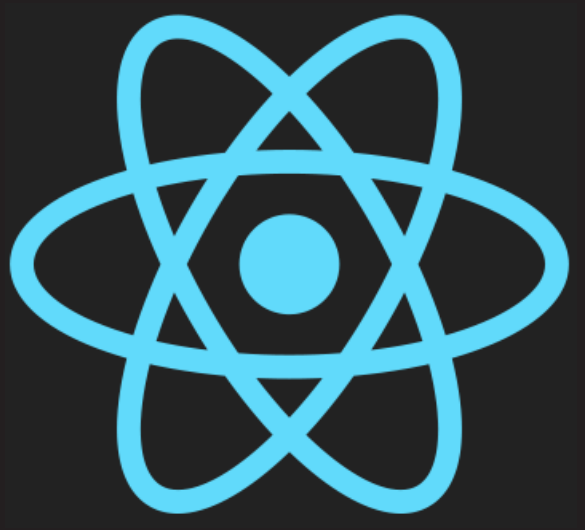
Isomorphic React

Server-Side Rendering

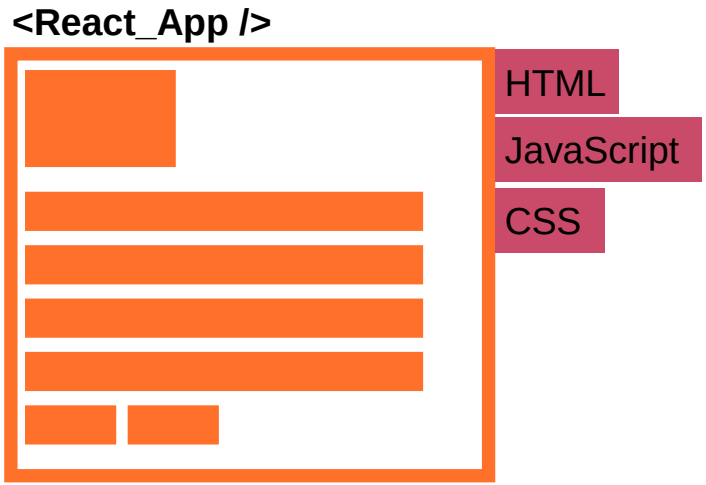


LEARNING OBJECTIVES

- Server-Side Rendering SPAs, also known as Isomorphic Apps
- Understanding the difference between Client Side Rendered & Server Side Rendered applications
- Benefits of Server-Side Rendering
- SSR with React - Concepts

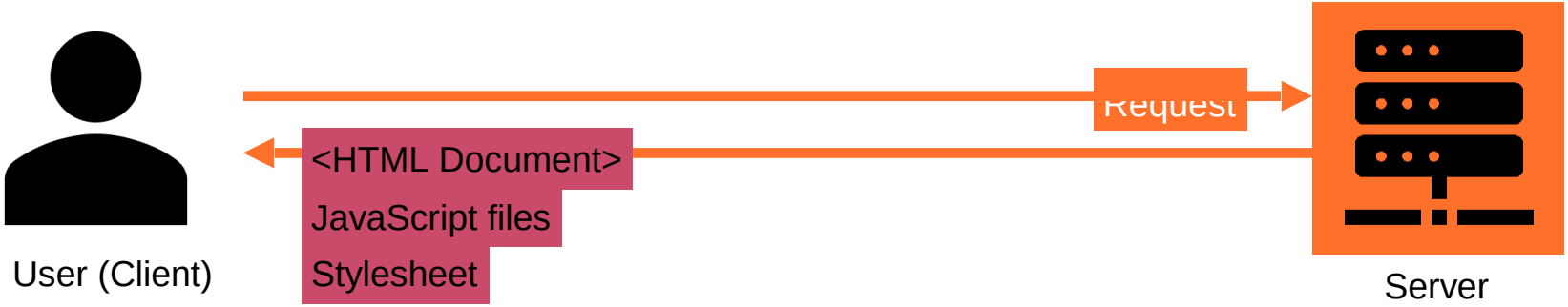
SEO with Client Side Rendered Apps

A TYPICAL REACT APP – CLIENT SIDE RENDERED



React apps are Client Side Rendered apps

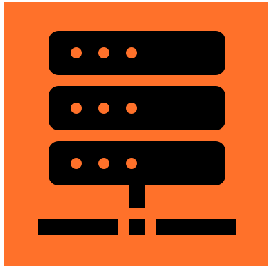
A TYPICAL REACT APP – CLIENT SIDE RENDERED



A Request sent to the server is responded with an HTML document, JavaScript files & stylesheets

A TYPICAL REACT APP – CLIENT SIDE RENDERED

User (Client)



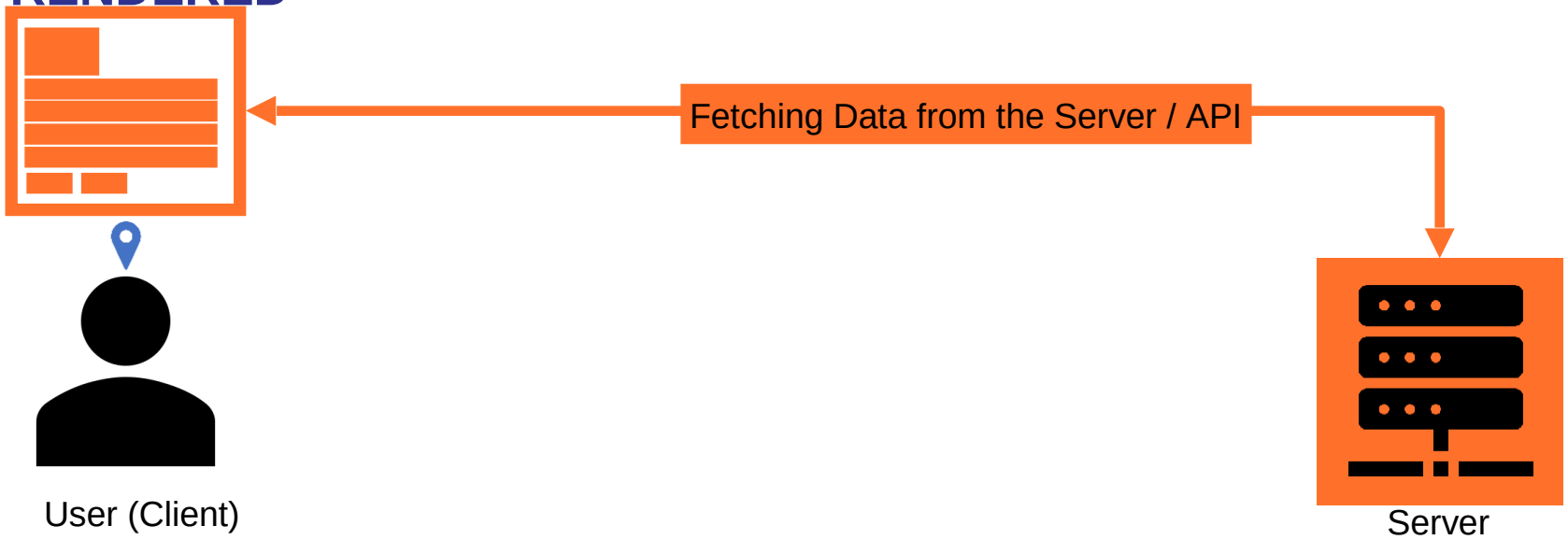
Server

The App renders in the client's browser

A TYPICAL REACT APP – CLIENT SIDE RENDERED

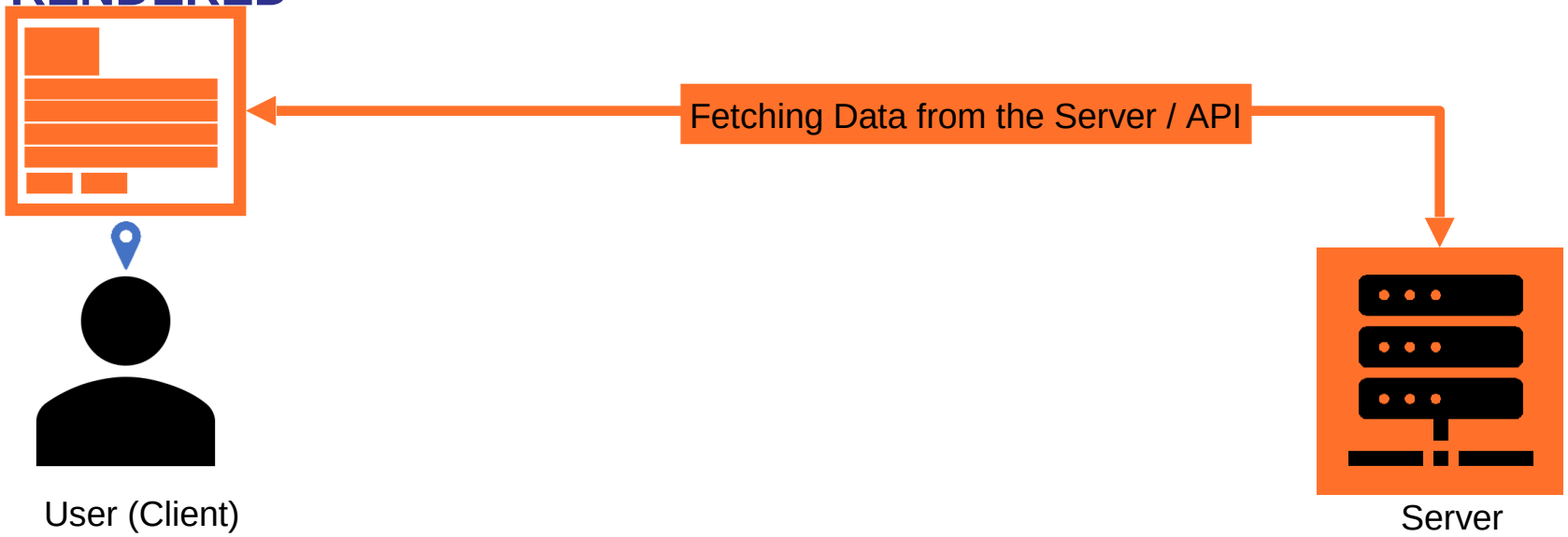
The app requests data as and when needed

A TYPICAL REACT APP – CLIENT SIDE RENDERED



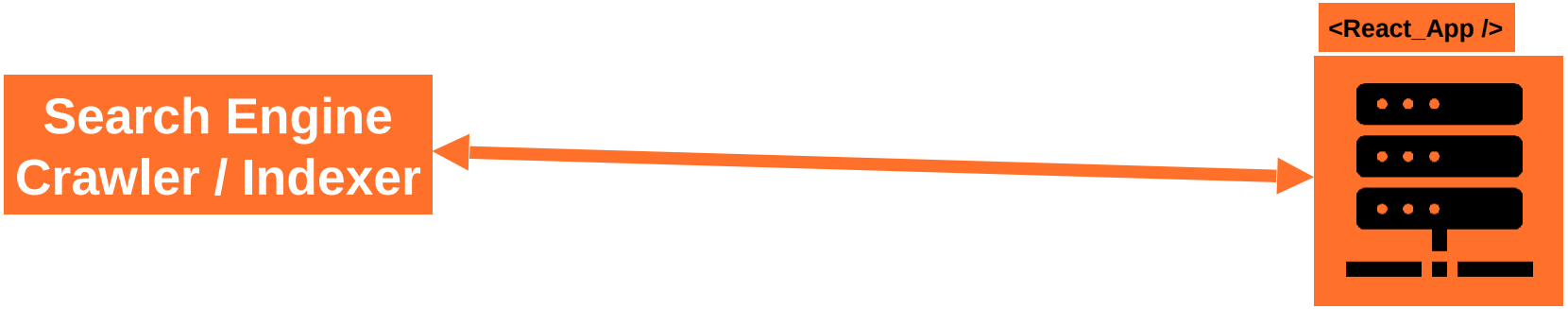
- √ App runs completely in the browser
- √ Offers a snappy user experience

A TYPICAL REACT APP – CLIENT SIDE RENDERED



- ✅ App runs completely in the browser
- ✅ Offers a snappy user experience
- ❌ You have to wait for the JavaScript app bundle to download before it executes.
- ❌ On slow networks, you will often see a blank white page until the app loads up

SEO WITH CLIENT SIDE RENDERED APPS



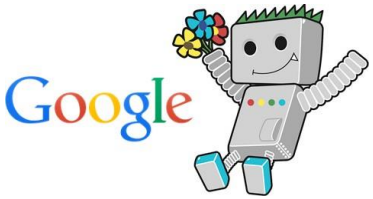
SEO WITH CLIENT SIDE RENDERED APPS

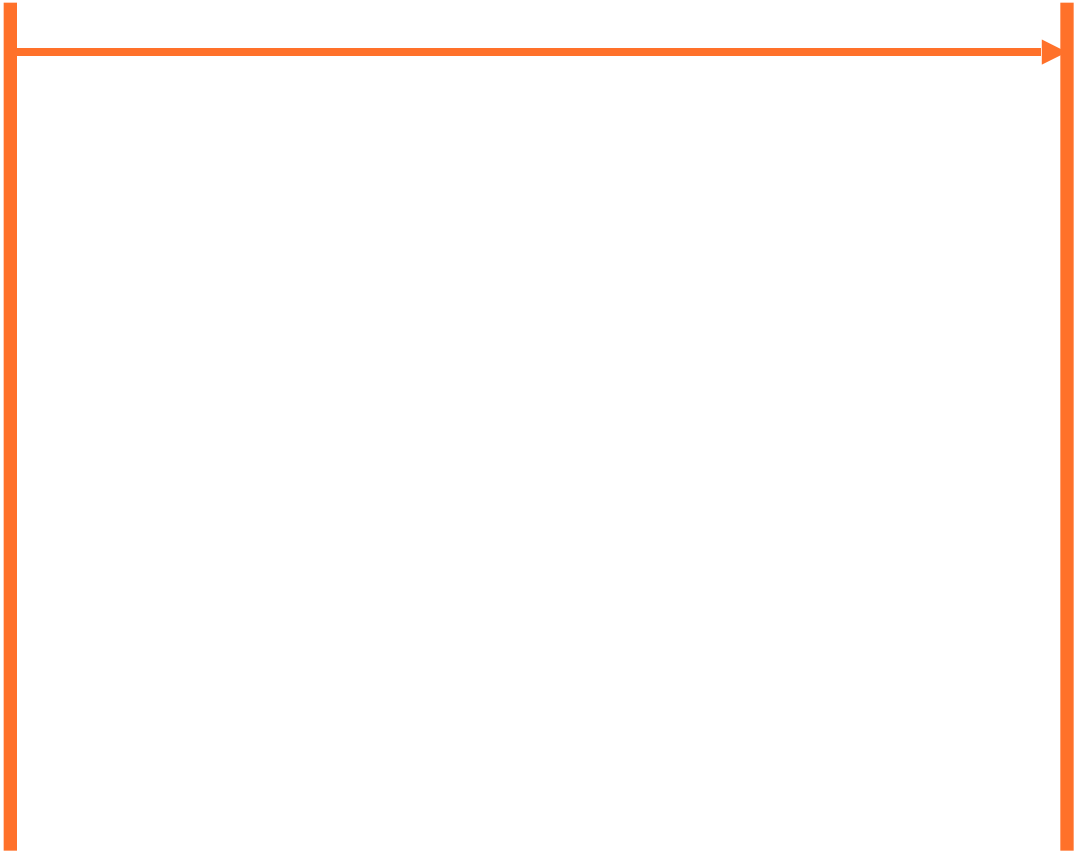
Search Engine Crawler / Indexer

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>

  <script src="/static/js/runtime.js"></script>
  <script src="/s
  <script src="/static/js/main.chunk.js"></script></body>
</html>
```

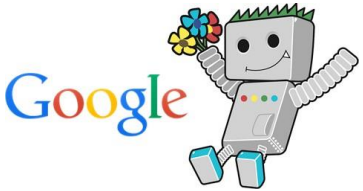
The HTML response for a client-side rendered app contains no actual content to index

GOOGLEBOT CRAWLING A REACT APP





GOOGLEBOT CRAWLING A REACT APP

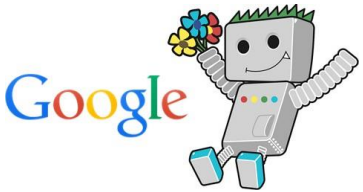



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this
app.</noscript>
    <div id="root"></div>

  <script src="/static/js/bundle.js"></script>
  <script src="/static/js/0.chunk.js"></script>
  <script src="/static/js/main.chunk.js"></script></
body>
</html>
```

HTML

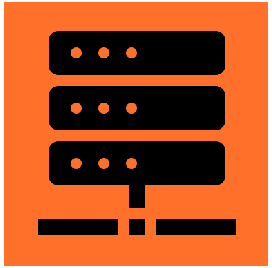
GOOGLEBOT CRAWLING A REACT APP

Google recognizes the app as a JavaScript SPA



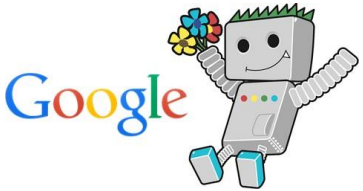

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this
app.</noscript>
    <div id="root"></div>

  <script src="/static/js/bundle.js"></script>
  <script src="/static/js/0.chunk.js"></script>
  <script src="/static/js/main.chunk.js"></script></
body>
</html>
```

HTML



GOOGLEBOT CRAWLING A REACT APP

Google recognizes the app as a JavaScript SPA
Defers the execution of the SPA to a later time

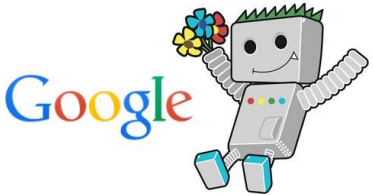



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this
app.</noscript>
    <div id="root"></div>

  <script src="/static/js/bundle.js"></script>
  <script src="/static/js/0.chunk.js"></script>
  <script src="/static/js/main.chunk.js"></script></
body>
</html>
```

HTML

GOOGLEBOT CRAWLING A REACT APP

- Google recognizes the app as a JavaScript SPA
- Defers the execution of the SPA to a later time
- Delay can run into days

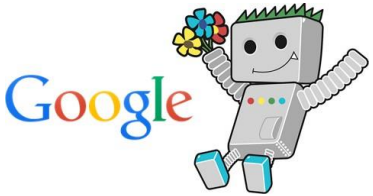



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this
app.</noscript>
    <div id="root"></div>

  <script src="/static/js/bundle.js"></script>
  <script src="/static/js/0.chunk.js"></script>
  <script src="/static/js/main.chunk.js"></script></
body>
</html>
```

HTML

GOOGLEBOT CRAWLING A REACT APP

- Google recognizes the app as a JavaScript SPA
- Defers the execution of the SPA to a later time
- Delay can run into days
- Dynamic content heavy apps cannot be indexed this way

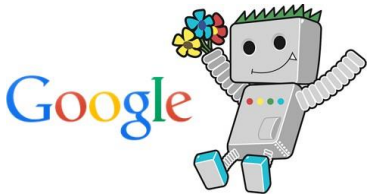



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this
app.</noscript>
    <div id="root"></div>

  <script src="/static/js/bundle.js"></script>
  <script src="/static/js/0.chunk.js"></script>
  <script src="/static/js/main.chunk.js"></script></
body>
</html>
```

HTML

SEO with Server Side Rendered Apps

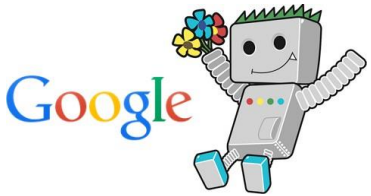
SERVER SIDE RENDERED CONTENT




```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root">
      <div class="list">
        <div class="item">Apples</div>
        <div class="item">Kiwi</div>
        <div class="item">Mango</div>
        <div class="item">Strawberry</div>
      </div>
    </div>

  <script src="/static/js/bundle.js"></script>
  <script src="/static/js/0.chunk.js"></script>
  <script src="/static/js/main.chunk.js"></script></body>
</html>
```

Server Side Rendered content is preferred for indexing

SERVER SIDE RENDERED CONTENT




```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="logo192.png" />
    <link rel="manifest" href="/manifest.json" />
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root">
      <div class="list">
        <div class="item">Apples</div>
        <div class="item">Kiwi</div>
        <div class="item">Mango</div>
        <div class="item">Strawberry</div>
      </div>
    </div>

  <script src="/static/js/bundle.js"></script>
  <script src="/static/js/0.chunk.js"></script>
  <script src="/static/js/main.chunk.js"></script></body>
</html>
```

Server Side Rendered content is preferred for indexing

Traditionally, websites have been server rendered since the beginning! ***#nostalgia***

SOMETHING TO THINK ABOUT

Should we go back to building sites and apps, the stone age way??

SOMETHING TO THINK ABOUT

Should we go back to building sites and apps, the stone age way??

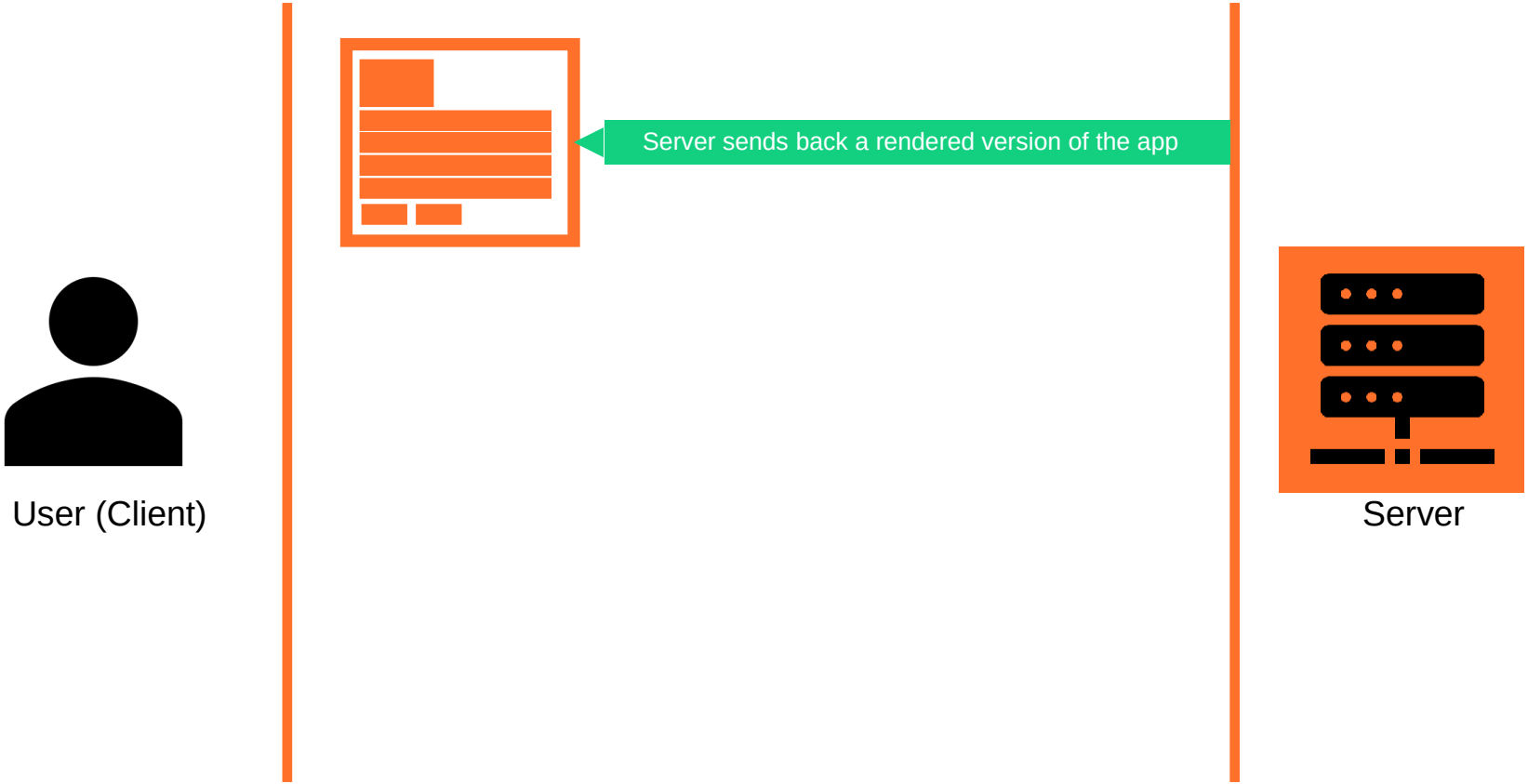
NOPES!

SOMETHING TO THINK ABOUT

SOLUTION : A middle-ground that offers the best of server side rendered pages & client side rendered apps

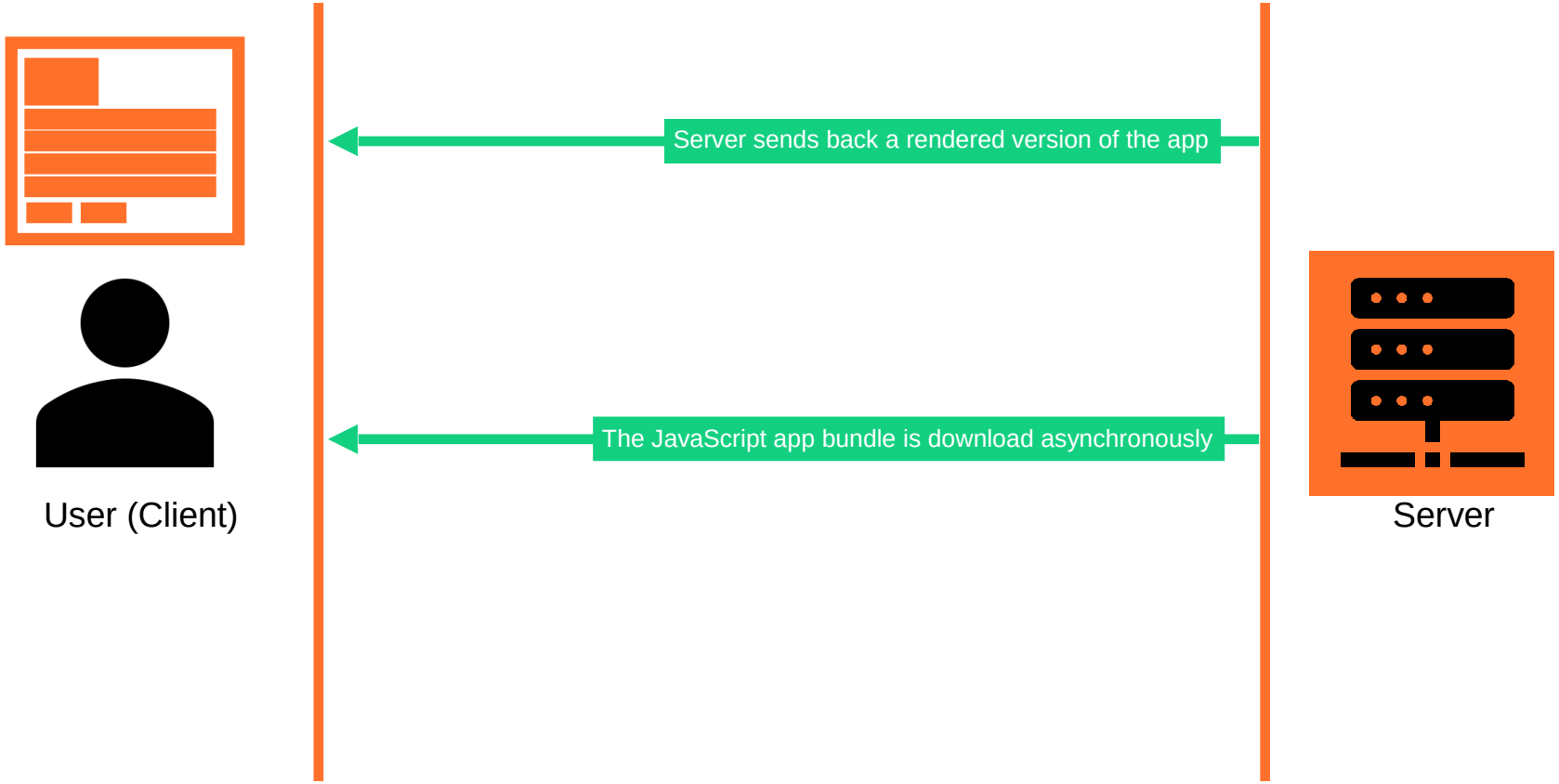
Serve a Server Side Rendered version of the app with initial data

SOLUTION TO THE PROBLEM

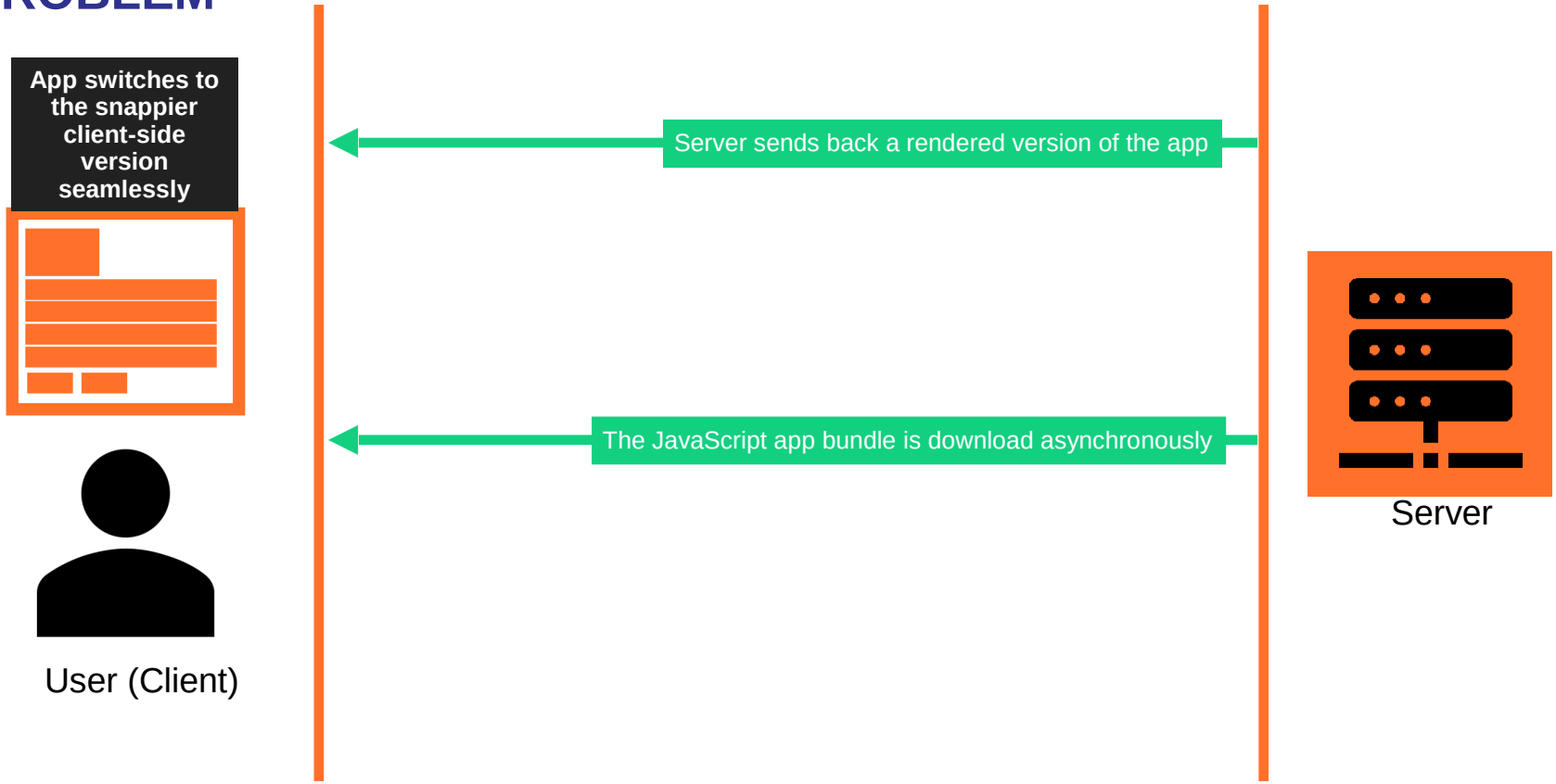


SOLUTION TO THE PROBLEM

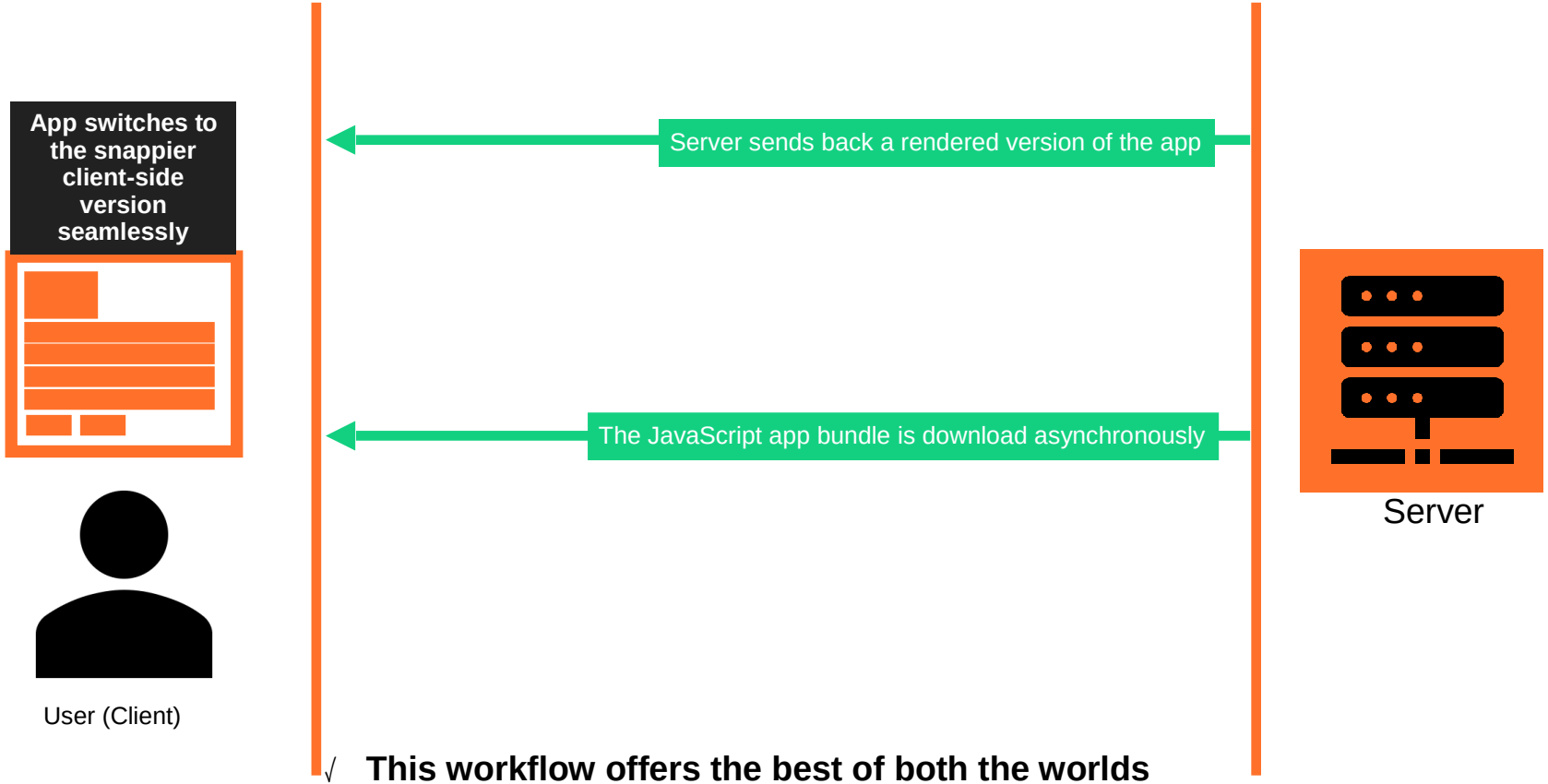
SOLUTION TO THE PROBLEM



SOLUTION TO THE PROBLEM

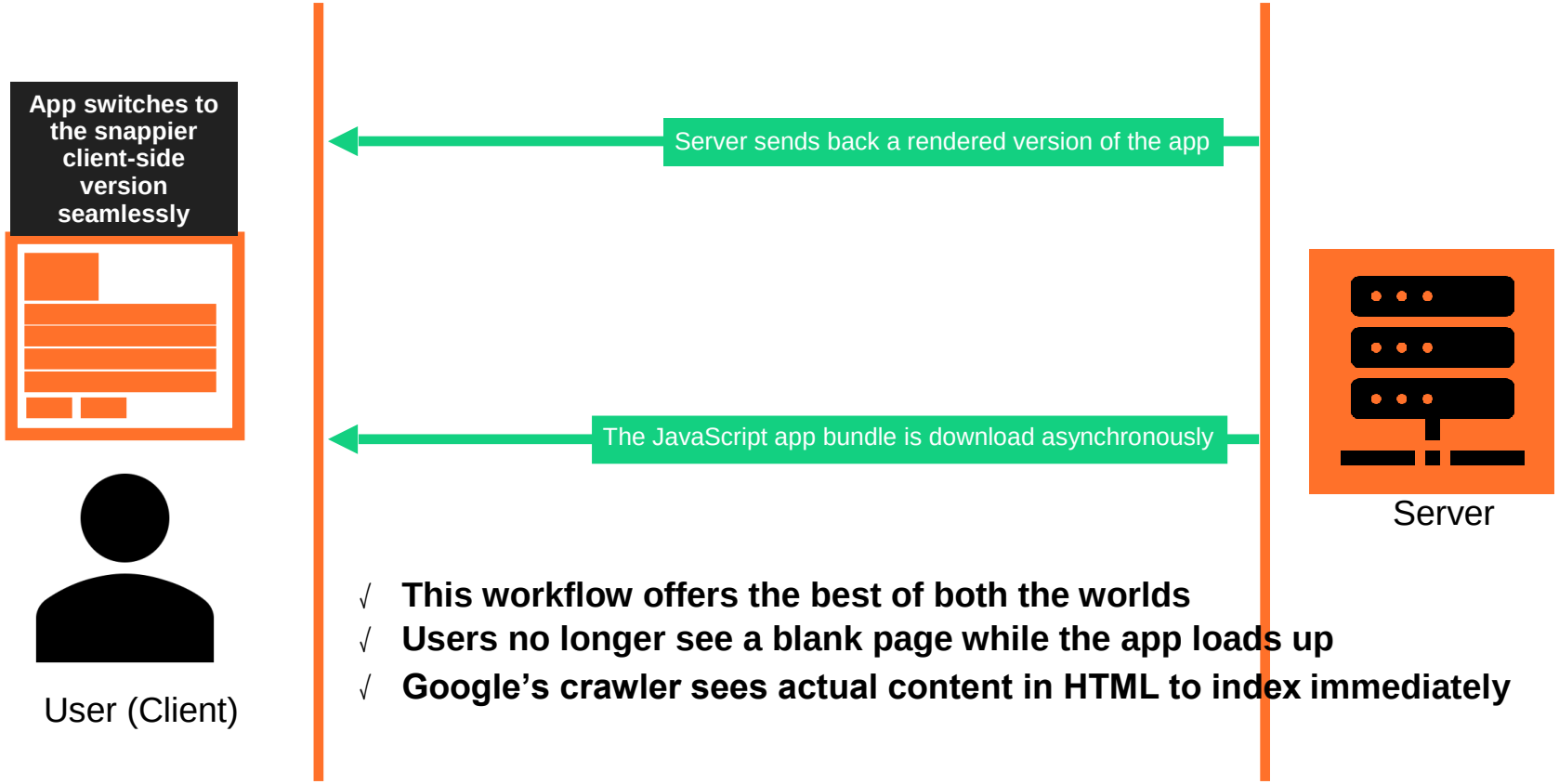


SOLUTION TO THE PROBLEM



- √ **This workflow offers the best of both the worlds**
- √ **Users no longer see a blank page while the app loads up**

SOLUTION TO THE PROBLEM

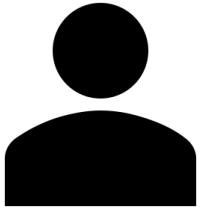


- √ **This workflow offers the best of both the worlds**
- √ **Users no longer see a blank page while the app loads up**
- √ **Google's crawler sees actual content in HTML to index immediately**

SOLUTION TO THE PROBLEM

SERVER SIDE RENDERED JAVASCRIPT APPS
Or
ISOMORPHIC OR UNIVERSAL APPS

App switches to the snappier client-side version seamlessly



User (Client)

Server sends back a rendered version of the app

The JavaScript app bundle is download asynchronously

Server

- √ **This workflow offers the best of both the worlds**
- √ **Users no longer see a blank page while the app loads up**
- √ **Google's crawler sees actual content in HTML to index immediately**

BENEFITS OF ISOMORPHIC APPS

- √ **Great for SEO & Crawlers**
- √ **Stellar experience for users**
- √ **Great solution for mobile users**
- √ **Users don’t see a blank page if the browser is unable to execute JavaScript**

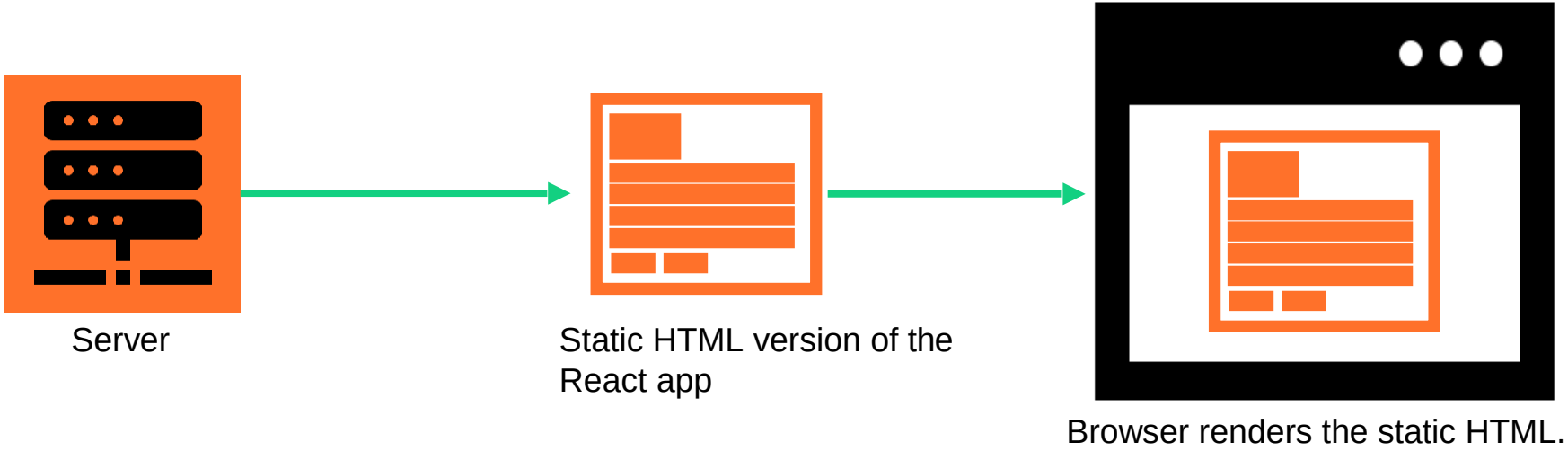
How does Server Side Rendering work?

HOW DOES SSR WORK?

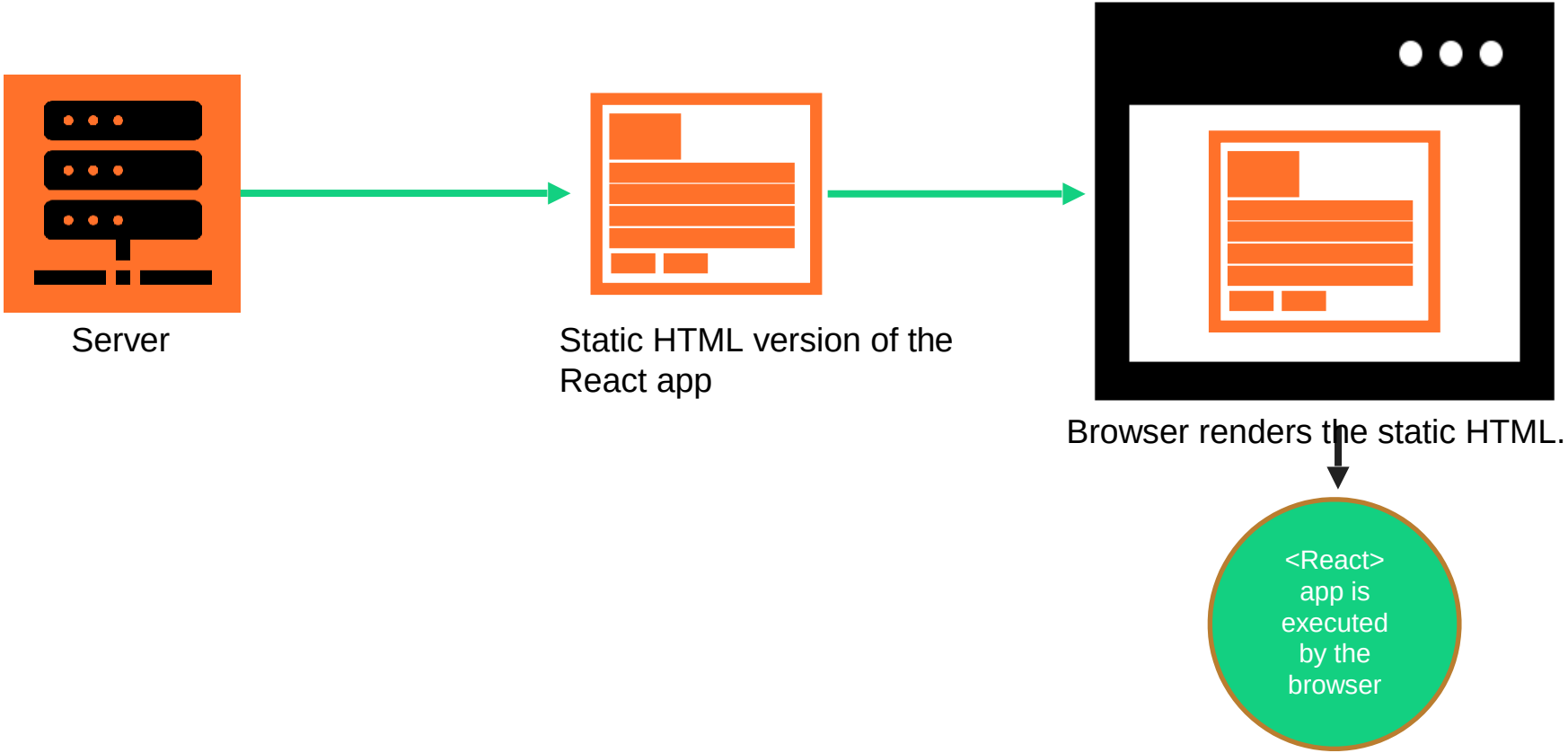
Server

Static HTML version of the React app

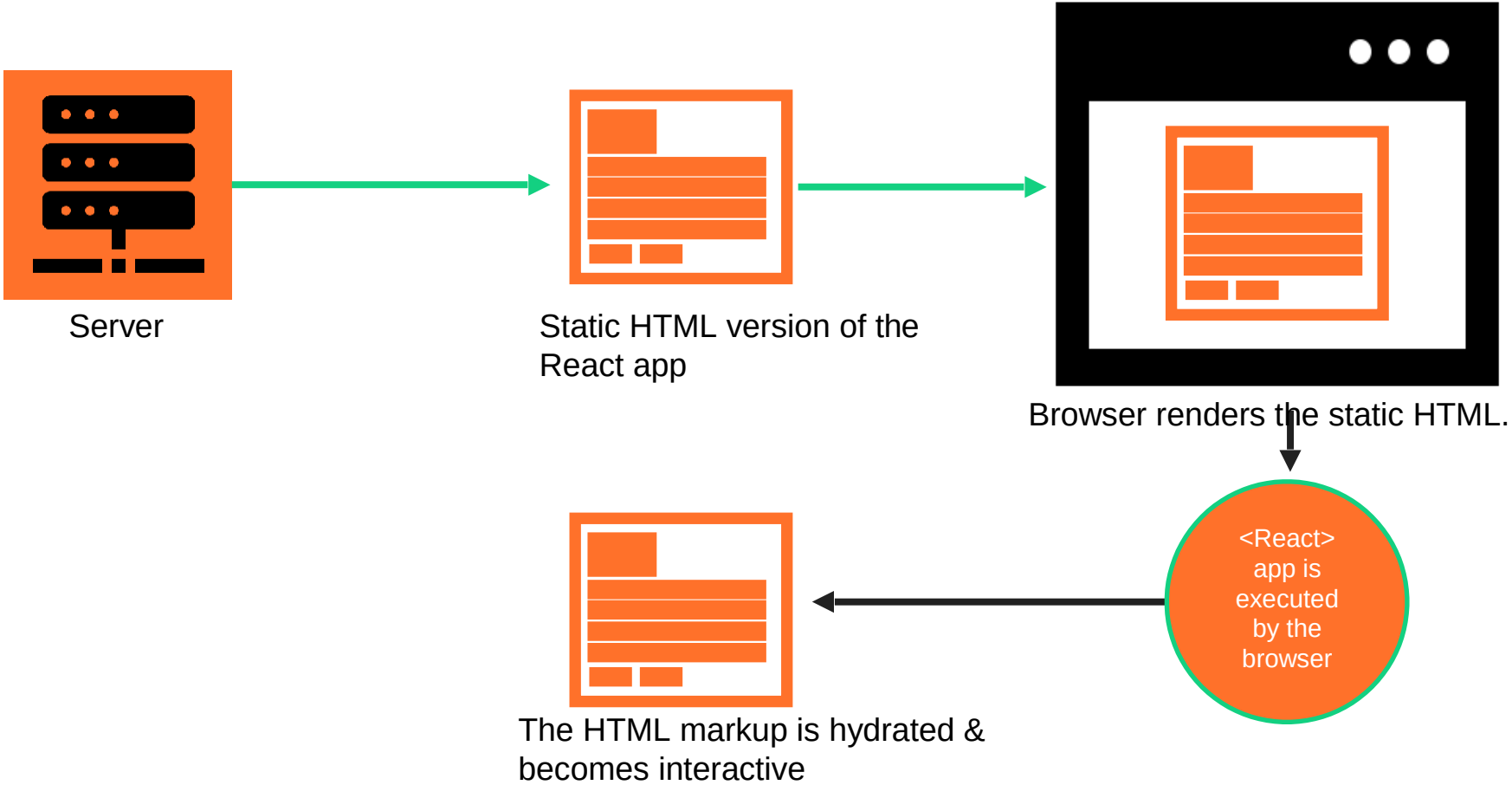
HOW DOES SSR WORK?



HOW DOES SSR WORK?



HOW DOES SSR WORK?



WHAT DO WE NEED FOR SSR?

SERVER

Node.js server (e.g. an Express app)
React & React-DOM libraries

WHAT DO WE NEED FOR SSR?

```
app.get("/", (req, res) => {
  const str = renderToString(<App />);
  res.send(html_template(str));
});
```

Renders static markup for React components

A Route Handler in an ExpressJS Application (**Server**)

WHAT DO WE NEED FOR SSR?

```
app.get("/", (req, res) => {
  const str = renderToString(<App />);
  res.send(html_template(str));
});
```

Renders static markup for React components

A Route Handler in an ExpressJS Application (**Server**)

```
ReactDOM.hydrate(<App />, document.getElementById("root"));
```

Use Hydrate to restore interactivity and the client-side version on the static markup (**Client)**

WHAT DO WE NEED FOR SSR?

```
app.get("/", (req, res) => {
  const str = renderToString(<App />);
  res.send(html_template(str));
});
```

Renders static markup for React components

A Route Handler in an ExpressJS Application (**Server**)

ISOMORPHIC REACT APPLICATIONS

```
ReactDOM.hydrate(<App />, document.getElementById("root"));
```

Use Hydrate to restore interactivity and the client-side version on the static markup (**Client)**

USING THE CREATE-REACT-APP ?

Apps scaffolded using create-react-app are client side rendered and are not easy to convert to server side rendered

Two ways to build Isomorphic React apps

TWO WAYS TO BUILD ISOMORPHIC REACT APPS

Custom toolchain using **Webpack + Babel + Node.js Server**

TWO WAYS TO BUILD ISOMORPHIC REACT APPS

Custom toolchain using **Webpack + Babel + Node.js Server**

NEXT.JS

A Framework for building Isomorphic React apps

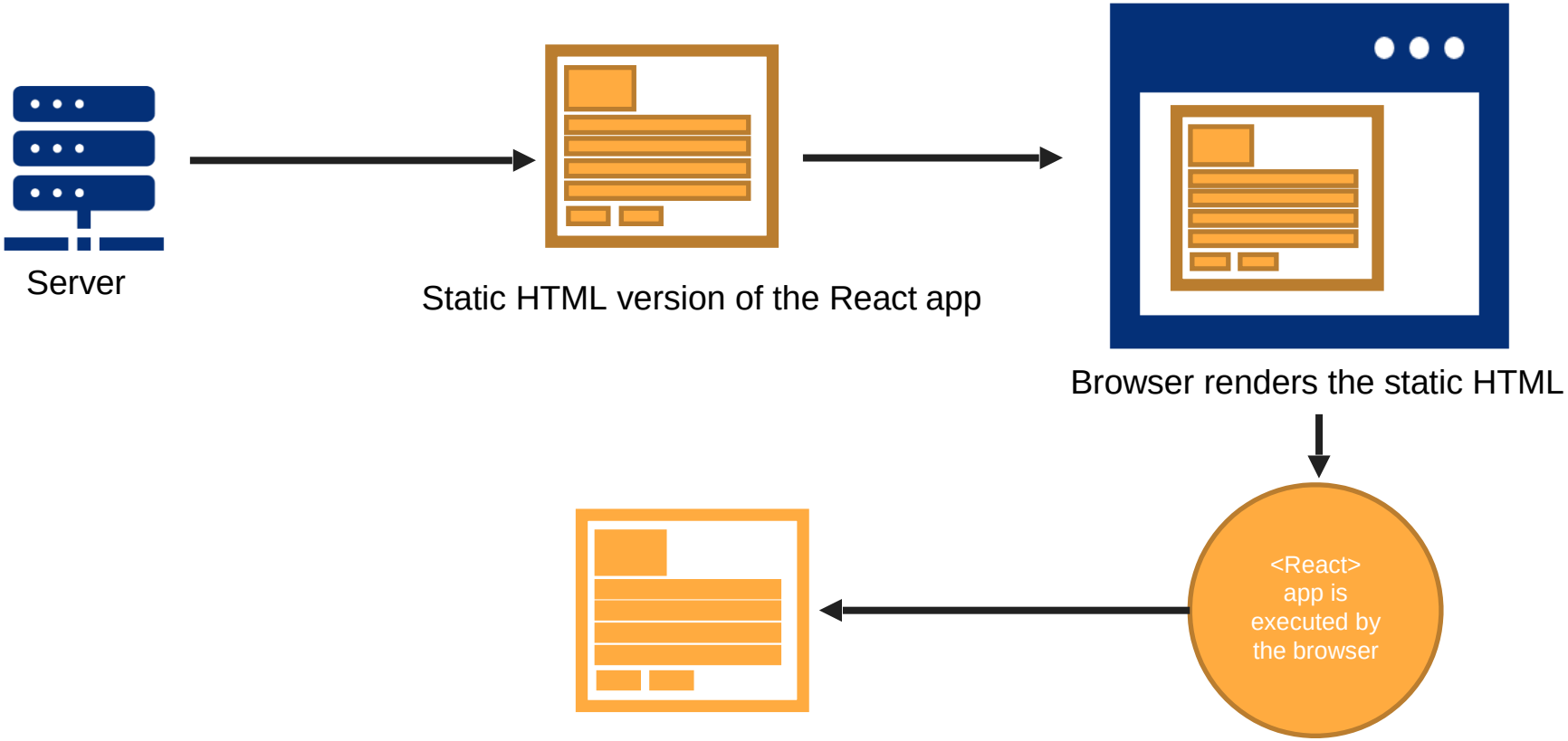
WHEN TO IMPLEMENT SSR?

- SEO Compatibility
- App bundles are big

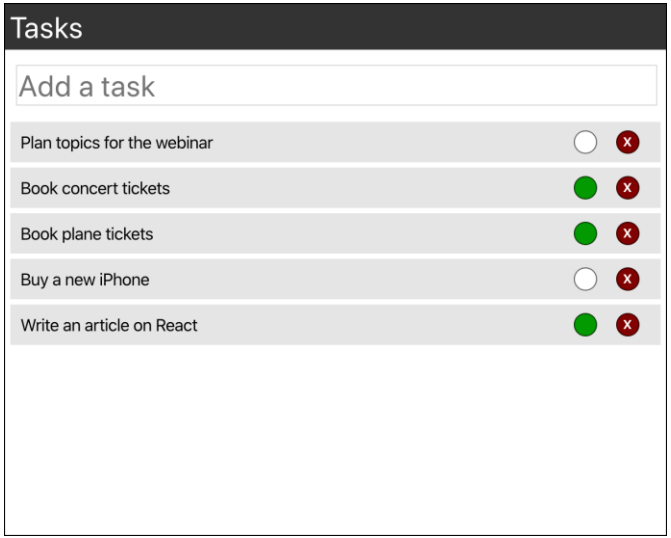
Isomorphic React

Server-Side Rendering with React- Setup & Server

HOW DOES SSR WORK?

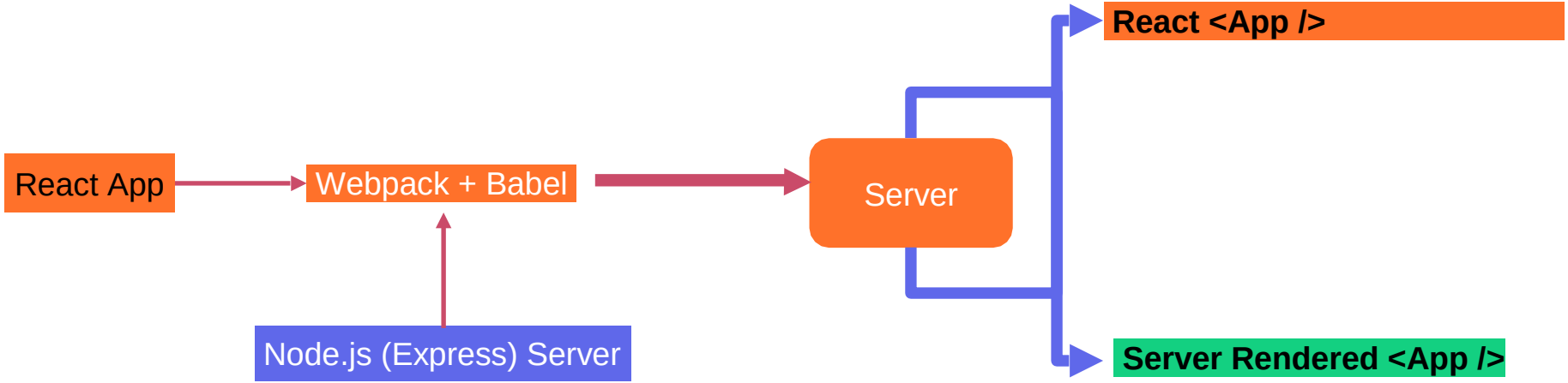


CONVERTING A CLIENT SIDE RENDERED APP TO A SERVER SIDE RENDERED APP



SSR

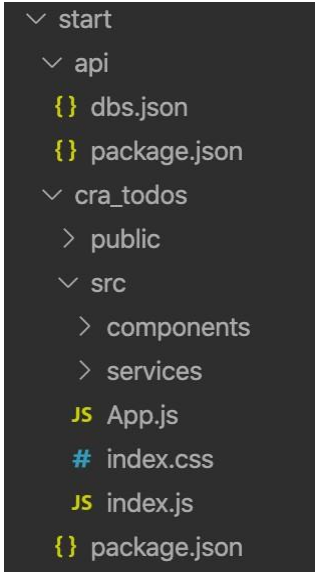
TOOLCHAIN TO COMPILE REACT + NODE.JS SERVER



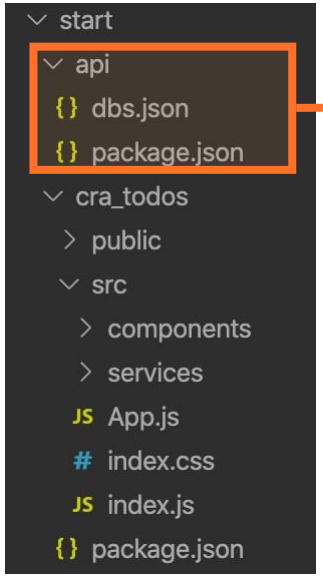
Download and Extract to a folder

https://my.pcloud.com/publink/show?code=XZoWRbkZEiOKvBQW9QHwh08FFargR8UyyM2k

THE PROJECT

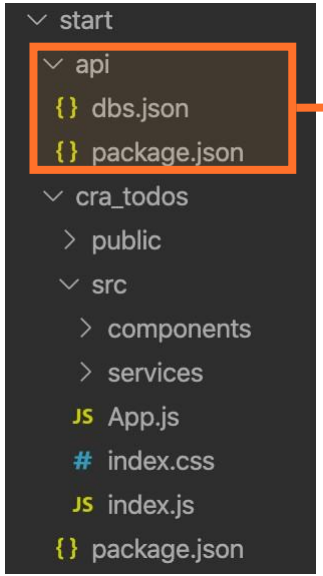

start
api
dbs.json
package.json
cra_todos
public
src
components
services
App.js
index.css
index.js
package.json

THE PROJECT



An API Server (Backend) built using the json-server package

THE PROJECT



npm install

An API Server (Backend) built using the json-server package

THE PROJECT

npm install

An API Server (Backend) built using the json-server package

An React App built using the create-react-app toolchain

THE renderToString FUNCTION

```
app.get("/", (req, res) => {
  // Render the React app & server
  const str = renderToString(<App />);
  res.send(html_template(str));
});
```

```
<div id="root">
  <div class="app" data-reactroot="">
    <div class="title">Tasks</div>
    <div class="add-todo"><input type="text" placeholder="Add a task" value=""/></div><div class="list"><div class="list-item"><div class="task">Plan topics for the webinar</div><div class="status-btn"></div><div class="delete-btn">X</div></div><div class="list-item"><div class="task">Book concert tickets</div><div class="status-btn status-done"></div><div class="delete-btn">X</div></div><div class="list-item"><div class="task">Book plane tickets</div><div class="status-btn status-done"></div><div class="delete-btn">X</div></div><div class="list-item"><div class="task">Buy a new iPhone</div><div class="status-btn"></div><div class="delete-btn">X</div></div><div class="list-item"><div class="task">Write an article on React</div><div class="status-btn status-done"></div><div class="delete-btn">X</div></div>
  </div>
</div>
</div>
```

THE renderToNodeStream FUNCTION

```
ReactDOM.renderToNodeStream(<App />)
```

Returns a **Readable Stream** of the HTML content

THE renderToStaticMarkup Function

```
ReactDOM.renderToStaticMarkup(<App />)
```

This function doesn’t add the **data-reactroot** attribute thus producing static HTML content which cannot be hydrated on the client

Isomorphic React

Server-Side Rendering with React- The tool chain

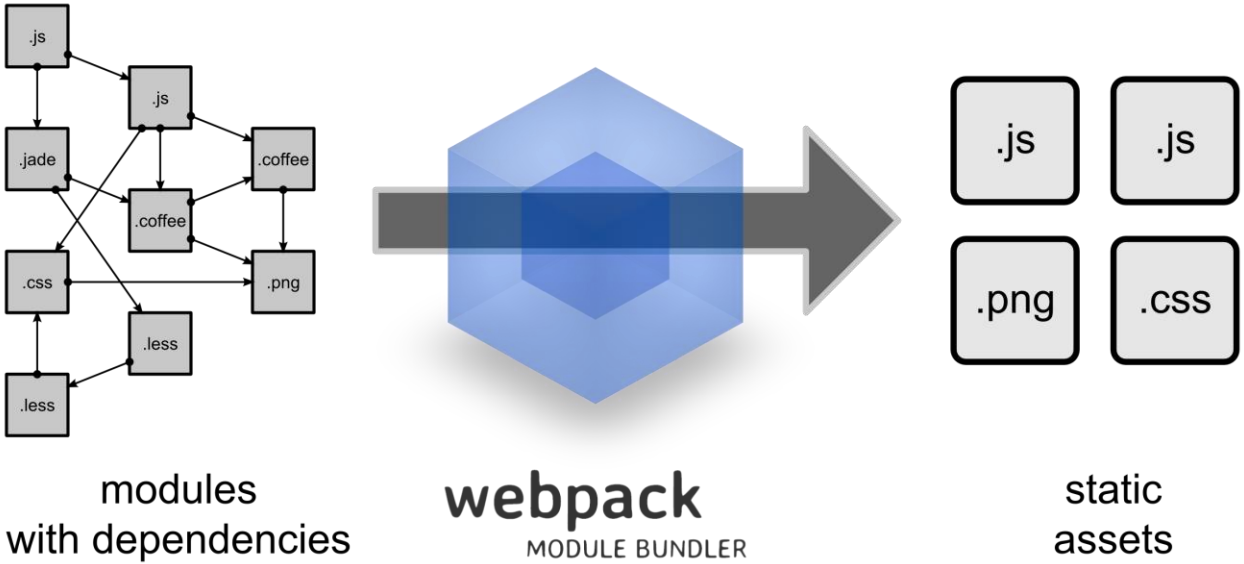
PACKAGES & MODULES USED IN THE SSR PROJECT

```
"dependencies": {
  "axios": "^0.19.0",
  "express": "^4.17.1",
  "react": "^16.12.0",
  "react-dom": "^16.12.0"
},
"devDependencies": {
  "@babel/core": "^7.7.5",
  "@babel/preset-env": "^7.7.6",
  "@babel/preset-react": "^7.7.4",
  "babel-loader": "^8.0.6",
  "css-loader": "^3.3.0",
  "ignore-loader": "^0.1.2",
  "mini-css-extract-plugin": "^0.8.0",
  "webpack": "^4.41.2",
  "webpack-cli": "^3.3.10",
  "webpack-merge": "^4.2.2",
  "webpack-node-externals": "^1.7.2"
}
```

NODE.JS (EXPRESS) SERVER APPLICATION

```
import express from "express";
import React from "react";
import { renderToString } from"react-dom/server";
import axios from "axios";
import App from "./app/App";


const apiServer = process.env.API_SERVER || "http://localhost:8080/todos";
const port = process.env.PORT || 3000;


const app = express();


app.use(express.static("build/js"));
app.use(express.static("build/css"));


const html_template = app =>
  `...`;


app.get("/", (req, res) => {
  // Render the React app & server
  const str = renderToString(<App />);
  res.send(html_template(str));
});


app.listen(port, () => console.log(`SSR Server running on port: ${port}`));
```

WEBPACK PROCESSED AND COMPILES TOGETHER MODULAR CODE INTO BUNDLES & ASSETS

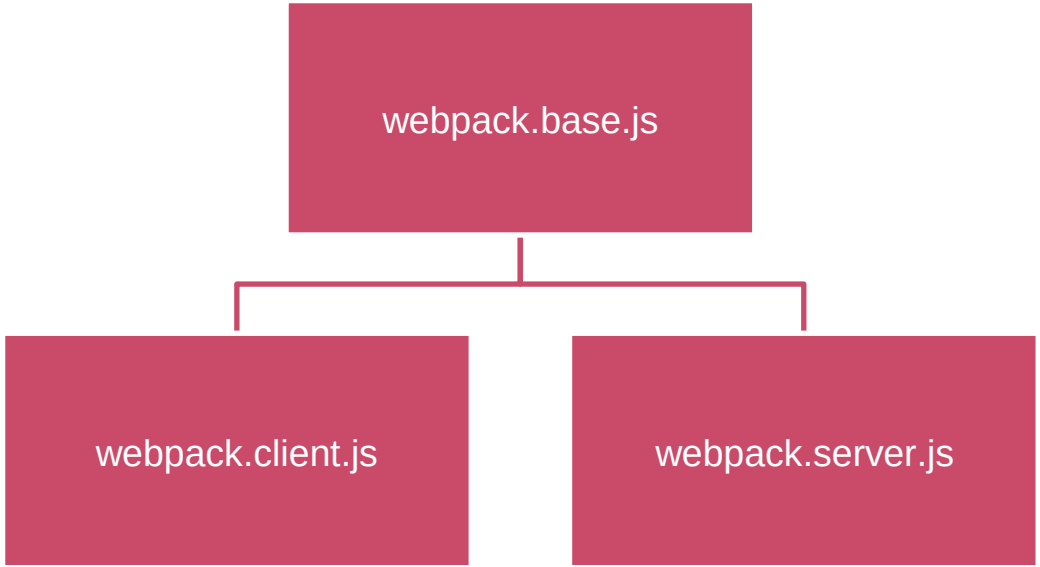


Source: https://webpack.github.io/

WEBPACK AND DEPENDENCIES

```
"devDependencies": {
    "@babel/core": "^7.7.5",
    "@babel/preset-env": "^7.7.6",
    "@babel/preset-react": "^7.7.4",
    "babel-loader": "^8.0.6",
    "css-loader": "^3.3.0",
    "ignore-loader": "^0.1.2",
    "mini-css-extract-plugin": "^0.8.0",
    "webpack": "^4.41.2",
    "webpack-cli": "^3.3.10",
    "webpack-merge": "^4.2.2",
    "webpack-node-externals": "^1.7.2"
  }
```

A BARE BONES SETUP

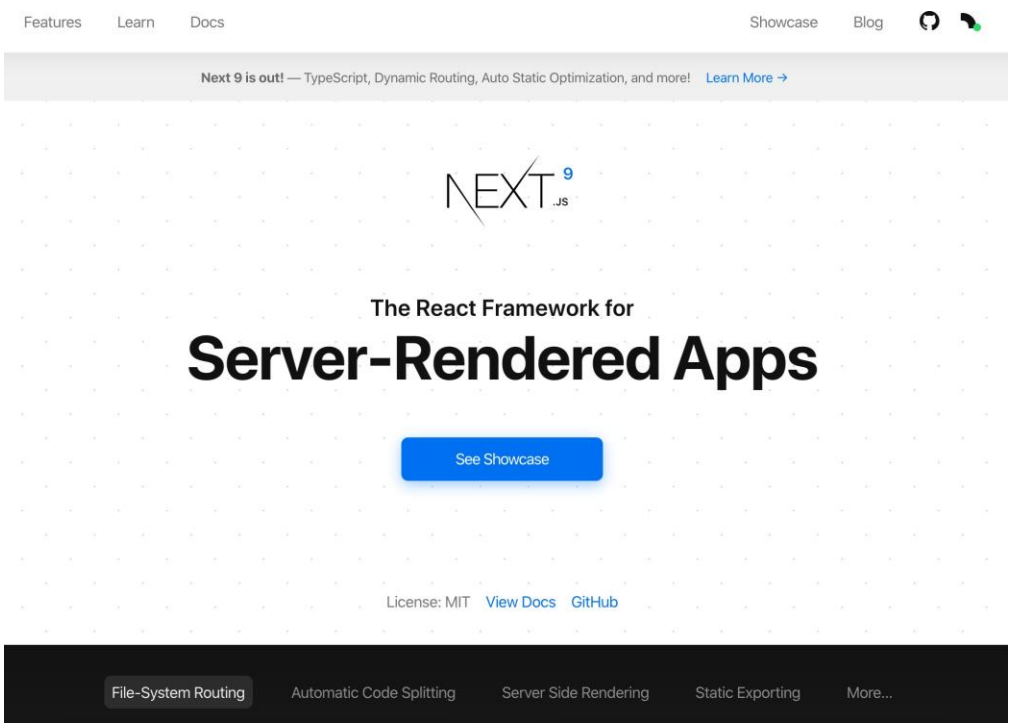


POSSIBILITIES WITH WEBPACK

- √ Support for chunks and dynamic imports
- √ Processing images and other assets

Configuration depends on use case

NEXT.JS FRAMEWORK



https://nextjs.org/

thank you!