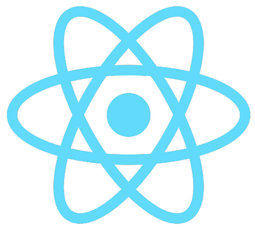
front-end architect

@devoorhoede



React amateur

DEVELOPERS ♡ CONVENIENCE



REACT

- does **1 thing**, does it well
- is **module** oriented
- plays well with other **JS**
- has **declarative** JSX syntax

- has great tools, docs, **community**

JSX is for developers

HTML is for browsers

@jbmoelker

BROWSERS

BROWSERS

modern

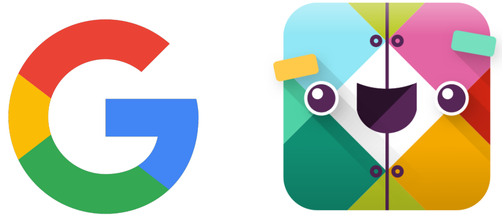
oldies

in-app

proxy

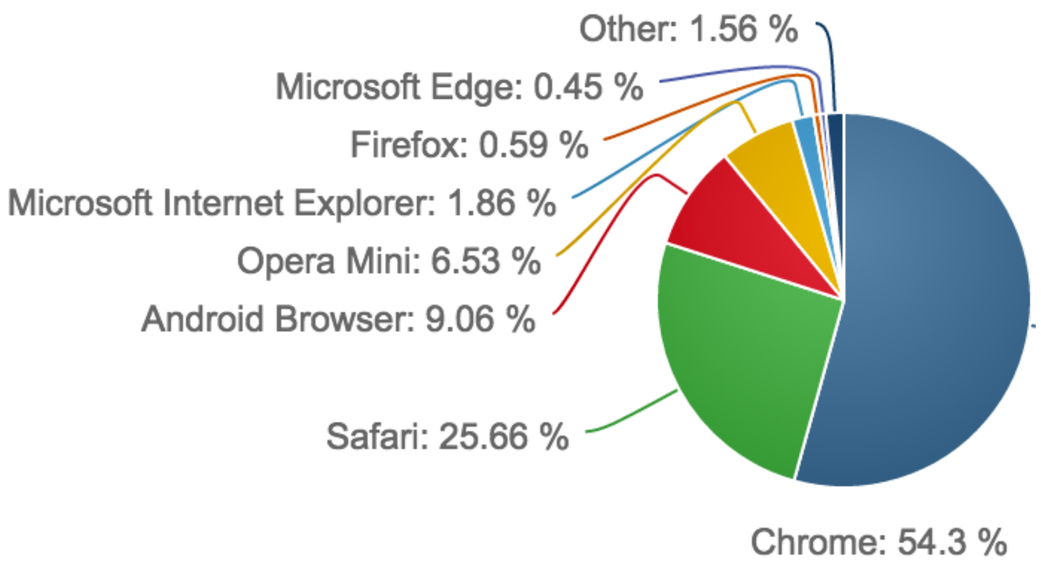
text only

robots

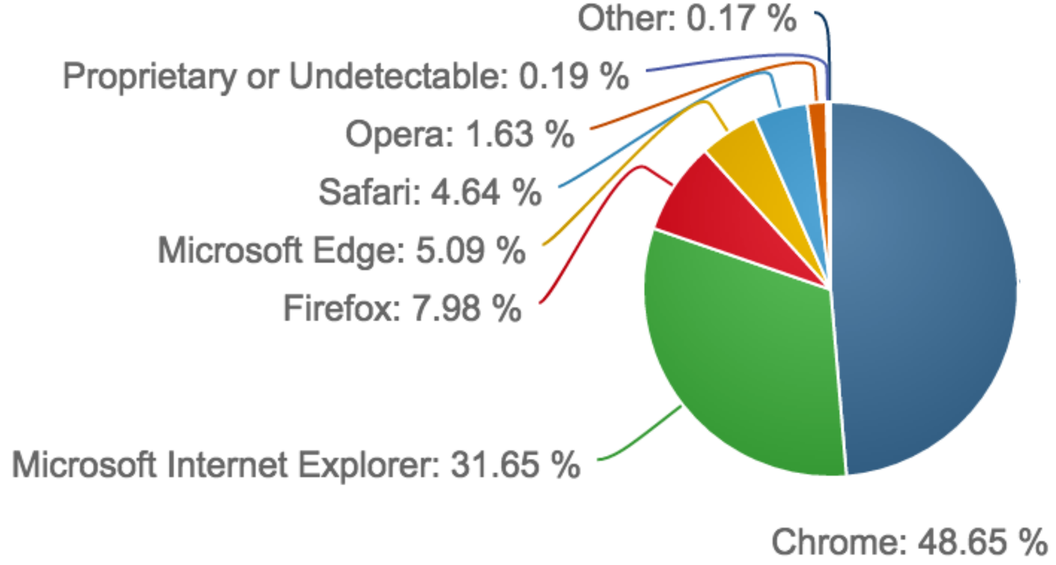


BROWSER MARKETSHARE

mobile



desktop



Opera Mini > 6%
marketshare mobile!

netmarketshare.com

MORE BROWSERS

autospies.com

i.ytimg.com

BROWSER TESTING

@jbmoelker

React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

Get Started | Download React v15.3.2

Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using React Native.

A Simple Component

React components implement a `render()` method that takes input data and returns what to display. This example uses an XML-like syntax called JSX. Input data that is passed into the component can be accessed by `render()` via `this.props`.

React is flexible and provides hooks that allow you to interface with other libraries and frameworks. This example uses **remarkable**, an external Markdown library, to convert the textarea's value in real time.

Live JSX Editor | Compiled JS

```
class MarkdownEditor extends React.Component {
  constructor(props) {
    super(props);
    this.handleChange = this.handleChange.bind(this);
    this.state = {value: 'Type some *markdown* here!'};
  }

  handleChange() {
    this.setState({value: this.refs.textarea.value});
  }

  getRawMarkup() {
    var md = new Remarkable();
    return { __html: md.render(this.state.value) };
  }

  render() {
    return (
      <div className="MarkdownEditor">
        <h3>Input</h3>
        <textarea
          onChange={this.handleChange}
          ref="textarea"
          defaultValue={this.state.value} />
        <h3>Output</h3>
        <div
          className="content"
          dangerouslySetInnerHTML={this.getRawMarkup()}
        />
      </div>
    );
  }
}

ReactDOM.render(<MarkdownEditor />, mountNode);
```

React is flexible and provides hooks that allow you to interface with other libraries and frameworks. This example uses **remarkable**, an external Markdown library, to convert the textarea's value in real time.

Live JSX Editor | Compiled JS

```
class MarkdownEditor extends React.Component {
  constructor(props) {
    super(props);
    this.handleChange = this.handleChange.bind(this);
    this.state = {value: 'Type some *markdown* here!'};
  }

  handleChange() {
    this.setState({value: this.refs.textarea.value});
  }

  getRawMarkup() {
    var md = new Remarkable();
    return { __html: md.render(this.state.value) };
  }

  render() {
    return (
      <div className="MarkdownEditor">
        <h3>Input</h3>
        <textarea
          onChange={this.handleChange}
          ref="textarea"
          defaultValue={this.state.value} />
        <h3>Output</h3>
        <div
          className="content"
          dangerouslySetInnerHTML={this.getRawMarkup()}
        />
      </div>
    );
  }
}

ReactDOM.render(<MarkdownEditor />, mountNode);
```

Input

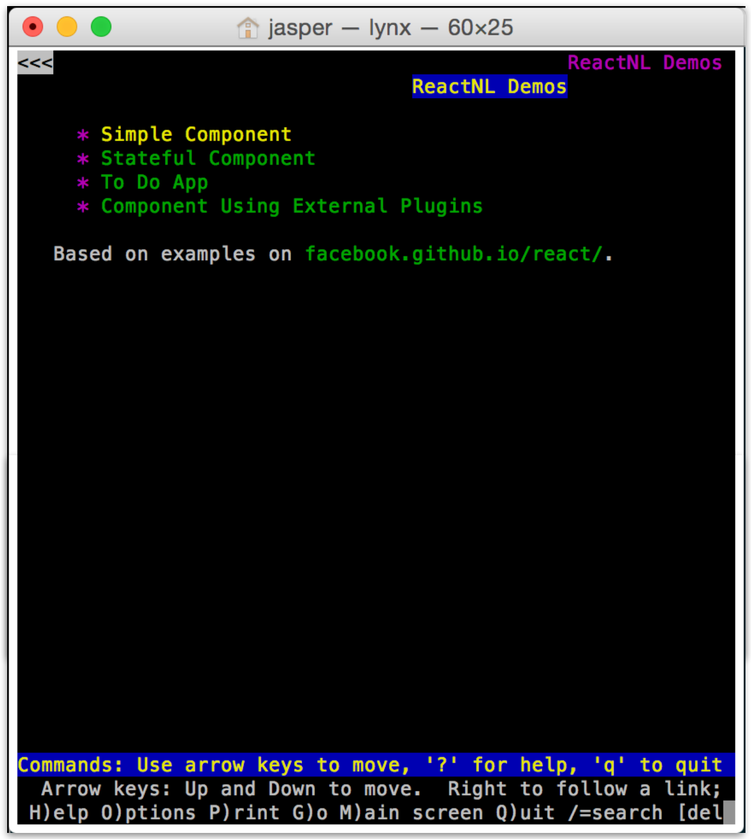
Type some *markdown* here!

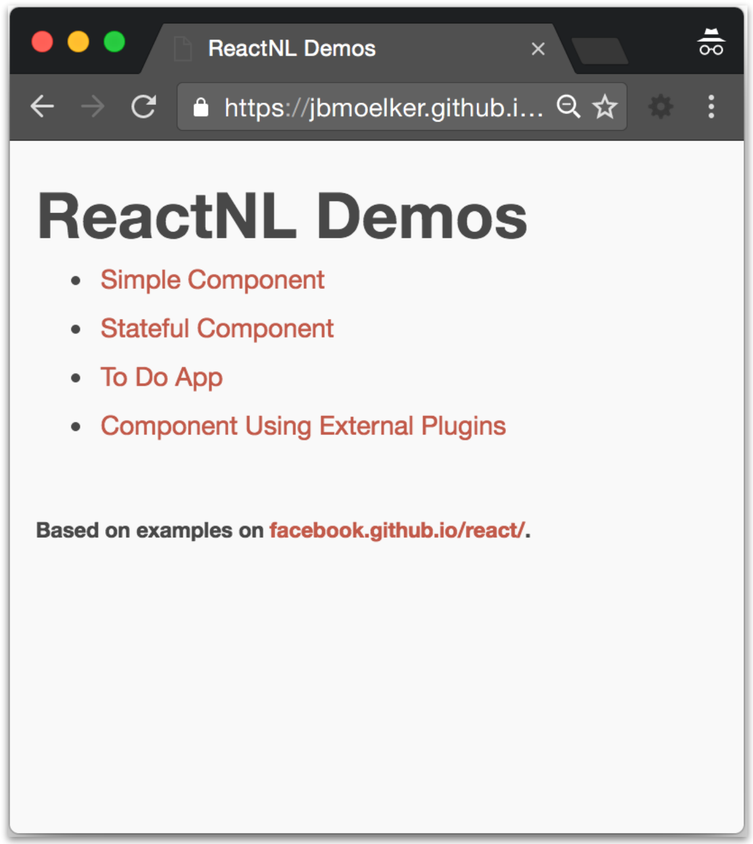
Output

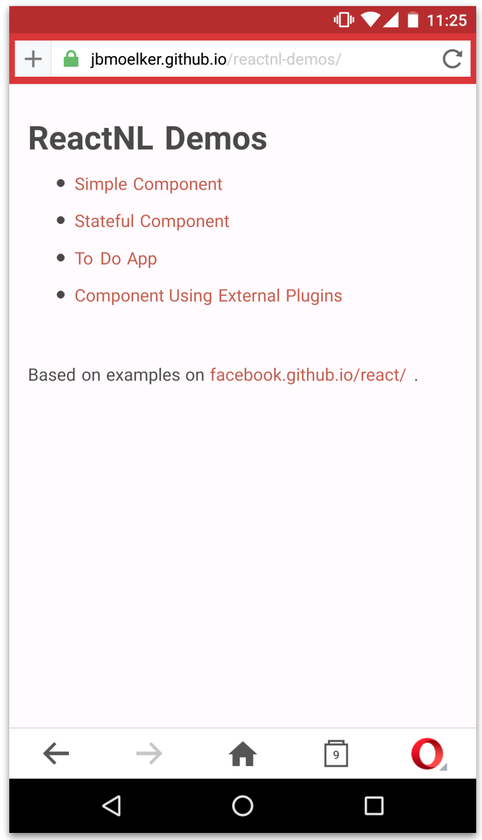
Type some *markdown* here!

facebook.github.io/react/

EXAMPLES FROM REACT DOCS

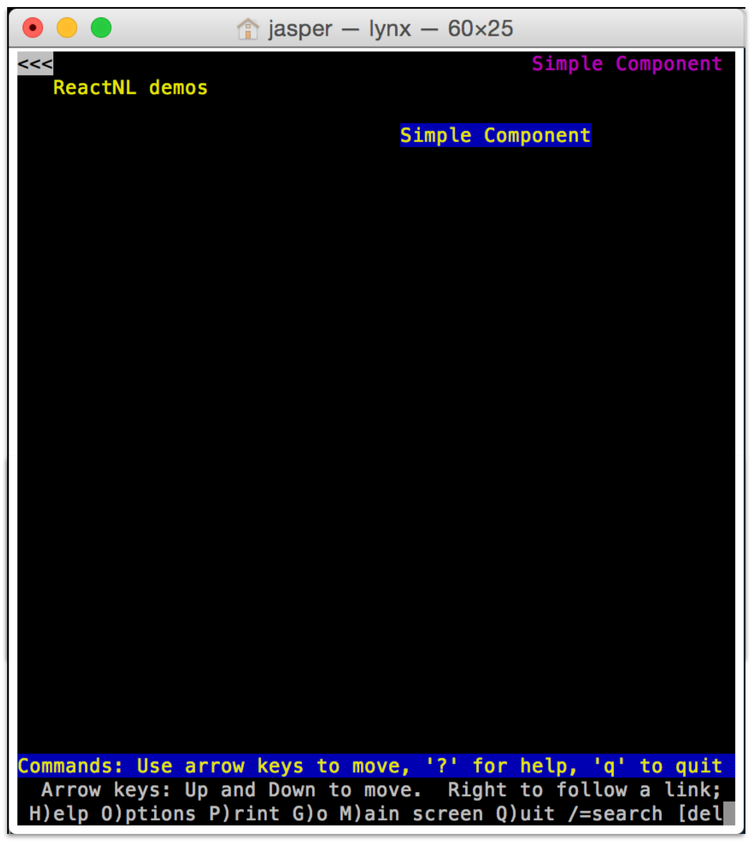


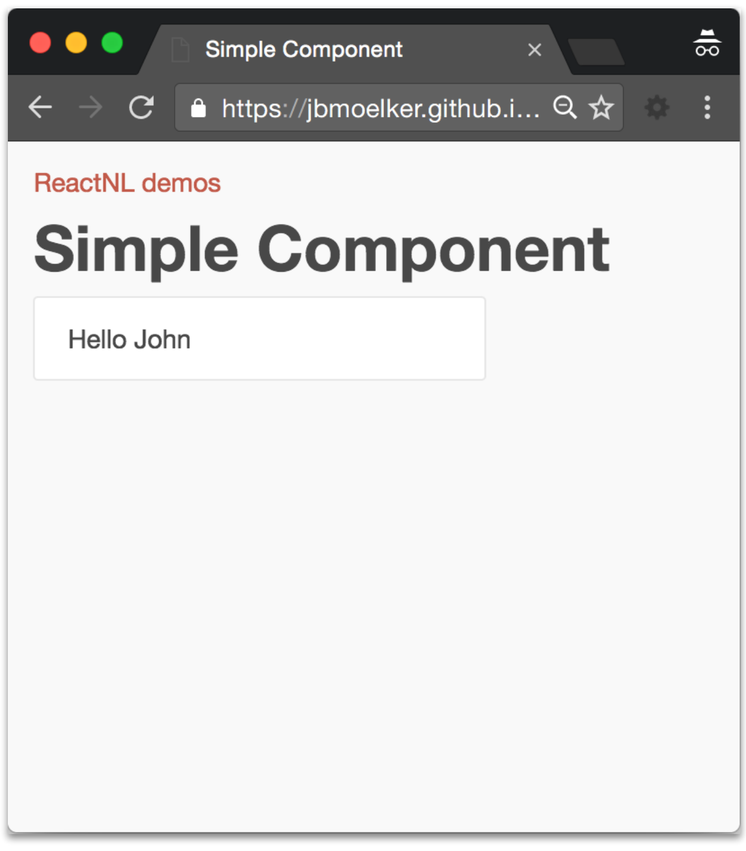


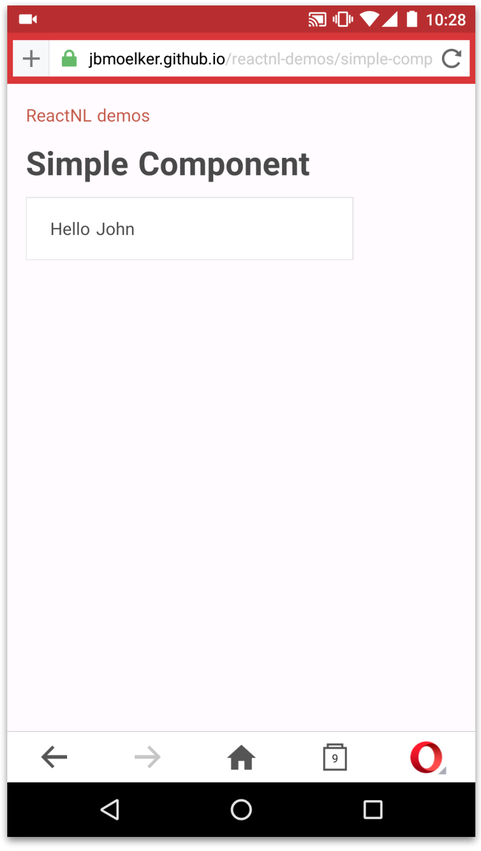


jbmoelker.github.io/reactnl-demos

EXAMPLE: SIMPLE

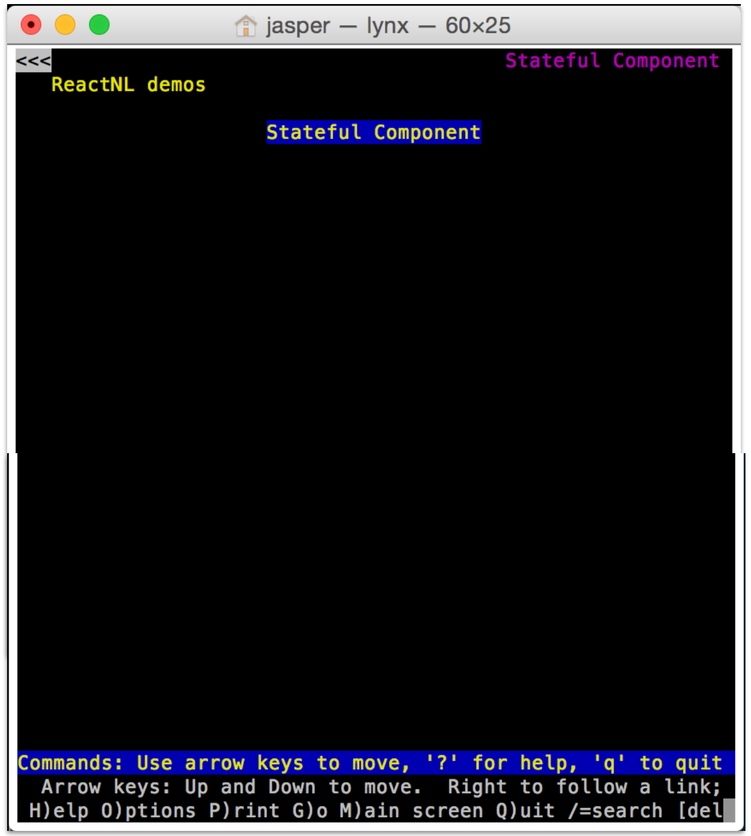


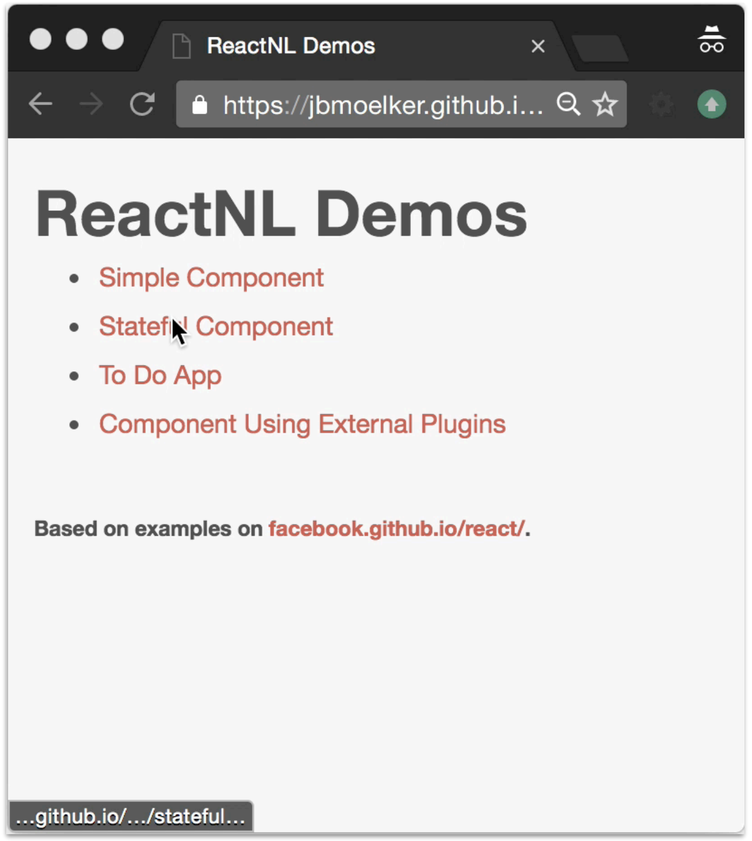


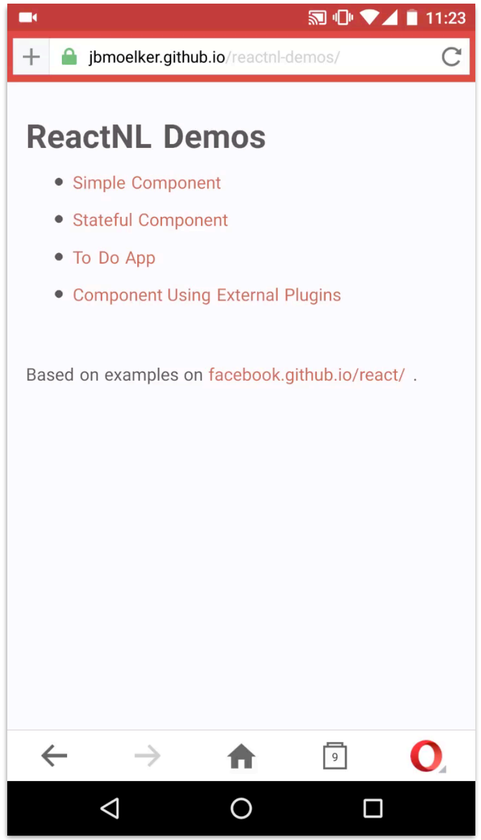


jbmoelker.github.io/reactnl-demos/simple-component

EXAMPLE: STATEFUL

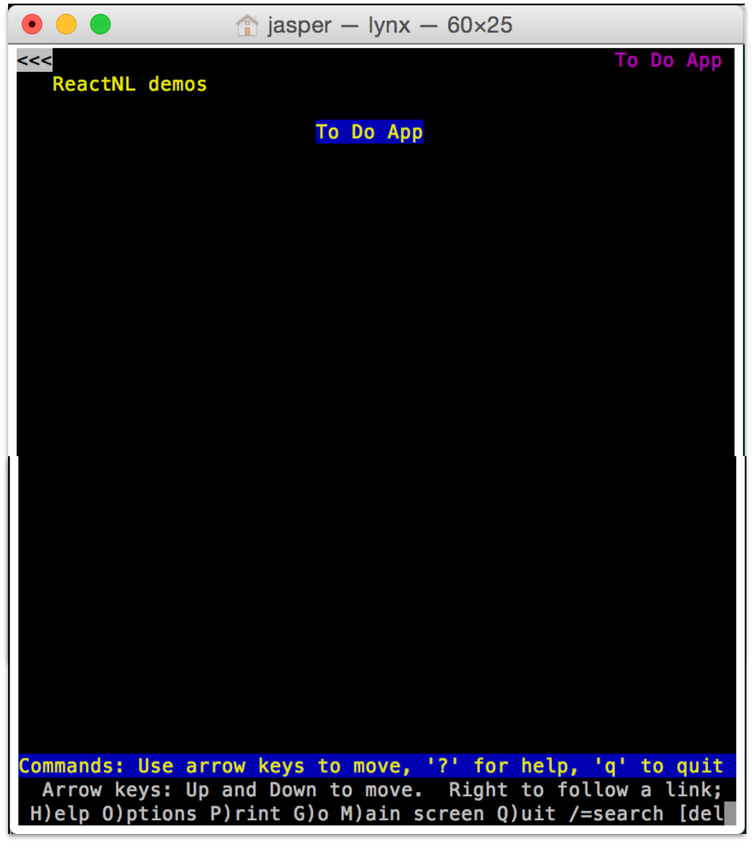


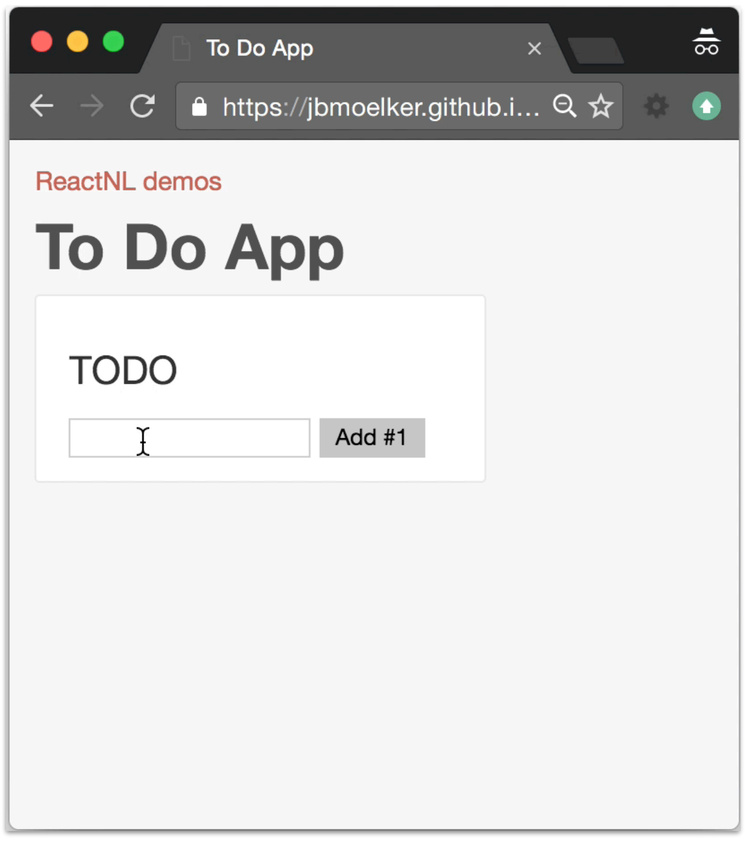


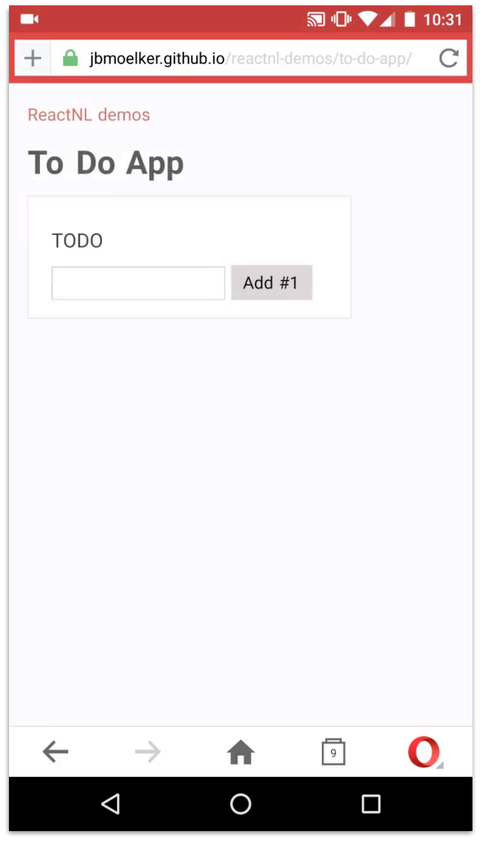


jbmoelker.github.io/reactnl-demos/stateful-component

EXAMPLE: TO DO APP

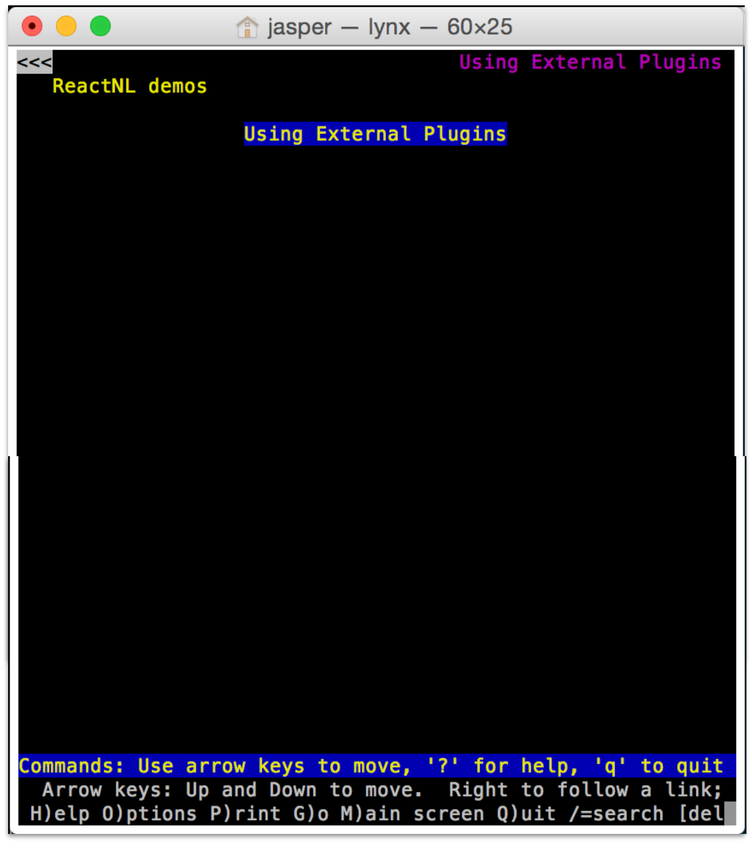


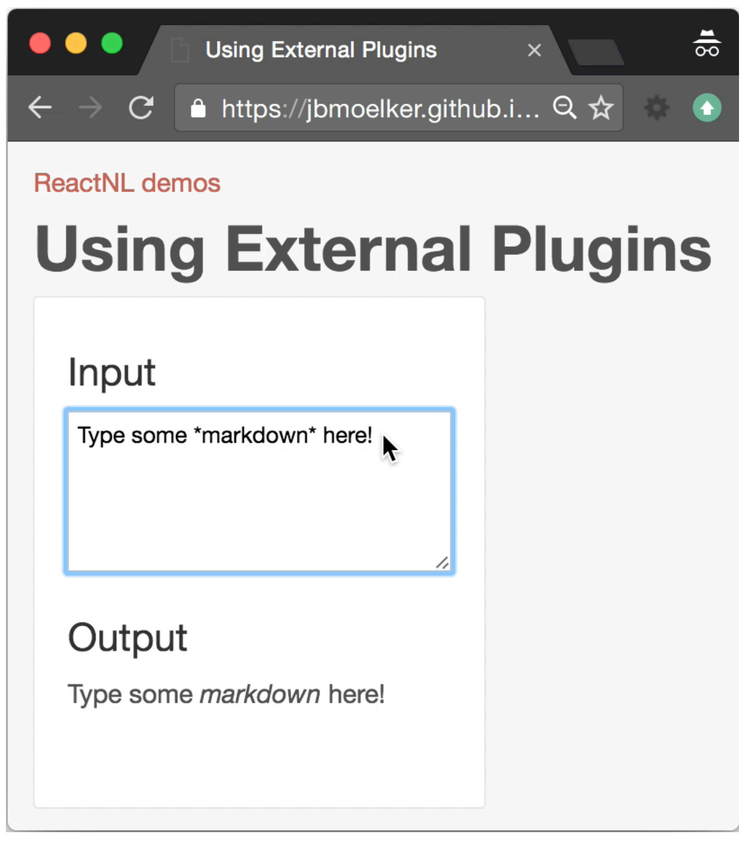


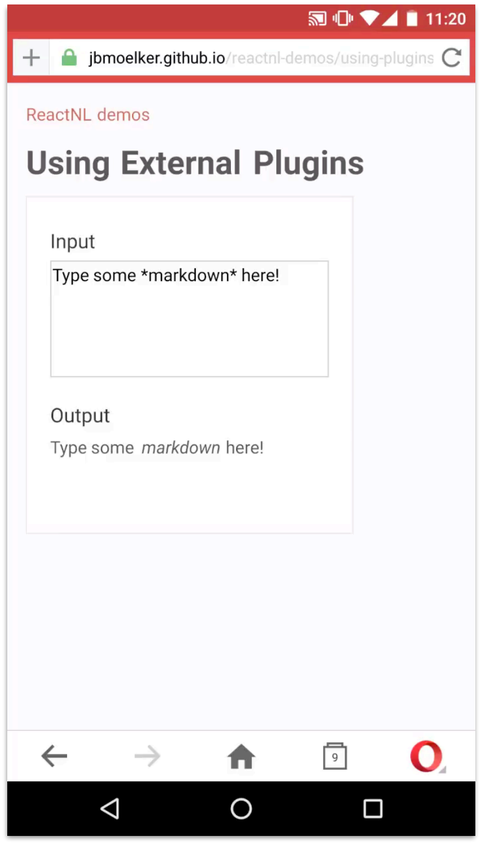


jbmoelker.github.io/reactnl-demos/to-do-app

EXAMPLE: USING PLUGINS







jbmoelker.github.io/reactnl-demos/using-plugins

ENHANCE WITH HTML

@jbmoelker

HTML is a standard

supported by browser vendors,

assistive technologies etc.

@jbmoelker

BUILT-IN SEMANTICS

<**article**> <h1> … </h1> <p> …

<**nav**>   …

<**form**> <fieldset> <legend> …

~~ ~~

BUILT-IN INTERACTIVITY

<**a** href="...">

<**form** action="..." method="...">

<**input** name="..." value="..." required>

<**button**>

BUILT-IN STATE

<form> values

:focus

:target

:disabled

:valid

BUILT-IN RELATIONSHIPS

<button> **submits** <form>

<h1-6> **indexes** <section>

<summary> **toggles** <details>

<label> **describes** <input>

<output> **combines** <input>s

STRUCTURED DATA

[**itemscope**]

[**itemtype**="http://schema.org/..."]

[**itemprop**="..."]

<meta property="**og:**..." content="...">

ENHANCED ACCESSIBILITY

[aria-**role**="alert"]

[aria-**controls**="..."]

[aria-**expanded**]

[aria-**label**="..."]

HTML BEFORE JS

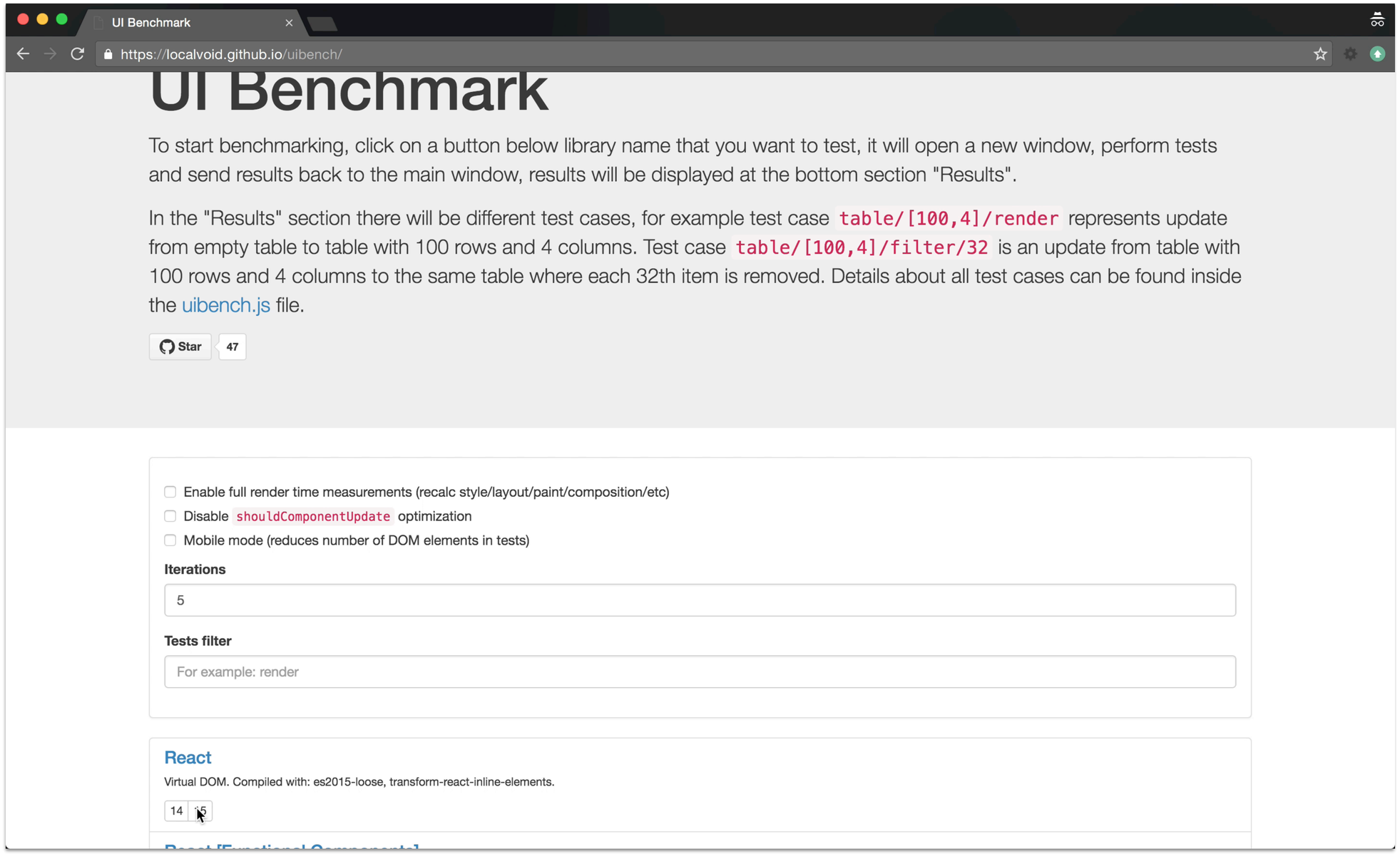
@jbmoelker

React is JavaScript first,

but most user agents are not

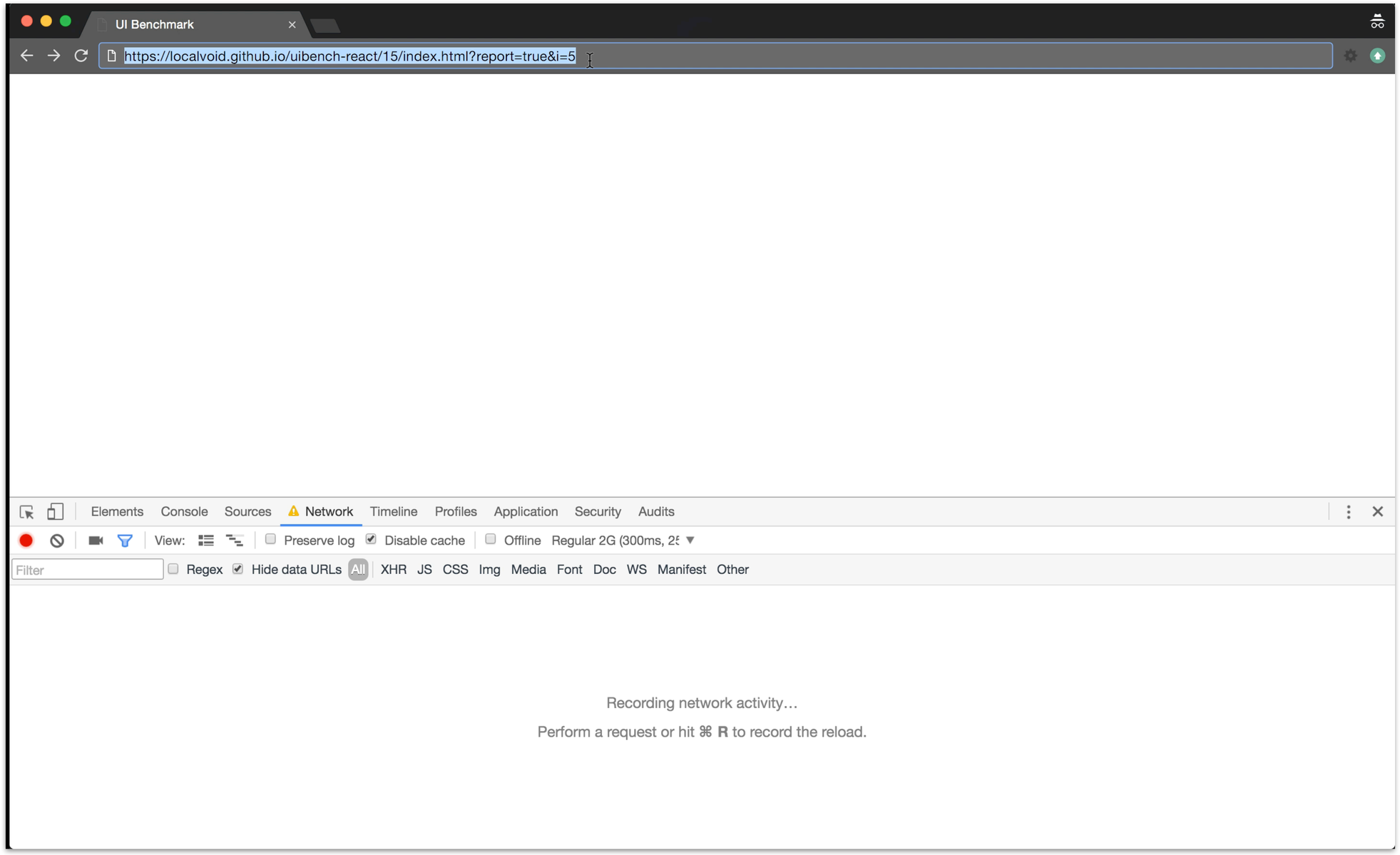
@jbmoelker

REACT IS LIGHTNING FAST ...



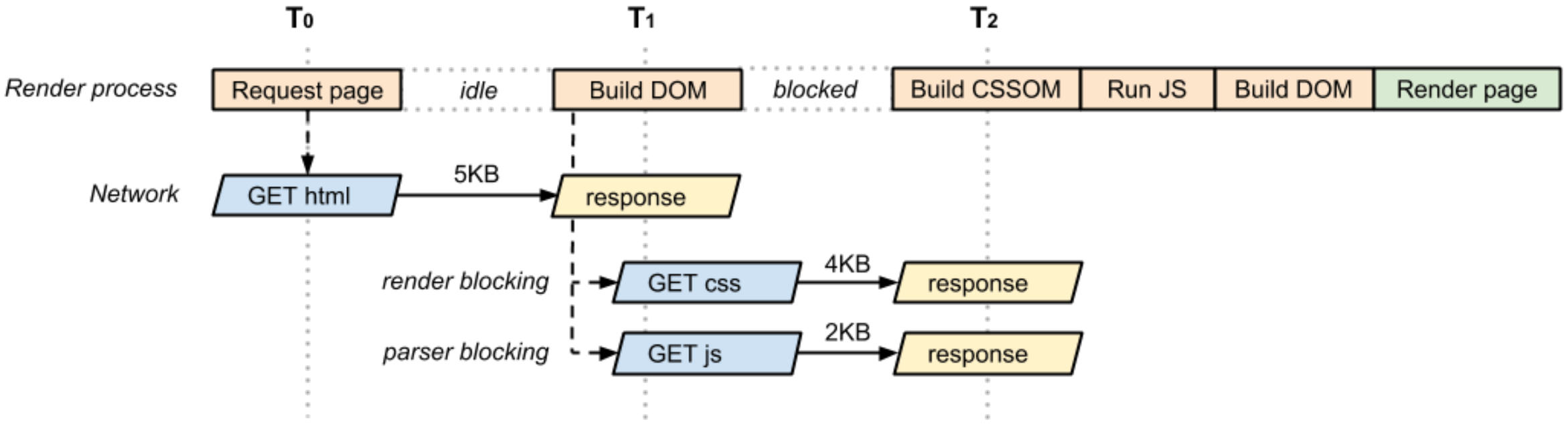
localvoid.github.io/uibench

… AFTER REACT IS LOADED



localvoid.github.io/uibench

CRITICAL RENDERING PATH



critical rendering path on developers.google.com

EVERYONE HAS JS, RIGHT?

YOUR USER REQUESTS YOUR WEB APP

HAS THE PAGE LOADED YET?

"All your users are non-JS while they're downloading your JS" — Jake Archibald

DID THE HTTP REQUEST FOR THE JAVASCRIPT SUCCEED?

If they're on a train and their net connection goes away before your JavaScript loads, then there's no JavaScript.

DID THE HTTP REQUEST FOR THE JAVASCRIPT COMPLETE?

How many times have you had a mobile browser hang forever loading a page and then load it instantly when you refresh?

DOES THE CORPORATE FIREWALL BLOCK JAVASCRIPT?

Because loads of them still do.

DOES THEIR ISP OR MOBILE OPERATOR INTERFERE WITH DOWNLOADED JAVASCRIPT?

Sky accidentally block jQuery, Comcast insert ads into your script, and if you've never experienced this before, drive to an airport and use their wifi.

HAVE THEY SWITCHED OFF JAVASCRIPT?

People still do.

DO THEY HAVE ADDONS OR PLUGINS INSTALLED WHICH INJECT SCRIPT OR ALTER THE DOM IN WAYS YOU DIDN'T ANTICIPATE?

There are thousands of browser extensions. Are you sure none interfere with your JS?

IS THE CDN UP?

CDNs are good at staying up (that's what being a CDN *is*) but a minute downtime a month will still hit users who browse in that minute.

DOES THEIR BROWSER SUPPORT THE JAVASCRIPT YOU'VE WRITTEN?

Check Can I Use for browser usage figures.

IS ALL THE ABOVE TRUE?

THEN YES, YOUR JAVASCRIPT WORKS.

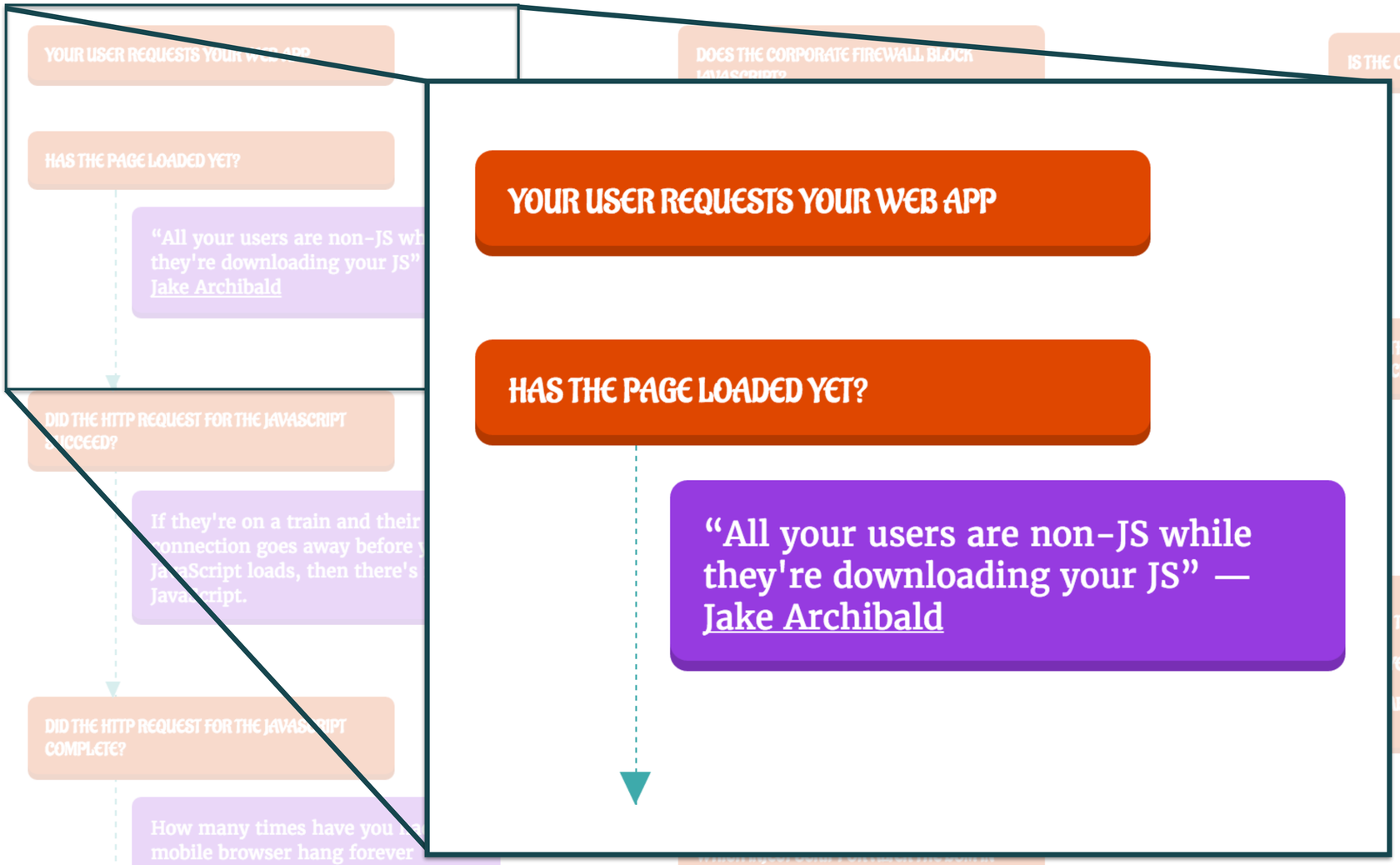
PROBABLY.

Progressive enhancement. Because sometimes your JavaScript just won't work.

Be prepared.

kryogenix.org @sil

EVERYONE HAS JS, RIGHT?



YOUR USER REQUESTS YOUR WEB APP

HAS THE PAGE LOADED YET?

"All your users are non-JS while they're downloading your JS" — Jake Archibald

kryogenix.org @sil

EVERYONE HAS JS, RIGHT?

YOUR USER REQUESTS YOUR WEB APP

HAS THE PAGE LOADED YET?

"All your users are non-JS wh
they're downloading your JS"
Jake Archibald

DID THE HTTP REQUEST FOR THE JAVASCRIPT SUCCEED?

If they're on a train and their
connection goes away before
JavaScript loads, then there's
JavaScript.

DID THE HTTP REQUEST FOR THE JAVASCRIPT COMPLETE?

How many times have you ha
mobile browser hang forever
loading a page and then load it
instantly when you refresh?

DOES THE CORPORATE FIREWALL BLOCK JAVASCRIPT?

DO THEY HAVE ADDONS OR PLUGINS INSTALLED WHICH INJECT SCRIPT OR ALTER THE DOM IN WAYS YOU DIDN'T ANTICIPATE?

There are thousands of browser extensions. Are you sure none interfere with your JS?

WAYS YOU DIDN'T ANTICIPATE?

There are thousands of browser extensions. Are you sure none interfere with your JS?

IS THE CDN UP?

CDNs are good at staying up (that's what being a CDN *is*) but a minute downtime a month will still hit users who browse in that minute.

THEIR BROWSER SUPPORT THE
CRIPT YOU'VE WRITTEN?

Check Can I Use for browser usage figures.

THE ABOVE TRUE?

ES, YOUR JAVASCRIPT WORKS.

BLY.

Progressive enhancement. Because sometimes your JavaScript just won't work.

Be prepared.

kryogenix.org @sil

EVERYONE HAS JS, RIGHT?

YOUR USER REQUESTS YOUR WEB APP

HAS THE PAGE LOADED YET?

"All your users are non-JS wh
they're downloading your JS"
Jake Archibald

DID THE HTTP REQUEST FOR THE JAVASCRIPT SUCCEED?

If they're on a train and their
connection goes away before
JavaScript loads, then there's
JavaScript.

DID THE HTTP REQUEST FOR THE JAVASCRIPT COMPLETE?

How many times have you ha
mobile browser hang forever
loading a page and then load
instantly when you refresh?

DOES THE CORPORATE FIREWALL BLOCK JAVASCRIPT?

DOES THEIR BROWSER SUPPORT THE JAVASCRIPT YOU'VE WRITTEN?

Check Can I Use for browser usage figures.

IS ALL THE ABOVE TRUE?

THEN YES, YOUR JAVASCRIPT WORKS.

PROBABLY.

interfere with your JS?

IS THE CDN UP?

CDNs are good at staying up (that's what being a CDN *is*) but a minute downtime a mo h will still hit users who browse in that minute.

HEIR BROWSER SUPPORT THE
RIPT YOU'VE WRITTEN?

Check Can I Use for browser usage figures.

THE ABOVE TRUE?

ES, YOUR JAVASCRIPT WORKS.

BLY.

Progressive enhancement. Because sometimes your JavaScript just won't work.

Be prepared.

kryogenix.org @sil

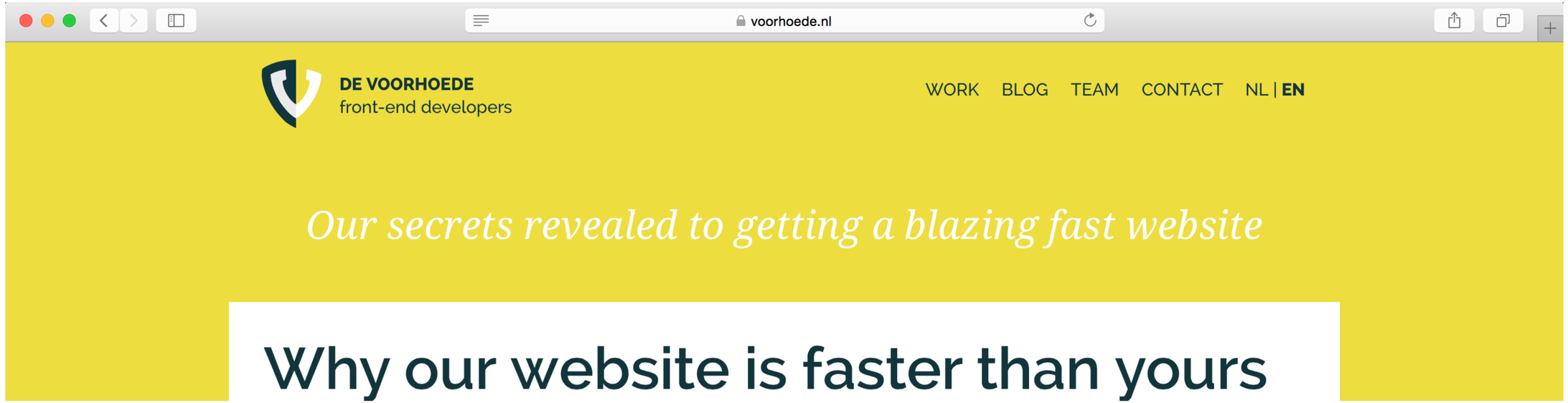
whatever language or
(JS) framework you use
start by serving HTML

@jbmoelker

ENHANCING REACT APPS

@jbmoelker

OPTIMISE FOR CRITICAL PATH



by Declan
July 26, 2016

Table of contents

We recently updated our site. Yes, it has a complete design overhaul, but as real software developers we focused a lot on the technical bits and pieces as well. Our goal was to take control, focus on performance, be flexible for the future and make it fun to write content for the site. Here's how we made our website faster than yours (Yup, sorry!)

Design for performance

In our projects we have daily discussions with designers and product owners about balancing aesthetics and performance. For our own site, this was easy. Simply said: we believe that a good user experience starts with delivering content as fast as possible. That means **performance > aesthetics**.

Good content, layout, images, and interactivity are essential for engaging your audience, but each of these elements have impact on page load time and the end-user experience. In every step we looked at how we could get

post on voorhoede.nl

ISOMORPHIC

"a seamless transition
from server-rendered HTML
to client-rendered DOM
without losing state"

REACT ROUTER

React Server

React Starter Kit

React Isomorphic Form

SMALLER MOUNT NODES

```
<body>
 <div id="megaApp">
 </div>
</body>


ReactDOM.render(
  <App/>,
  getById('megaApp')
);
```

```
<body>
  <h1>Static title</h1>
  <div id="miniApp1">
  </div>
  static content
  <div id="miniApp2">
  </div>
  more static content
</body>
```

USEFUL INITIAL CONTENT

<**div id="map"** data-address="Sieraad, Amsterdam">

```
const map = document.getElementById('map');
const address = map.dataset.address;
ReactDOM.render(
  <Map address={ address } />,
  map
)
```

USEFUL INITIAL CONTENT

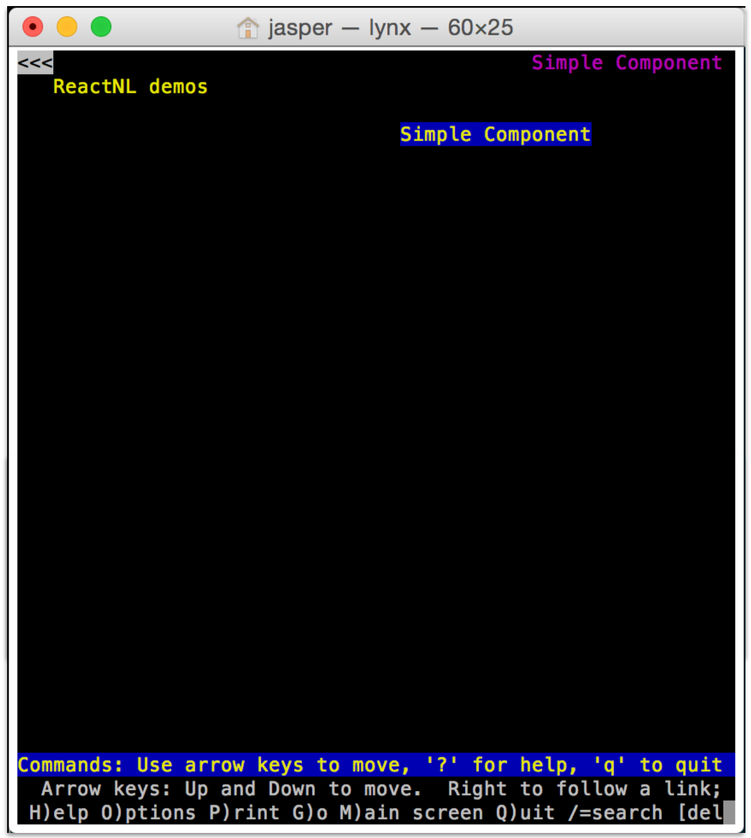
~~ ~~
~~ ~~

<**a href**="https://**google.com/maps**/place/Sieraad,Amsterdam">
<**img src**="https://maps.googleapis.com/maps/api/**staticmap**?
center=Sieraad,Amsterdam&zoom=17&size=400x300">

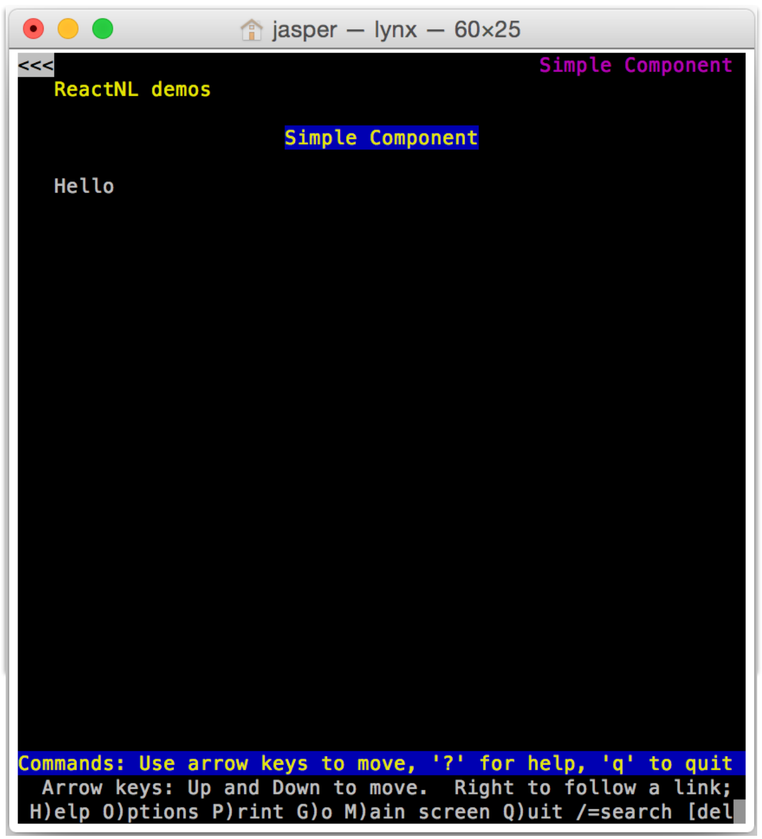
ENHANCED EXAMPLES

EXAMPLE: SIMPLE

LYNX original



LYNX enhanced




```
<div id="mount">
</div>
```

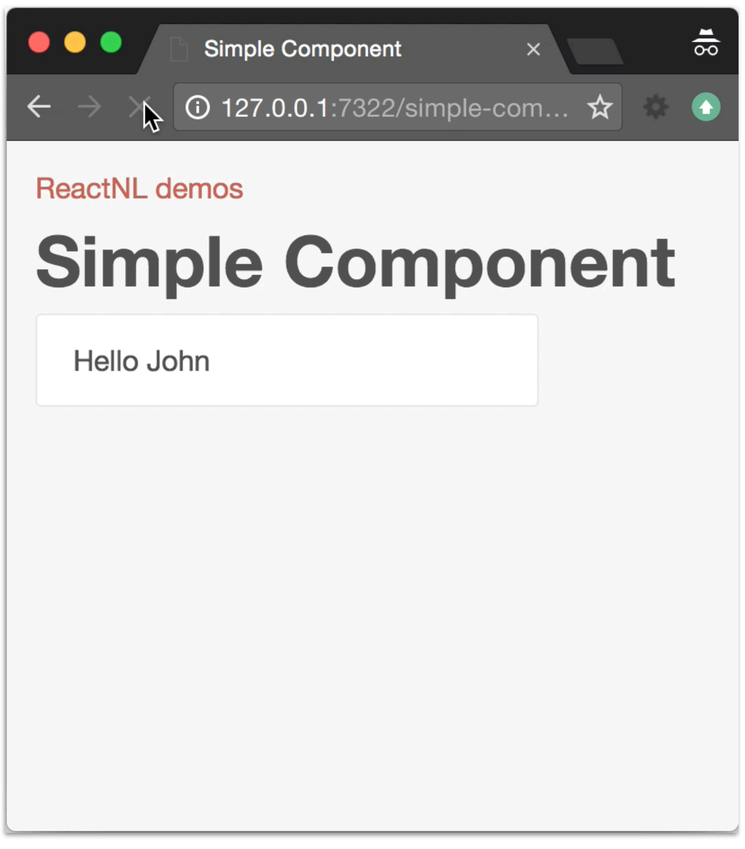
↓

```
<div id="mount">
  <p>Hello</p>
</div>
```

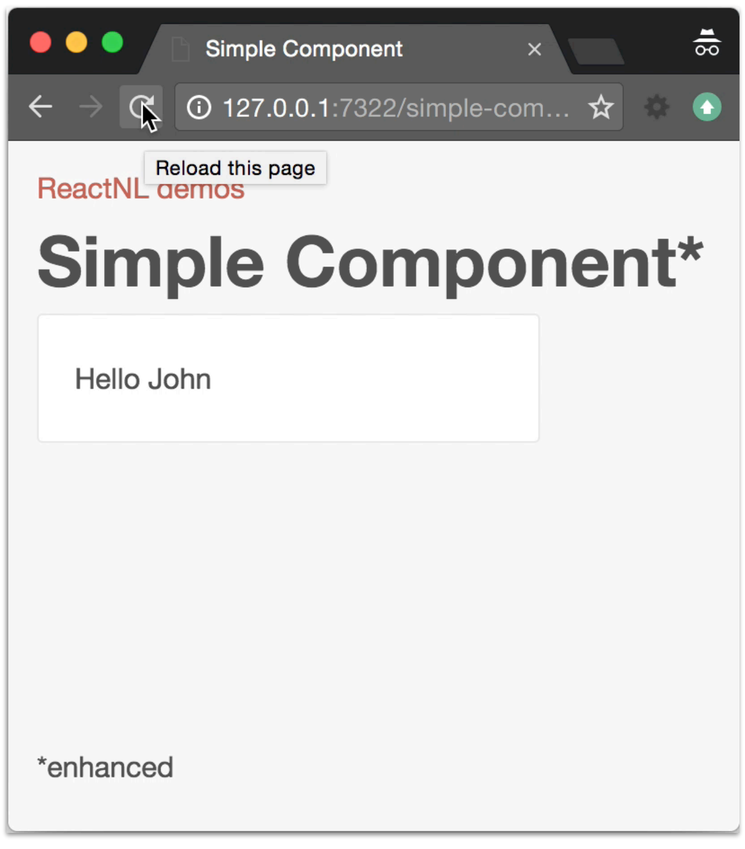
jbmoelker.github.io/reactnl-demos/simple-component-enhanced

EXAMPLE: SIMPLE

original



enhanced




```
<div id="mount">
</div>
```

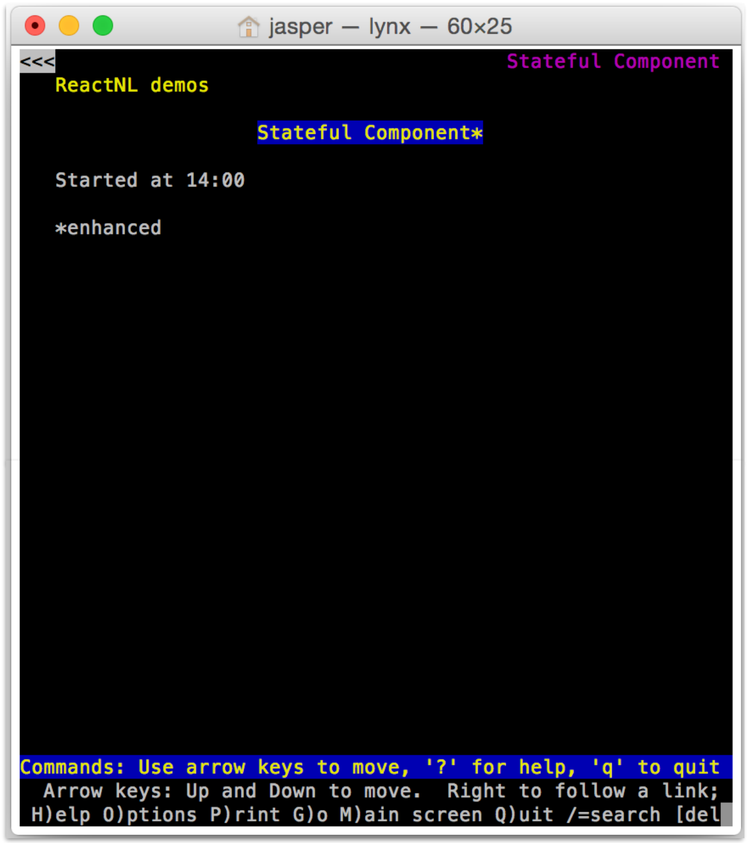
↓

```
<div id="mount">
  <p>Hello</p>
</div>
```

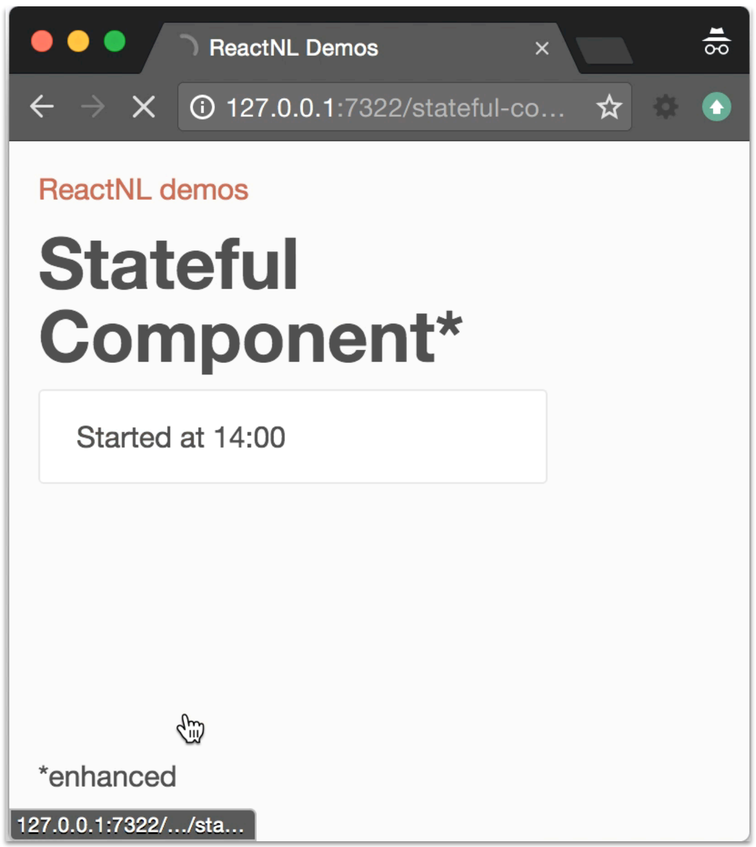
jbmoelker.github.io/reactnl-demos/simple-component-enhanced

EXAMPLE: STATEFUL

LYNX enhanced



enhanced




```
<div id="mount">
</div>
```

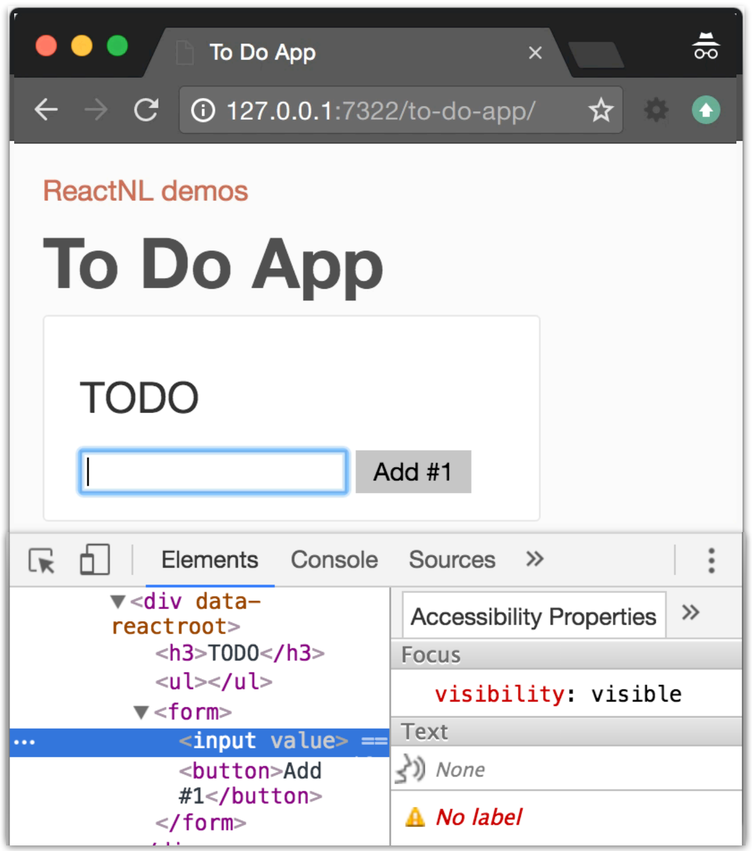
↓

```
<time id="mount"
  datetime="2016-
  10-13T14:00">
 Started at 14:00
</time>
```

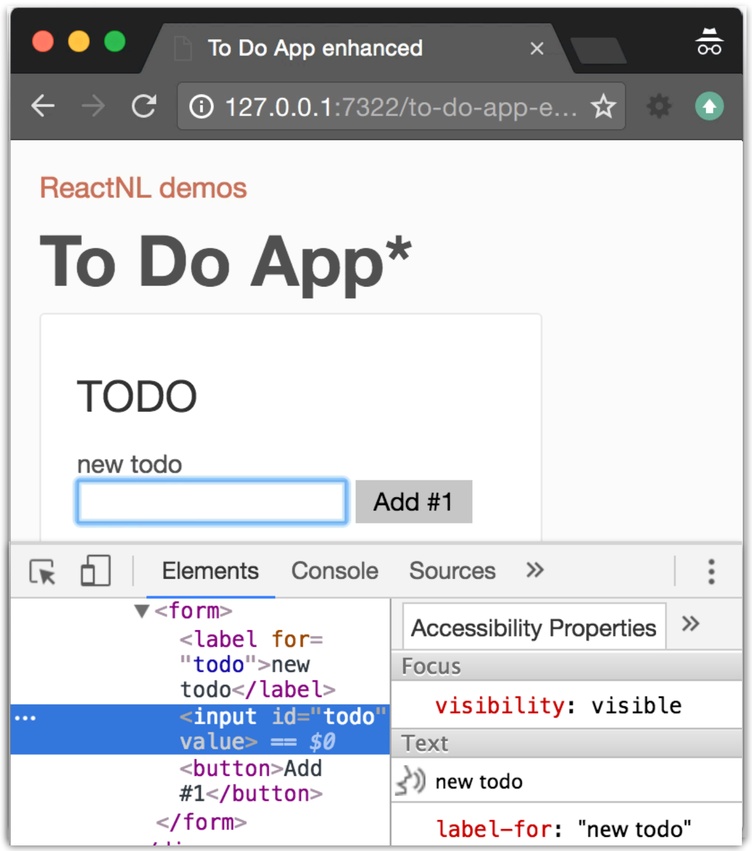
jbmoelker.github.io/reactnl-demos/stateful-component-enhanced

EXAMPLE: TO DO APP

original



enhanced



<input id="todo">

↓

<label for="todo">

new todo

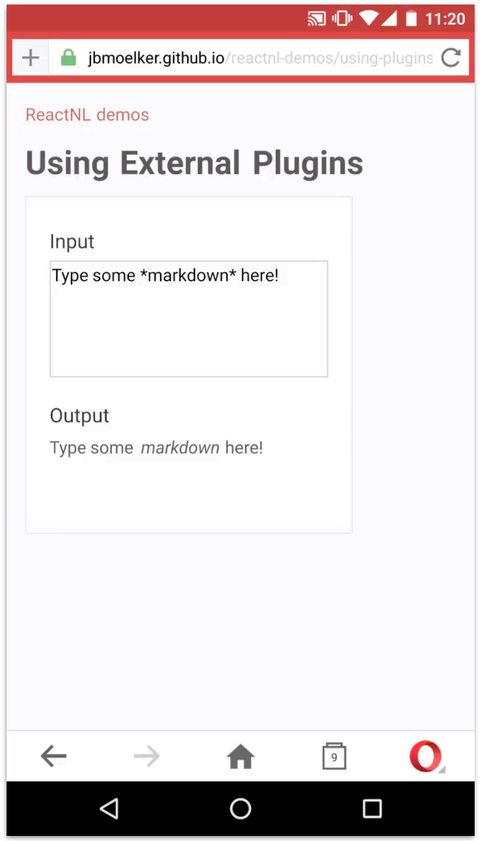
</label>

<input id="todo">

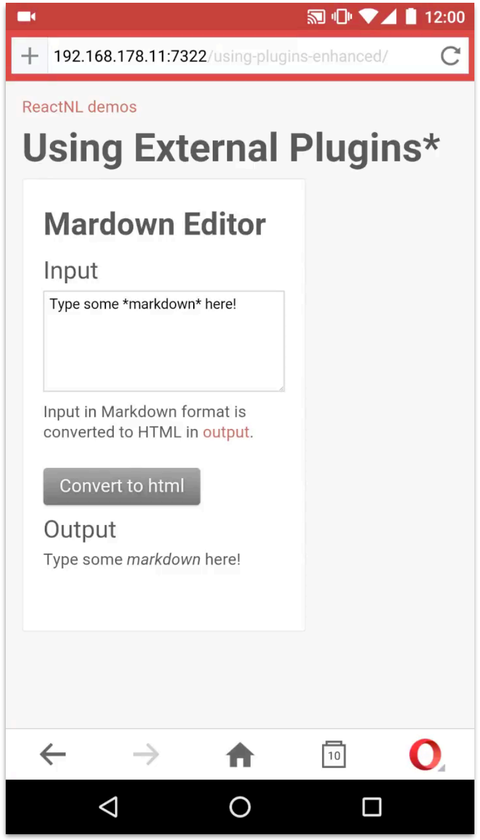
jbmoelker.github.io/reactnl-demos/to-do-app-enhanced

EXAMPLE: USING PLUGINS

original



enhanced




```
<div class="MarkdownEditor">
```

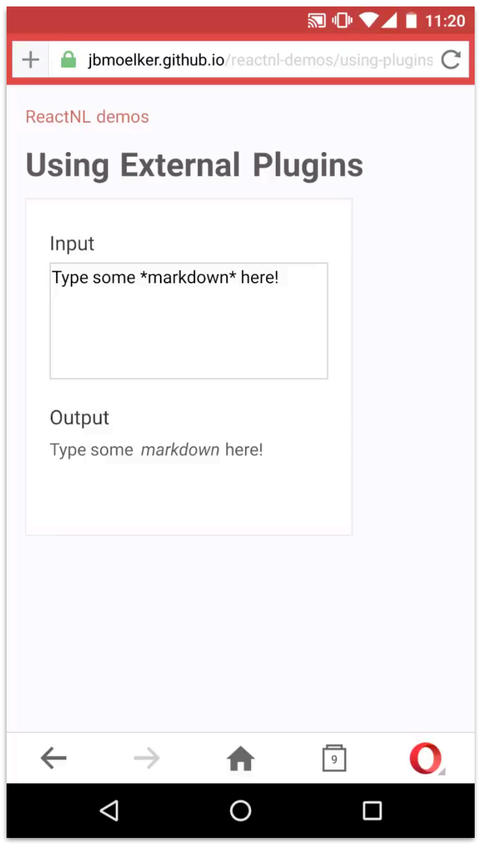
↓

```
<form method="..." action="..."
class="MarkdownEditor">
<h2>Markdown Editor</h2>
...
<button>Convert</button>
```

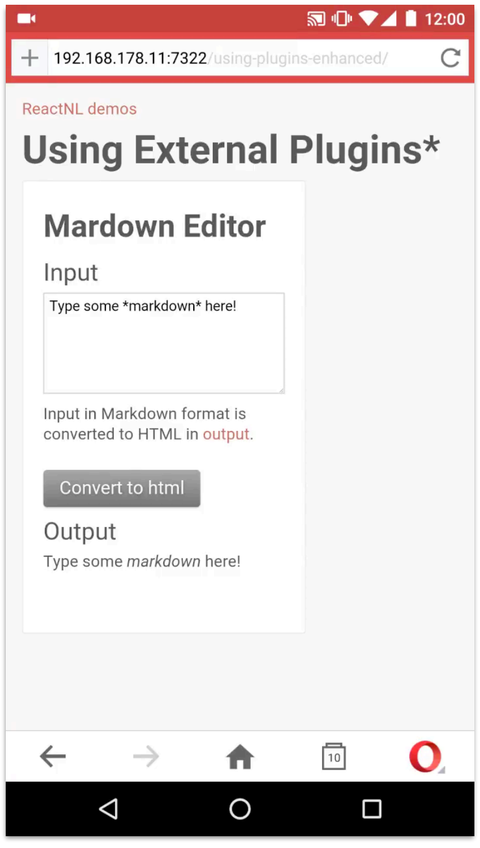
jbmoelker.github.io/reactnl-demos/using-plugins-enhanced

EXAMPLE: USING PLUGINS

original



enhanced




```
<h3>Input</h3>
<textarea …>…</textarea>
```

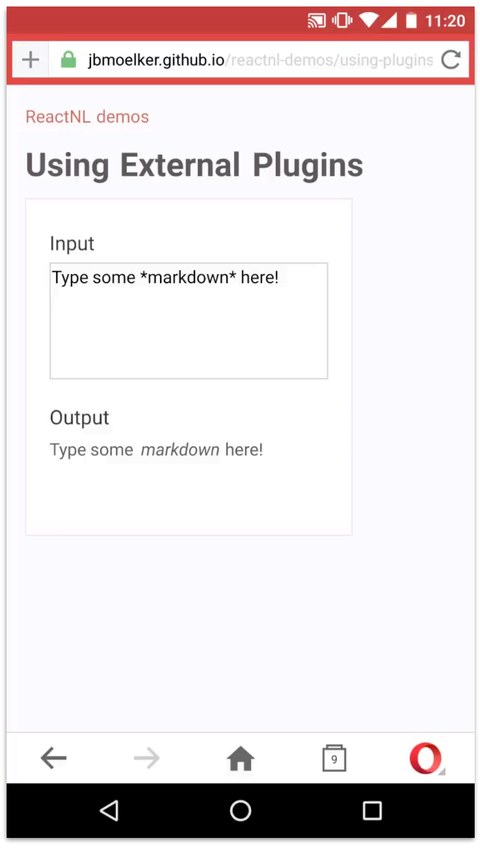
↓

```
<label for="input" class="h3">
Input</label>
<textarea id="input" …>
…</textarea>
```

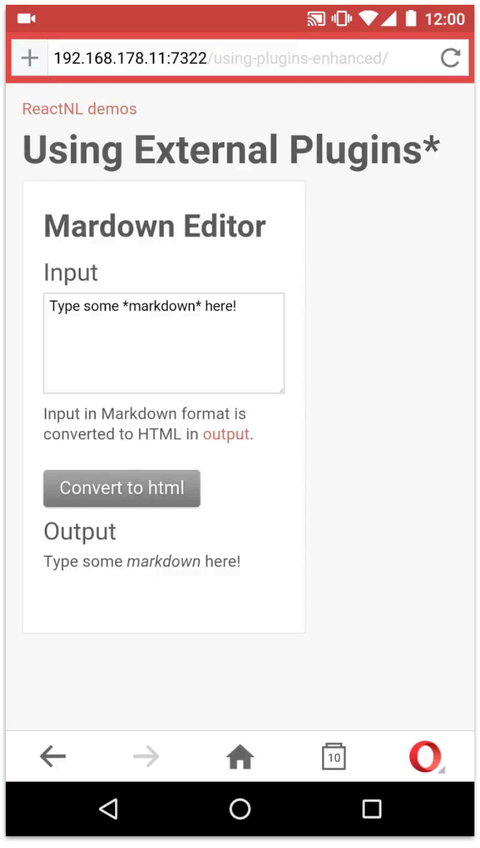
jbmoelker.github.io/reactnl-demos/using-plugins-enhanced

EXAMPLE: USING PLUGINS

original



enhanced




```
<h3>Output</h3>
<div> … </div>
```

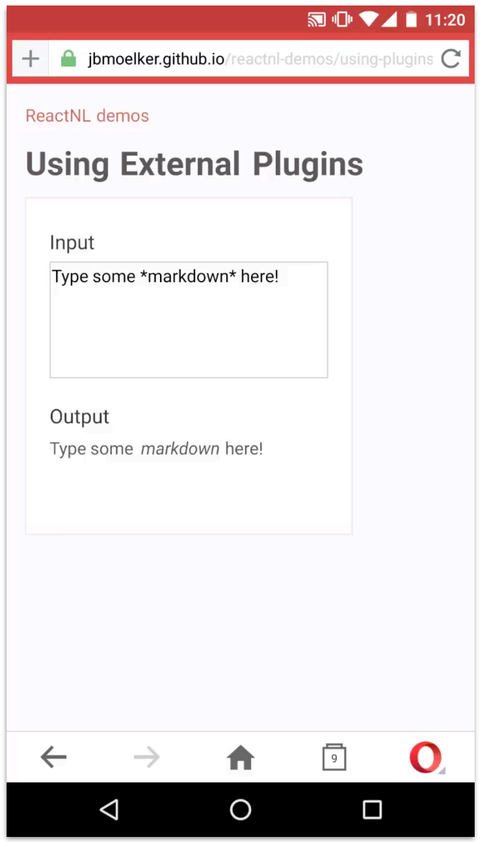
↓

```
<label for="output"
   class="h3">Output</label>
<output id="output"
   tabindex="-1"
   for="input"> … </output>
```

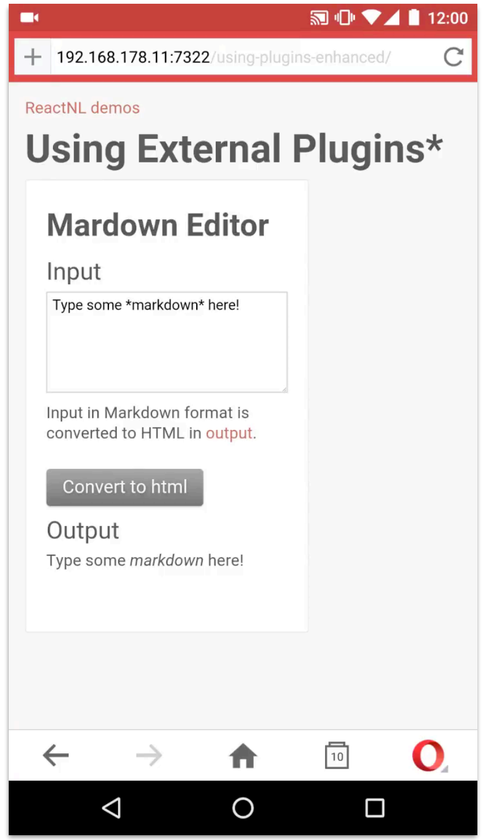
jbmoelker.github.io/reactnl-demos/using-plugins-enhanced

EXAMPLE: USING PLUGINS

original



enhanced




```
<textarea id="input"
  aria-describedby="tooltip">
</textarea>
<p role="tooltip" id="tooltip">
  Input in Markdown format
  is converted to HTML in
  <a href="#output">output
</a></p>
```

jbmoelker.github.io/reactnl-demos/using-plugins-enhanced

IN SUMMARY

@jbmoelker

think about **users & browsers**

serve useful **initial HTML**

use HTML's **built-in features**

@jbmoelker

THANK YOU!

@jbmoelker

☐ questions?

@jbmoelker