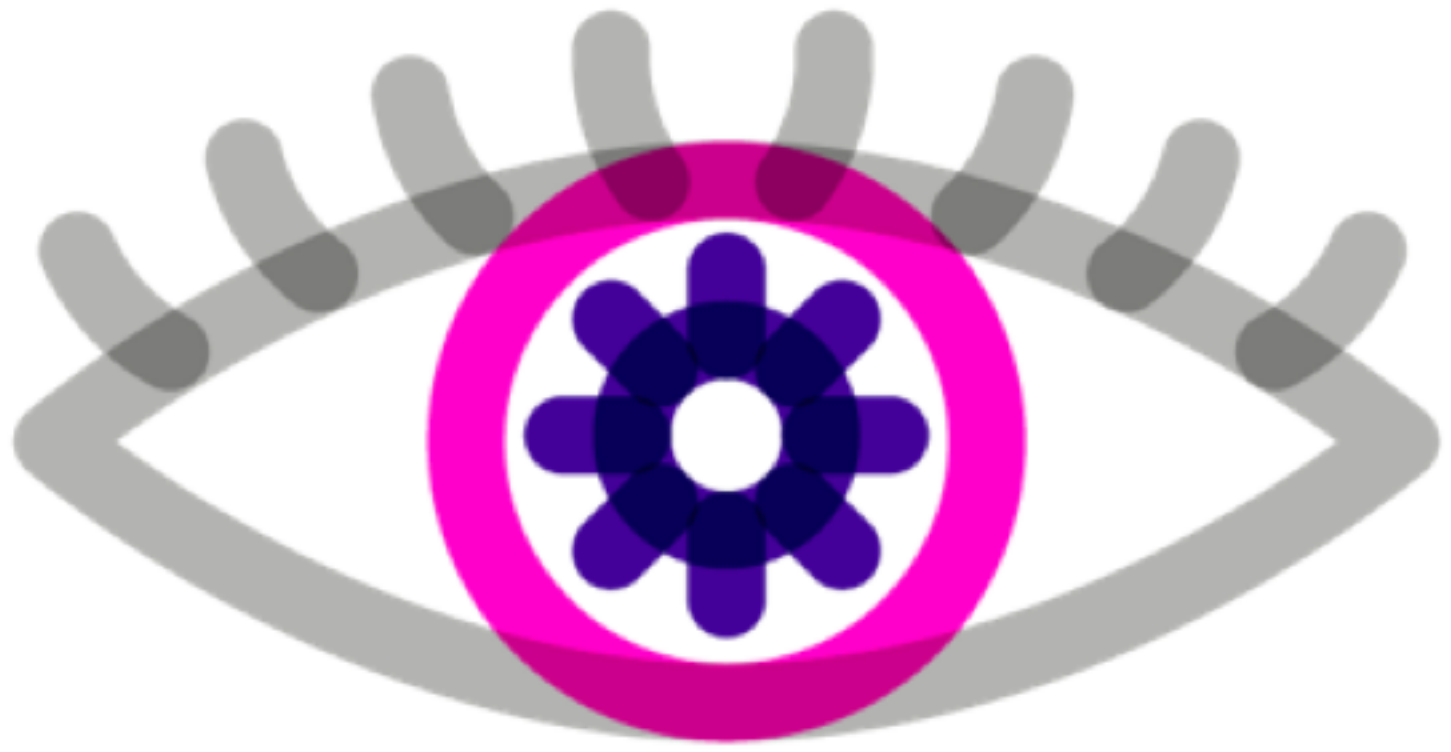
Mastering Server Rendered React

React Amsterdam
16 april 2016

Sven Anders Robbestad

This is me

I work at Inmeta consulting



I wrote this book

http://bit.ly/reactjsbook

Community Experience Distilled

ReactJS Blueprints

Create powerful applications with ReactJS, the most popular platform for web developers today

Sven Anders Robbestad

[PACKT] open source* community experience distilled PUBLISHING

Server-rendered apps

You may know it as **isomorphic**

...or you may know it as **Universal**

But what is it?

It's about **sharing** code between your server code and your browser code

It's also about delivering content **faster**

You might as well call it writing **Shared** code.

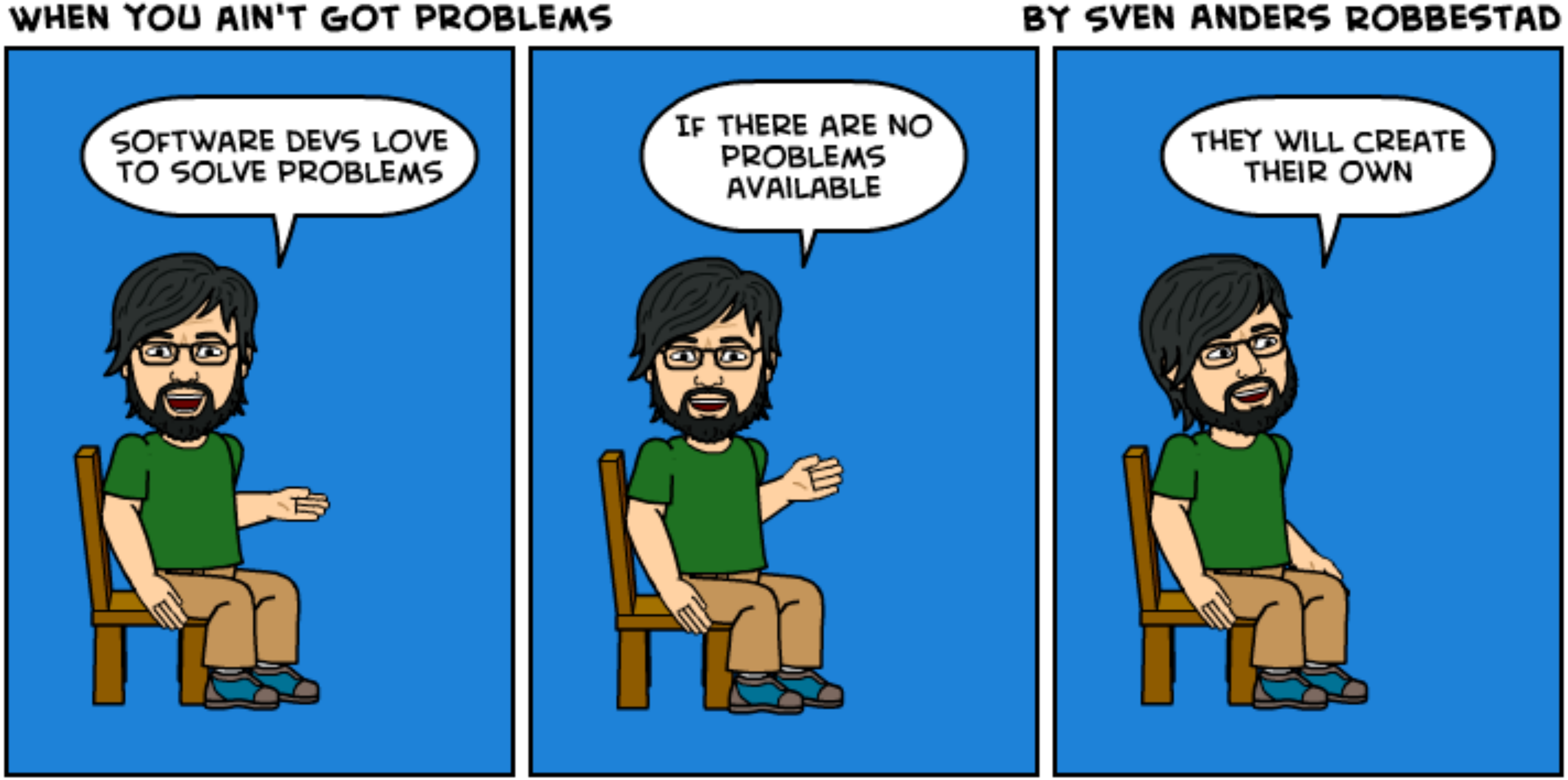
Server-rendered apps

A bit of history

PHP - smarty, twig etc...

Java - jsp, thymeleaf etc...

JavaScript - express/swig, ejs etc...
but what about React?


WHEN YOU AIN'T GOT PROBLEMS
BY SVEN ANDERS ROBBESTAD
SOFTWARE DEVS LOVE TO SOLVE PROBLEMS
IF THERE ARE NO PROBLEMS AVAILABLE
THEY WILL CREATE THEIR OWN

When does a user leave your site before it's even loaded?

Answer:

When your site is **slow**

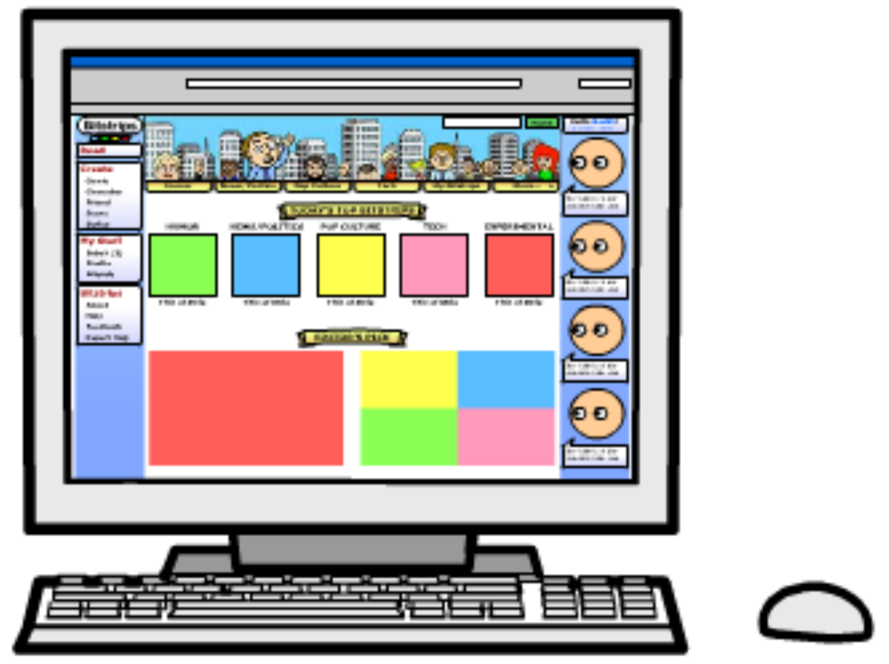
Page load is critical

Every delay make users go away

So when is JavaScript **slow**?

It's slow when...

...the user is on a **slow computer**



...the user has a **slow connection**

...the user is on a slow **smart phone** or tablet

...the user is **stuck** on a computer they don't control (school, library).

...the user trying to **download** your site to read away from an Internet connection

...that someone is **Google** cache or the **Internet archive**

...the user is using **NoScript** and visits your site

...the user is using using adblock and you've named a critical JS bundle with something related to ads

Good news!

ALL of these problems can be solved with server-rendering

Rendering your content before you serve them makes:

Google happy

Slow computers happy

Smart phones happy

NoScript happy

Best of all

It makes **you** happy

Let's make it happen

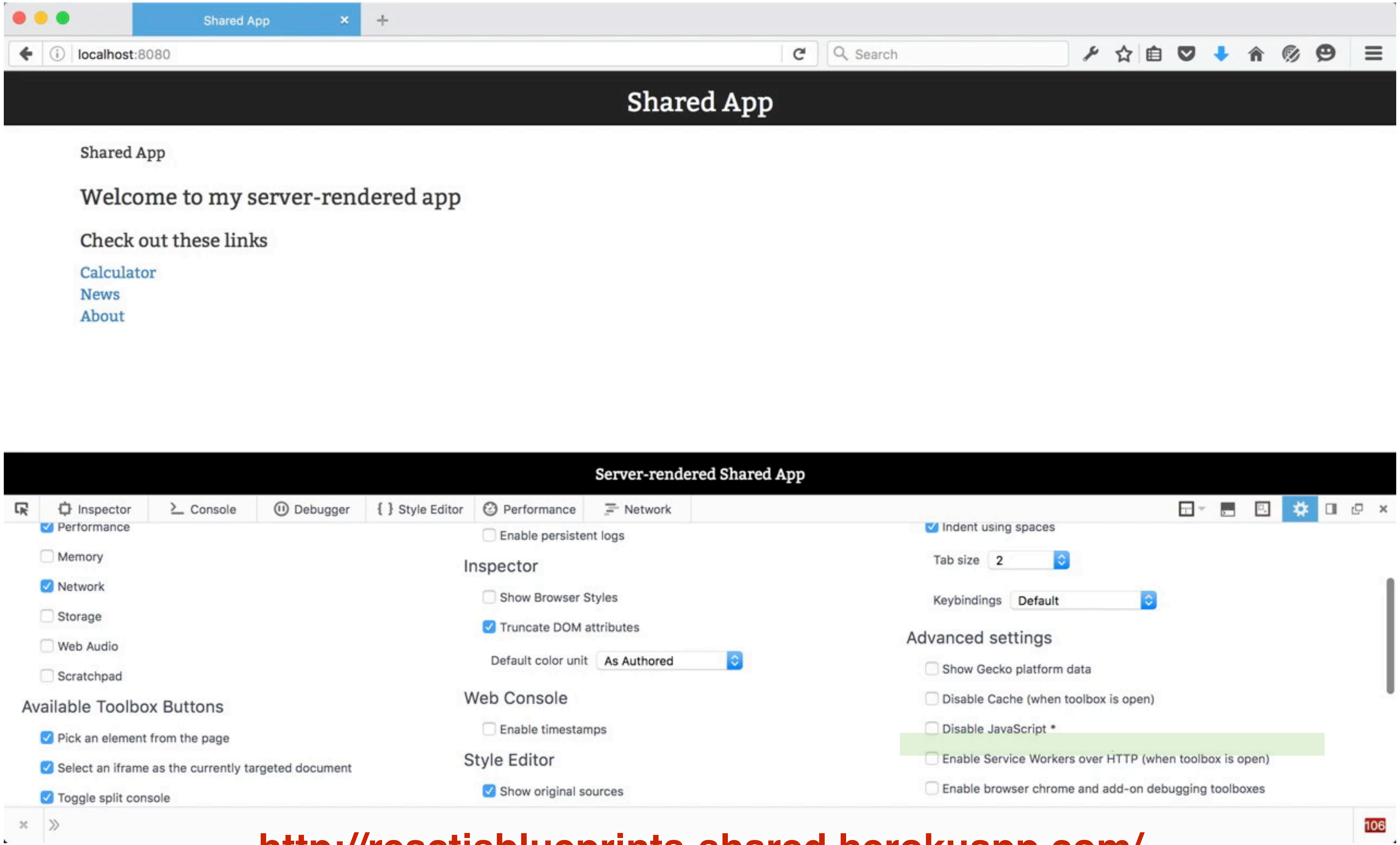
Goals

Write code that you can use everywhere

"Render" your app so that your client doesn't have to

Let the dynamic bundle take over for interactivity

This is the behavior we want



Shared App

Shared App

Welcome to my server-rendered app

Check out these links

Calculator
News
About

Server-rendered Shared App

- [✓] Performance
- [ ] Memory
- [✓] Network
- [ ] Storage
- [ ] Web Audio
- [ ] Scratchpad

Available Toolbox Buttons

- [✓] Pick an element from the page
- [✓] Select an iframe as the currently targeted document
- [✓] Toggle split console

- [ ] Enable persistent logs

Inspector

- [ ] Show Browser Styles
- [✓] Truncate DOM attributes

Default color unit: As Authored

Web Console

- [ ] Enable timestamps

Style Editor

- [✓] Show original sources

- [✓] Indent using spaces

Tab size: 2

Keybindings: Default

Advanced settings

- [ ] Show Gecko platform data
- [ ] Disable Cache (when toolbox is open)
- [ ] Disable JavaScript *
- [ ] Enable Service Workers over HTTP (when toolbox is open)
- [ ] Enable browser chrome and add-on debugging toolboxes

http://reactjsblueprints-shared.herokuapp.com/

www.browseo.net/?url=http%3A%2F%2Freactjsblueprints-shared.herokuapp.com%2Fnews
Search
Yes, we are for hire! Contact us today...
BROWSEO
Blog | About | FAQ | Contact
http://reactjsblueprints-shared.herokuapp.com/new
Browse
Highlight links
internal
external
nofollow

Shared App

Shared App > News

Last updated at 9:10:54 AM.

- Opening DOCX Files on a Mac, Without Microsoft Office
- How to Enable Wi-Fi Calling on iPhone
- 8 New Apple Watch Commercials and 2 Apple TV Ads Now Airing
- How to Remove a Disk from Time Machine on Mac
- Check Packages for Expired Certificates in Mac OS X
- How to Use “Hey Siri” on Apple Watch
- How to View Unread Email Only in Mail on iPhone & iPad
- How to Add Email Attachments in Mail for iPhone & iPad
- Play Playstation 4 Games on Mac (or Windows) with PS4 Remote Play
- Make Animated GIFs from Movies in Mac OS X with Drag & Drop Ease
- How to Loop Video with QuickTime Player on Mac OS X
- Enable & Disable Night Shift in iOS Quickly from Control Center
- How to Show Full Email Headers in Mail for Mac OS X
- First Betas of iOS 9.3.2, OS X 10.11.5, WatchOS 2.2.1, tvOS 9.2.1 Available for Testing
- How to Password Lock Notes on iPhone & iPad

Server-rendered Shared App

Response

Response-Code	200 (OK)

Text Information

Words	158
Internal Links	1
External Links	0
Nofollow Links	0

Head

title	Shared App (10 Characters)
viewport	width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no

SERP Preview

Shared App
http://reactjsblueprints-share... - Cached
There is no META description currently available.

Headings

H1	Shared App

Make a node/express server

```
app.get('*', function(req, res) {
  res.sendFile(path.join(__dirname, 'assets', req.path));
});


app.listen(port, 'localhost', function(err) {
  console.log('Server up at http://localhost:' + port);
});
```

Import routes

```
import { routes } from './source/routes';
```

Import your API method(s)

```
import { fetchPostsAsync } from './source/shared/api/fetch-posts'
```

Perform fetch on route change

```
const appRoutes = (app) => {
  app.get('*', (req, res) => {
    match({ routes, location: req.url }, (err, redirectLocation, props) => {
      if (err) {
        res.status(500).send(err.message);
      } else if (redirectLocation) {
        res.redirect(302, redirectLocation.pathname + redirectLocation.search);
      } else if (props) {
        fetchPostsAsync((posts) => {
          const isFetching = false;
          const lastUpdated = Date.now()
          const initialState = {
            posts,
            isFetching,
            lastUpdated
          }
          const store = configureStore(initialState)
          const app = ReactDOMServer.renderToString(
            <Provider store={store}>
              <RoutingContext {...props} />
            </Provider>);
          res.send(`<!DOCTYPE html>
          <html>
            <head>
              <meta charSet="utf-8" />
              <meta httpEquiv="X-UA-Compatible" content="IE=edge" />
              <meta name="viewport" content="width=device-width,
                initial-scale=1, maximum-scale=1, user-scalable=no"/>
              <link async rel="stylesheet" type="text/css"
                href="//maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css"/>
              <link async rel="stylesheet" type="text/css"
                href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css" />
              <link async href='https://fonts.googleapis.com/css?family=Bitter'
              rel='stylesheet' type='text/css'/>
              <link async rel="stylesheet" href="/app.css" />
                <title>${settings.title}</title>
                </head>
                 <script>
                  window.__INITIAL_STATE__ = ${JSON.stringify(initialState)}
                 </script>
                <body><div id="app">${app}</div><script src="/bundle.js"></script></body></html>`);
        })
      }
    })
  })
}
```

Advantages

Less work on the client = faster render

Reuse code from the frontend

App works even if your JS bundle breaks

And if the user has NoScript

Or is a bot like Google and Internet Archive

Drawbacks

Need to write a complex server file

Need to rely on a fetch method before render

Relatively slow due to lack of optimisations like cache

Add streaming to make it faster

```
import ReactDOMStream from 'react-dom-stream/server';
```

```
  res.write(`<!DOCTYPE html>
    <html>
      <head>
        <meta charSet="utf-8" />
        <meta httpEquiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width,
          initial-scale=1, maximum-scale=1, user-scalable=no"/>
        <link async rel="stylesheet" type="text/css"
          href="//maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css"/>
        <link async rel="stylesheet" type="text/css"
          href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css" />
        <link async href='https://fonts.googleapis.com/css?family=Bitter'
        rel='stylesheet' type='text/css'/>
        <link async rel="stylesheet" href="/app.css" />
          <title>${settings.title}</title>
          </head>
           <script>
            window.__INITIAL_STATE__ = ${JSON.stringify(initialState)}
           </script>
          <body><div id="app">`);
           const stream = ReactDOMStream.renderToString(
           <Provider store={store}>
             <RoutingContext {...props} />
           </Provider>);
           stream.pipe(res, {end: false});
           stream.on("end", ()=> {
               res.write(`</div><script src="/bundle.js"></script></body></html>`);
               res.end();
           });''
         })
```

Resources, Q&A

- http://bit.ly/isomorphicreact
- Out of the box solution: fluxible.io
- http://bit.ly/reactjsbook

My twitter: **#svenardocom**

Tweet from the conference with the hashtag **#reactamsterdam**

Community Experience Distilled
ReactJS Blueprints
Create powerful applications with ReactJS, the most popular platform for web developers today
Sven Anders Robbestad
[PACKT] PUBLISHING
open source
community experience distilled