Artículo
Javier Lorenzo Mesa · Abr 20, 2020 Lectura de 5 min

Node.js: Cómo crear una aplicación web básica con React (Parte 1)

¡ Hola Comunidad!

El desarrollo de una aplicación web Full-Stack en JavaScript con Caché requiere que se junten los bloques correctos para construirla. Anteriormente, describí cuáles son los pasos básicos para instalar y conectar Node.js con Caché y hacer que sus potentes capacidades como base de datos multimodelo estén disponibles para utilizarse con Node.js. Se puede usar Caché como una base de datos NoSQL, de documentos (¡ con acceso único al nivel de la clave!), SQL y base de datos orientada a objetos con Node.js. Cuando desarrolles aplicaciones en JavaScript, verás lo potente que es esta combinación y cómo hace que Caché encaje perfectamente con Node.js.

En la primera parte de esta serie de artículos, mostraré cómo comenzar con el framework React, uno de los entornos más populares y que actualmente es una potente alternativa de librería para el desarrollo front-end. En los siguientes artículos aprenderemos cómo conectar una aplicación web básica con un back-end de Caché.

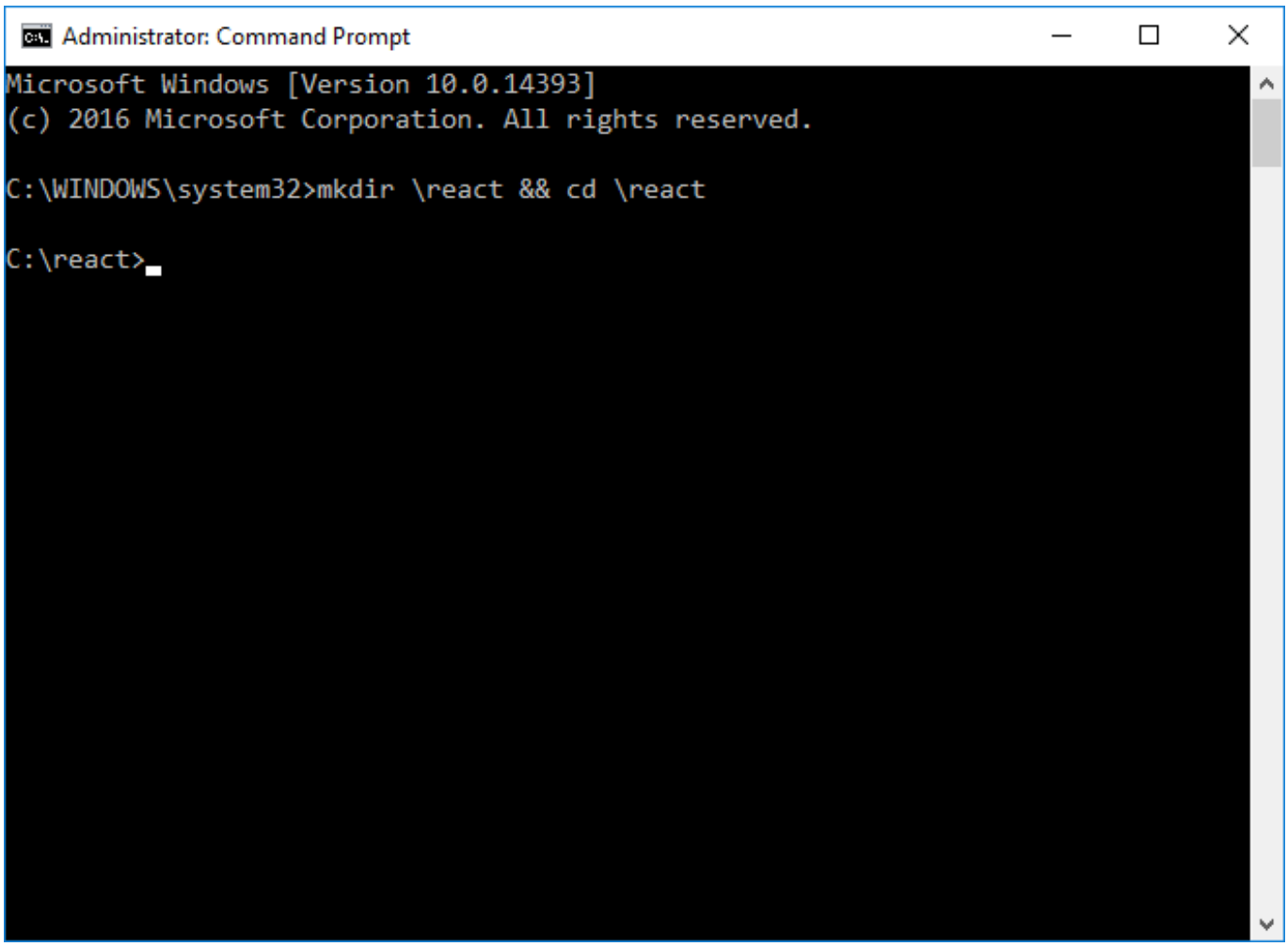
Ya verás, es muy fácil utilizar esta tecnología. Comparado con la cantidad de conocimientos básicos que se necesitan cuando se utiliza COS, solo se necesitan unos conceptos básicos para empezar.

Primero, necesitas tener Node.js instalado y funcionando con Caché, como he descrito en este artículo sobre Node.js. Una vez que tengas Node.js instalado en tu sistema, podrás crear esta primera demo de una aplicación en React muy fácilmente.

De hecho, esta parte es el primer paso donde se muestra cómo crear una aplicación en Node.js con React. Crearemos el front-end de una aplicación básica, la cual mejoraremos más adelante en la parte 3 para conectar su base de datos con Caché.

Ya hemos tenido suficiente introducción... ¡ nos ponemos manos a la obra y empezamos!

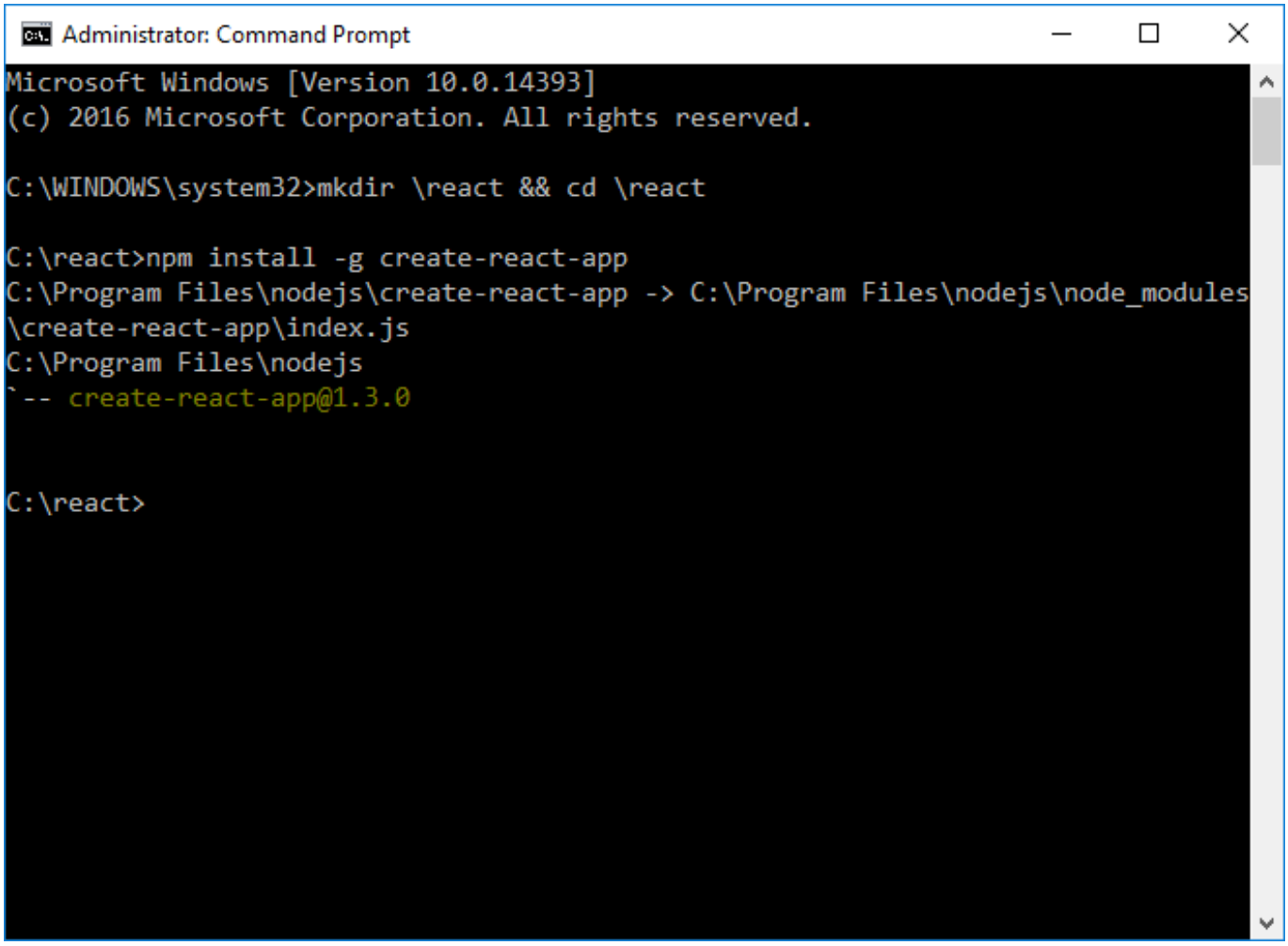
Primero, necesitas abrir una línea de comandos en Windows y crear el directorio donde se almacenarán tus aplicaciones web de Node.js. Mi directorio será C: \ react



Para facilitar nuestra vida mientras utilizamos React, los desarrolladores de React crearon un módulo Node.js que permite generar un esqueleto completo de la aplicación en React y configurar automáticamente todos los nodos en los módulos que sean necesarios: create-react-app. Este módulo permite desarrollar un entorno de ejecución completo para depurar y ejecutar tu aplicación.

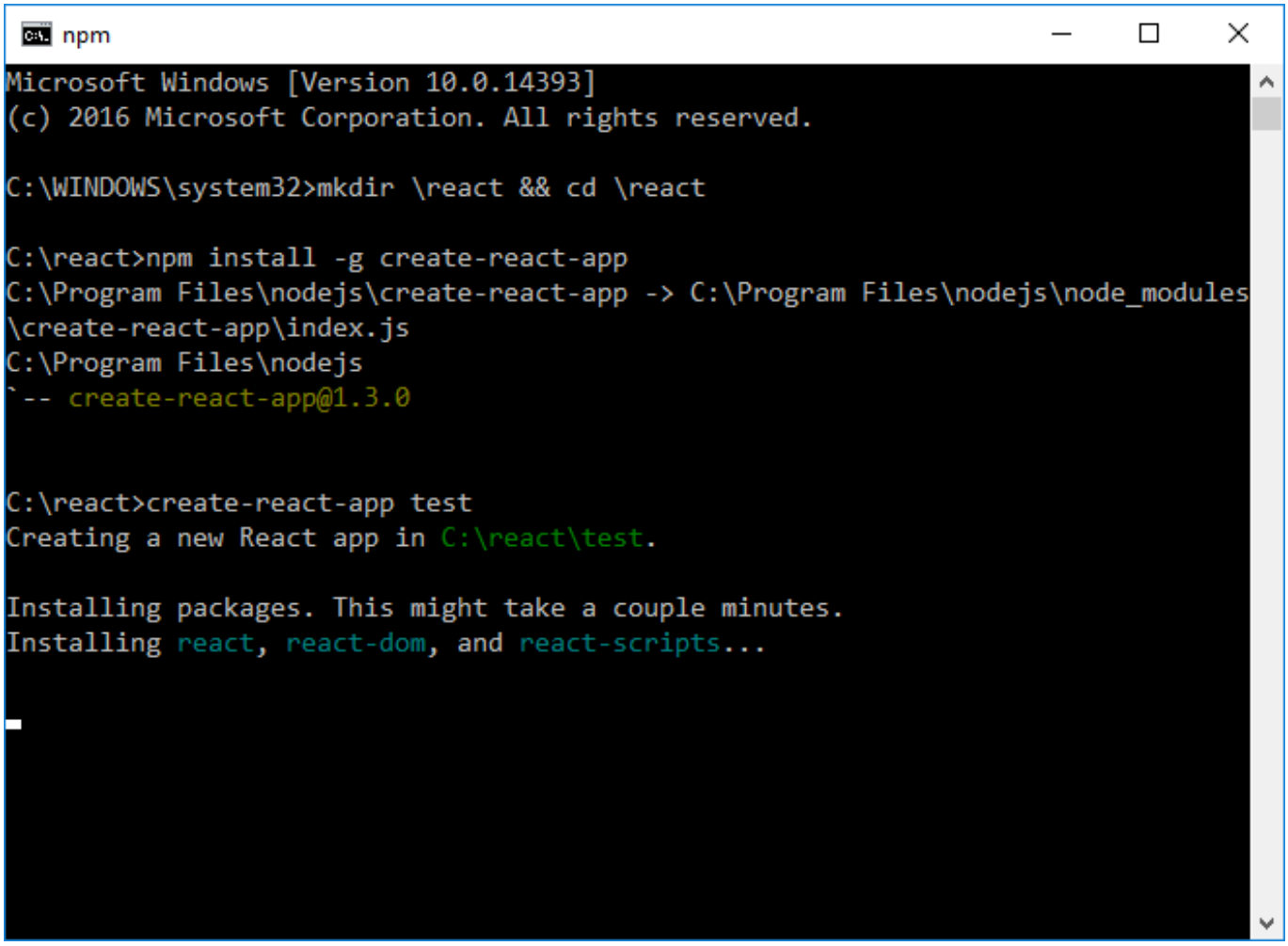
Para que este potente módulo esté disponible en todo el sistema, tenemos que instalarlo mediante la utilidad NPM. El módulo NPM (Node package manager) es la utilidad que necesitas para instalar y administrar los módulos node.js en tus aplicaciones. Se instalará automáticamente cuando instales Node.js. Cuando desarrolles aplicaciones en los nodos, utilizarás esta utilidad de línea de comandos muy frecuentemente.

Veamos cómo realizar la instalación del módulo create-react-app de manera general:



Verás que hay mucha actividad durante la instalación, no te preocupes porque este proceso utiliza muchos módulos de Node.js que están ocultos.

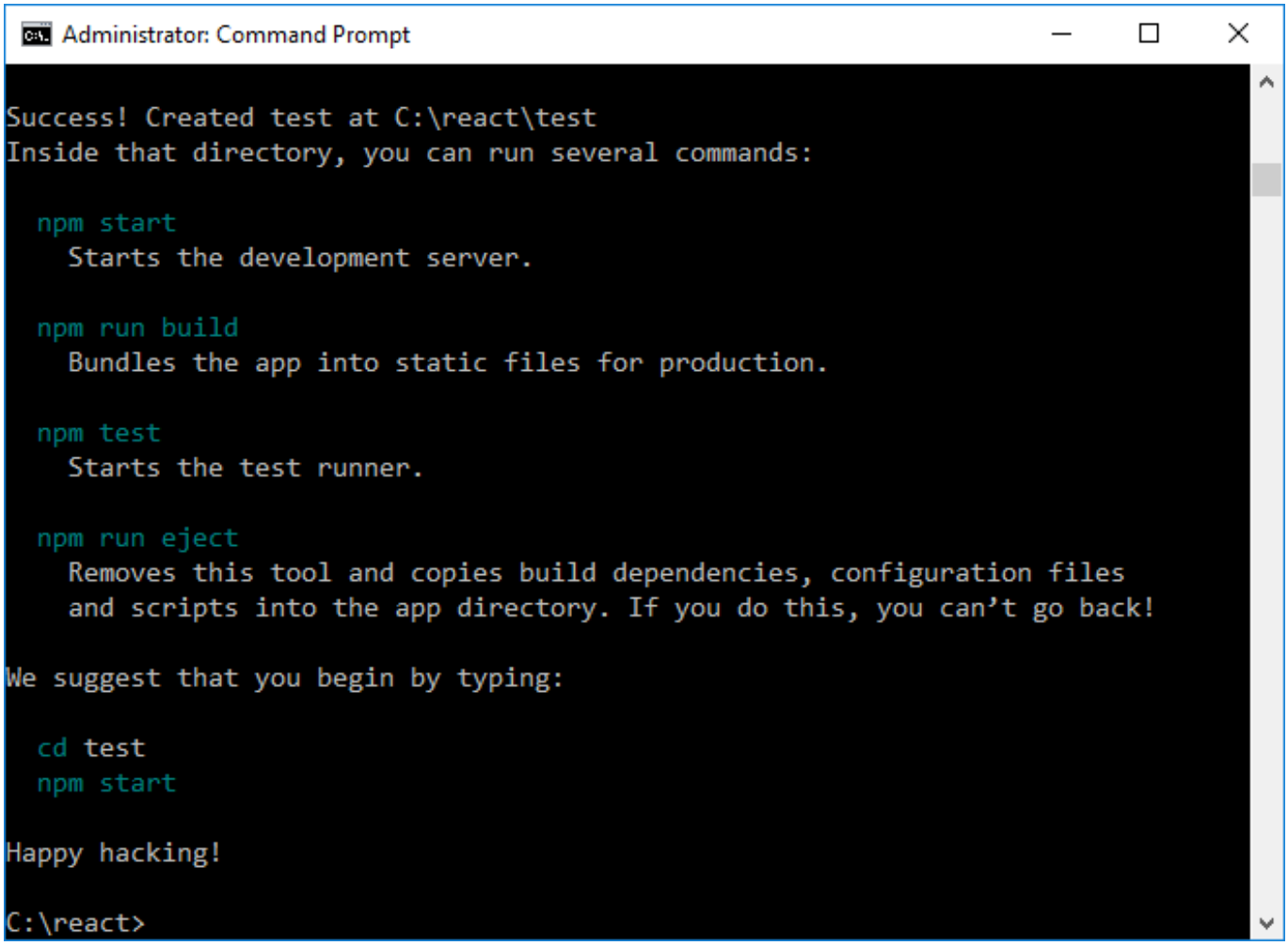
Ahora que tenemos habilitado el comando create-react-app, el primer paso es dejar que esta herramienta genere una aplicación para nosotros:



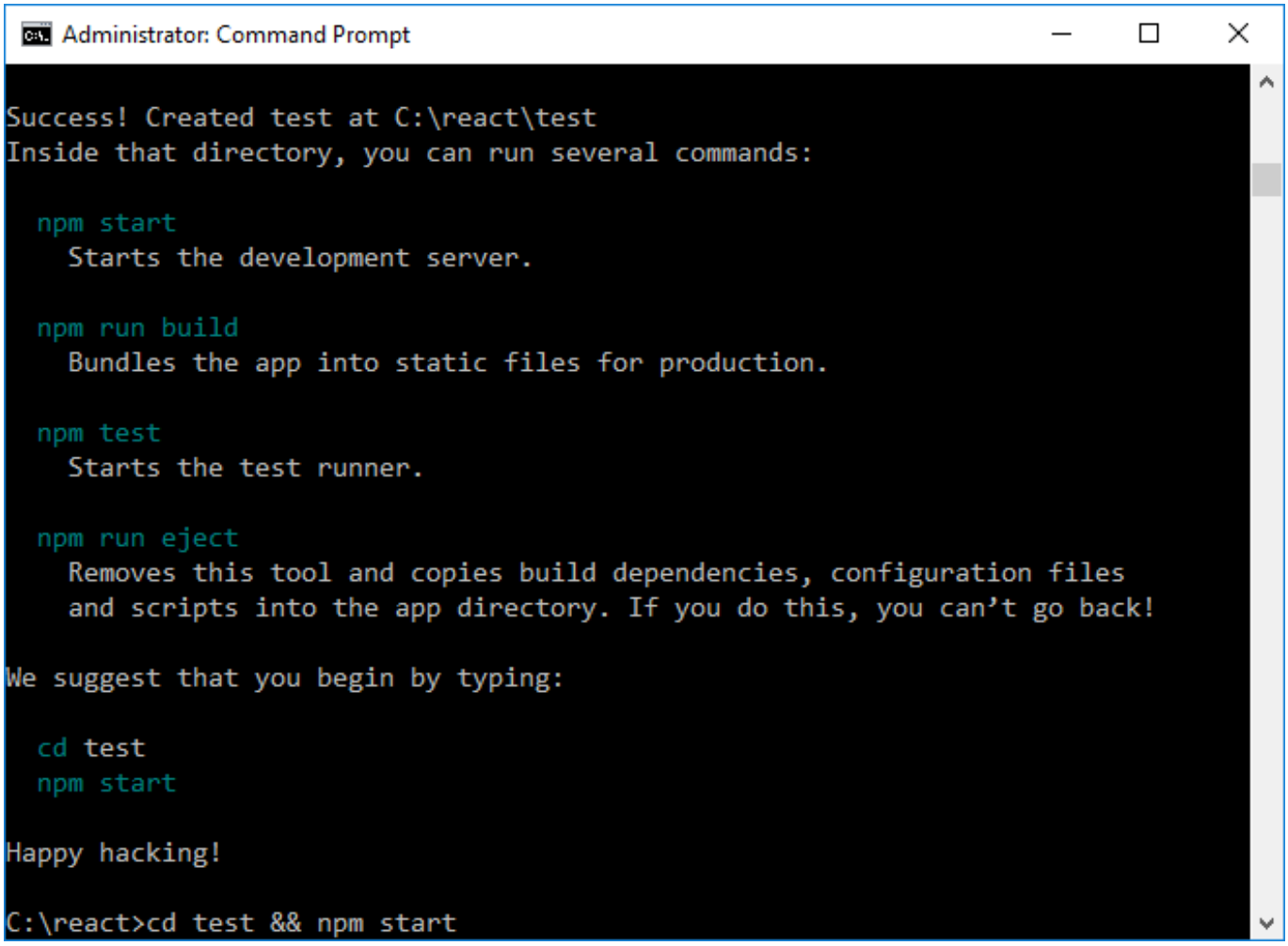
Verás que se creó un directorio de la aplicación para ti (C: \ react \ test) y que también se descargan e instalan un montón de módulos del nodo (npm) en este subdirectorio de la prueba. No te asustes, la mayoría de estos módulos solamente son importantes para el desarrollo, ¡ no son necesarios en producción!

Esto también te muestra los conocimientos básicos sobre cómo funciona Node.js: el entorno de ejecución de Node.js contiene las características que necesitas del lenguaje JavaScript y las funciones básicas del sistema, sin embargo, ¡ el verdadero poder viene de los más de 450.000 módulos que están disponibles en la comunidad de Node.js!

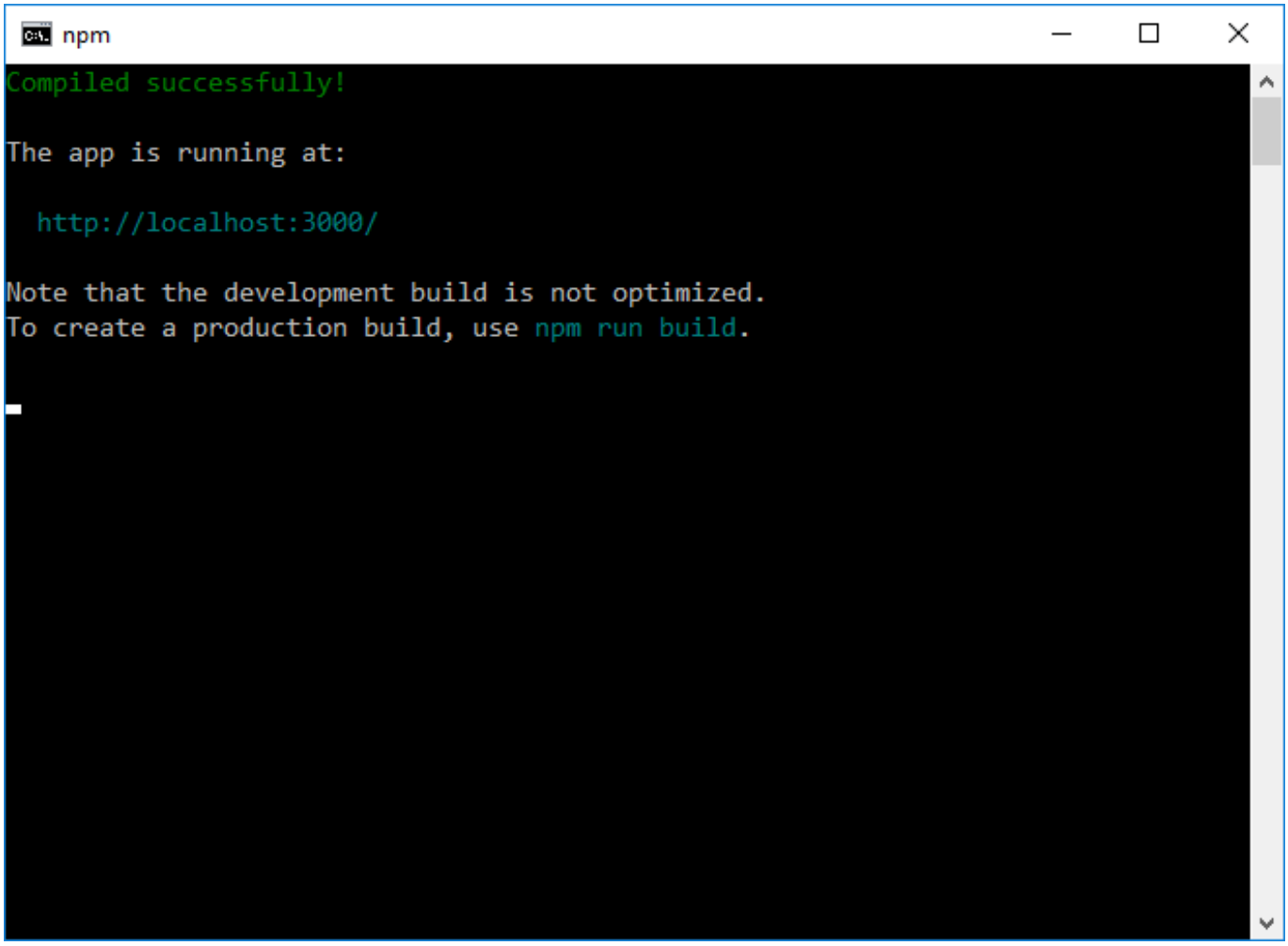
Cuando la aplicación de prueba se haya generado completamente, deberías ver lo siguiente:



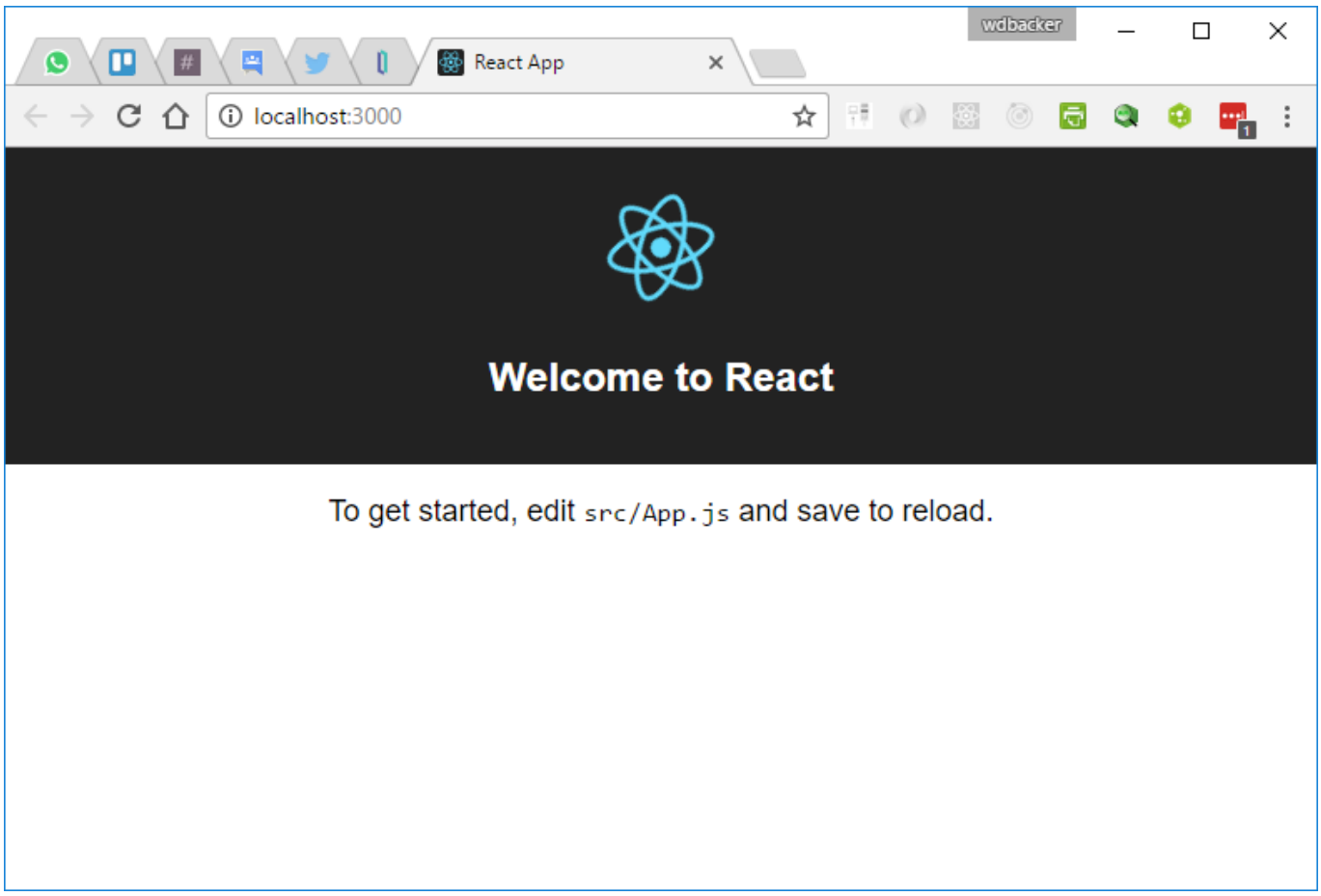
Ahora todo lo que debemos hacer es entrar en el directorio de prueba de la aplicación y ejecutar "npm start" para ejecutar la aplicación:



Este comando iniciará un servidor de programación en Node.js, que se ejecutará en localhost en el puerto 3000 y abrirá la aplicación para ti en tu navegador estándar (Chrome):



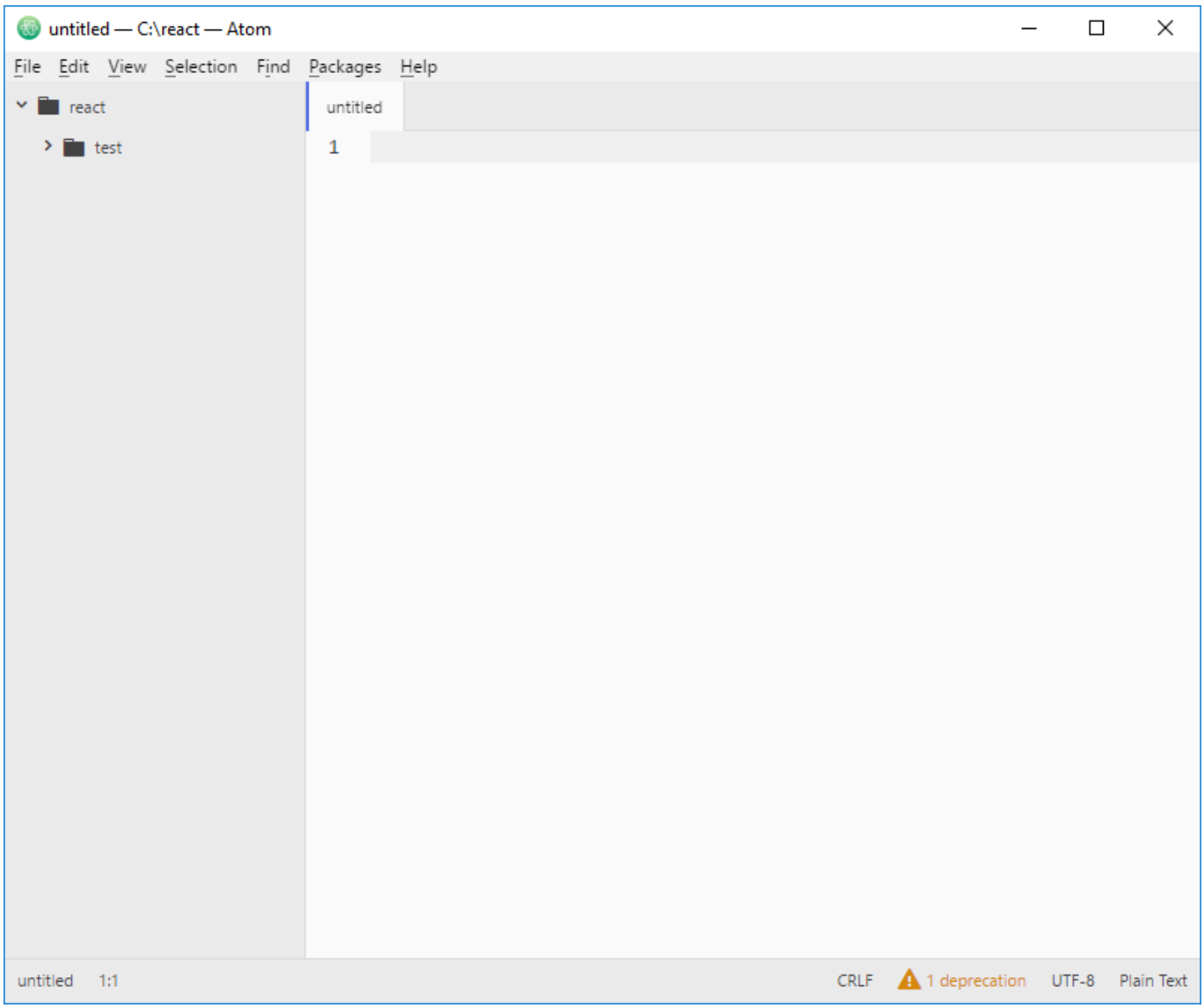
Y en tu navegador, deberías ver lo siguiente:



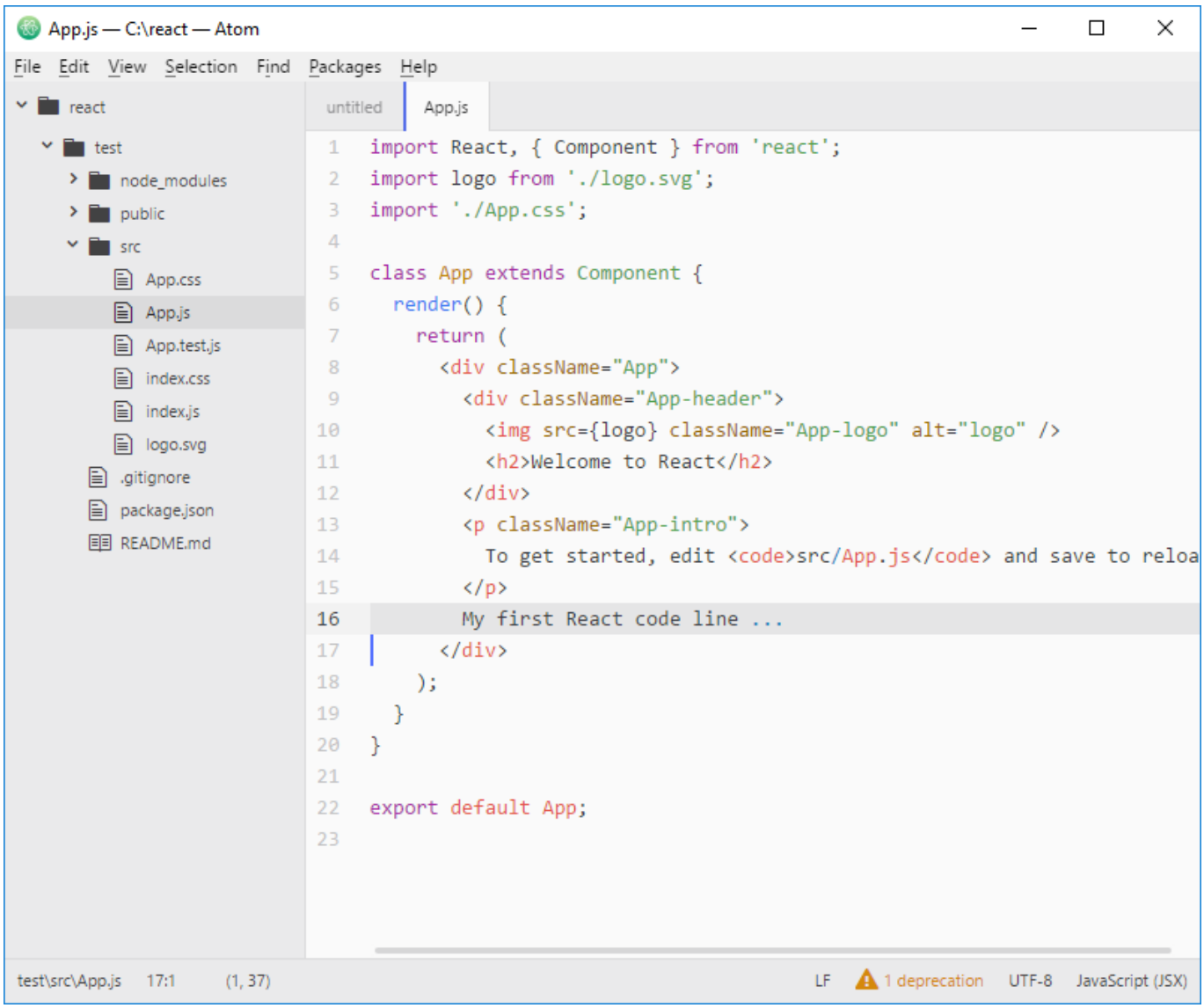
¡ Enhorabuena!¡ Acabas de crear tu primera aplicación en React! Eso es todo lo que necesitas para crear una aplicación web que sea totalmente funcional.

Como todavía no hemos escrito ni una sola línea de código, ahora también configuraremos nuestro entorno de programación para editar nuestro código fuente. El editor de código fuente (o IDE) que elijas dependerá totalmente de ti, pero una opción muy popular en estos días es el editor Atom. Solo tienes que descargarlo e instalarlo en tu sistema, es muy ligero.

Inicia el editor, y añade tu carpeta del proyecto C: \ react en el menú File:

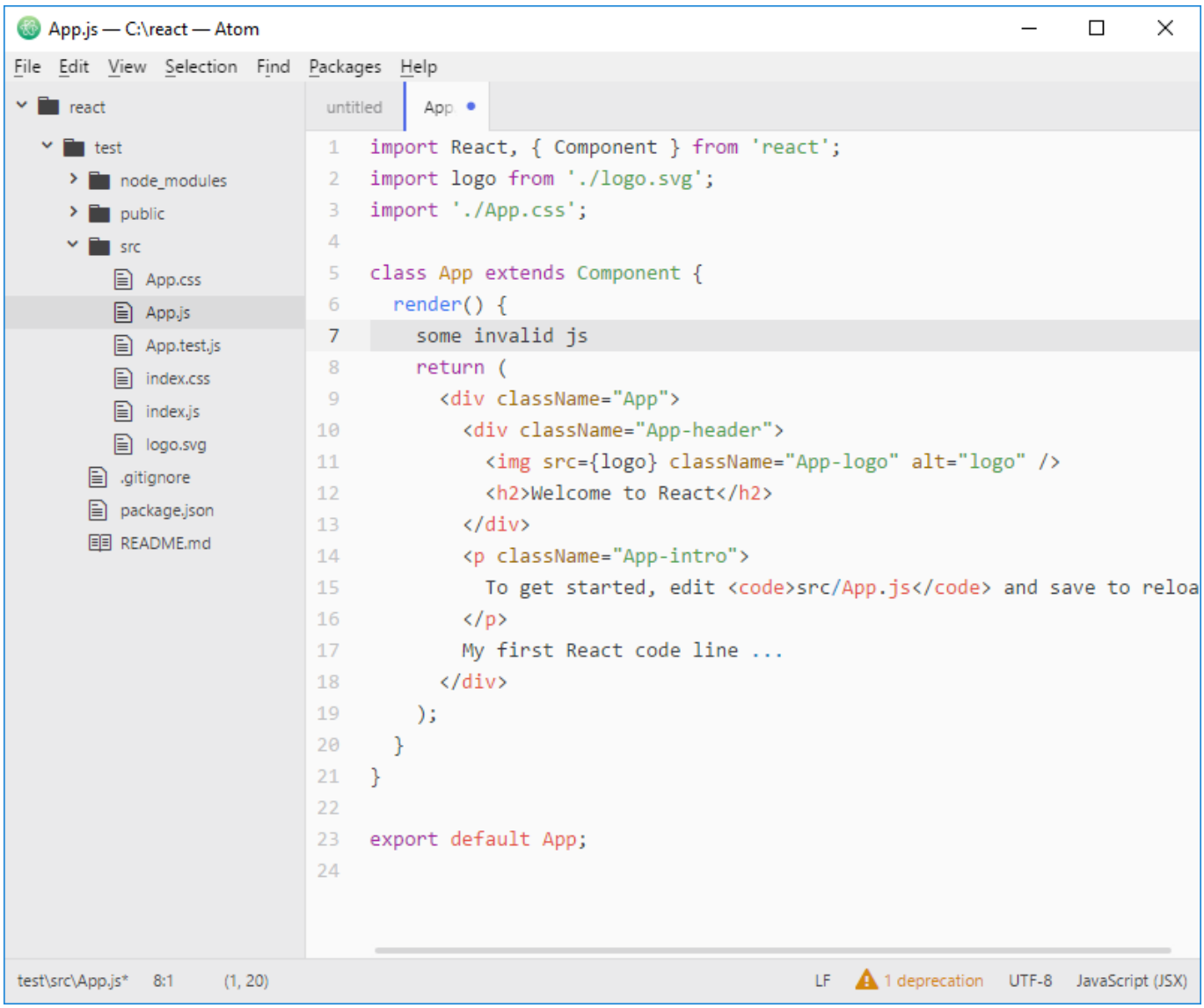


Abre la carpeta de la aplicación de prueba y dentro de la subcarpeta src, abre el archivo App.js. Añade a este archivo la línea de código que aparece resaltada, presiona el botón "Save" y mira como se actualiza la aplicación (refresca) por sí misma en tu navegador, en tiempo real:

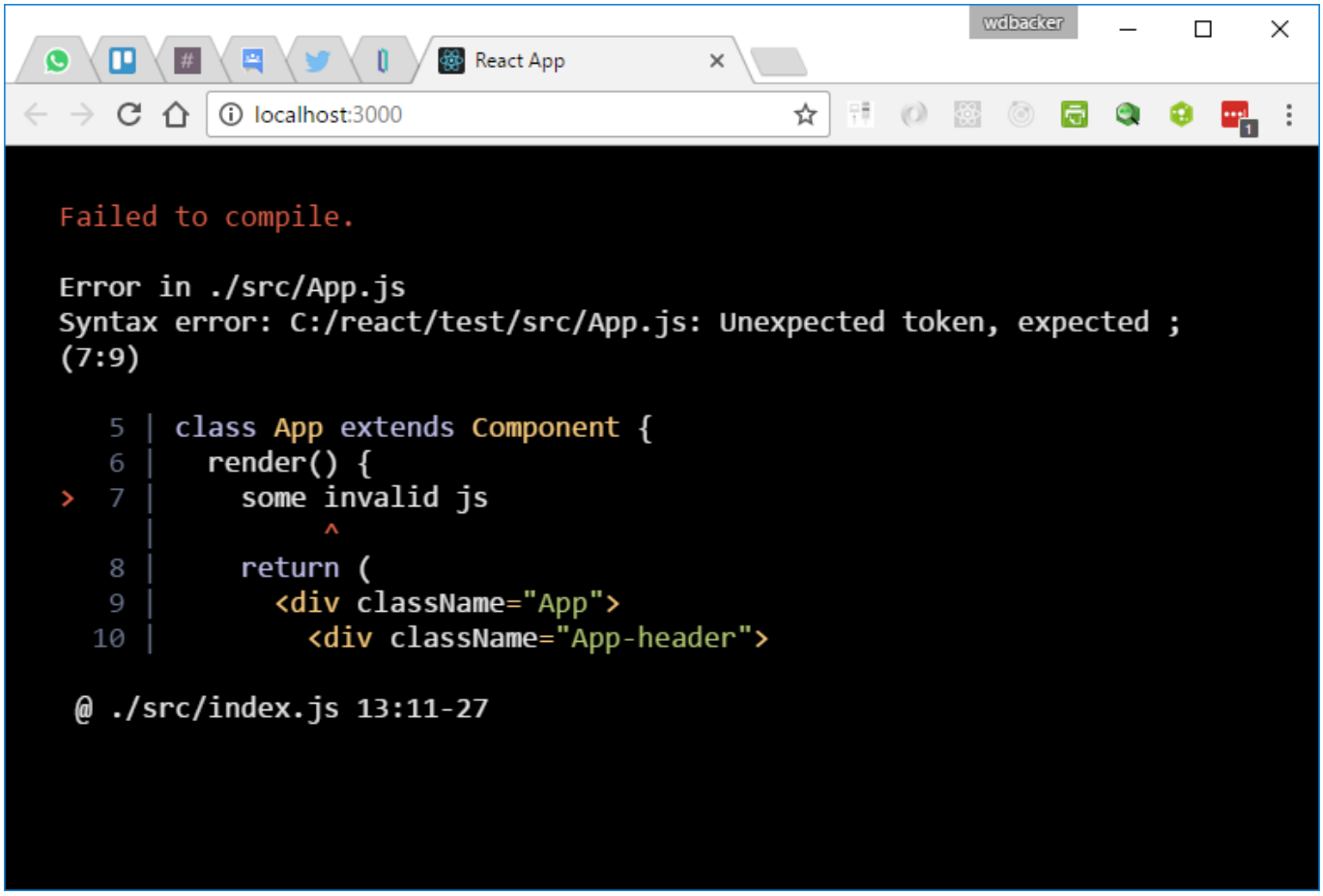


Lo que acabas de ver en tu navegador es lo que hace que las herramientas de Node.js sean tan potentes: ya están completamente integradas para ti y puedes usarlas libremente para desarrollar y depurar tus aplicaciones de una forma muy sencilla.

Ahora provoquemos un error en el código fuente al añadir algún JavaScript que no sea válido dentro del método render():

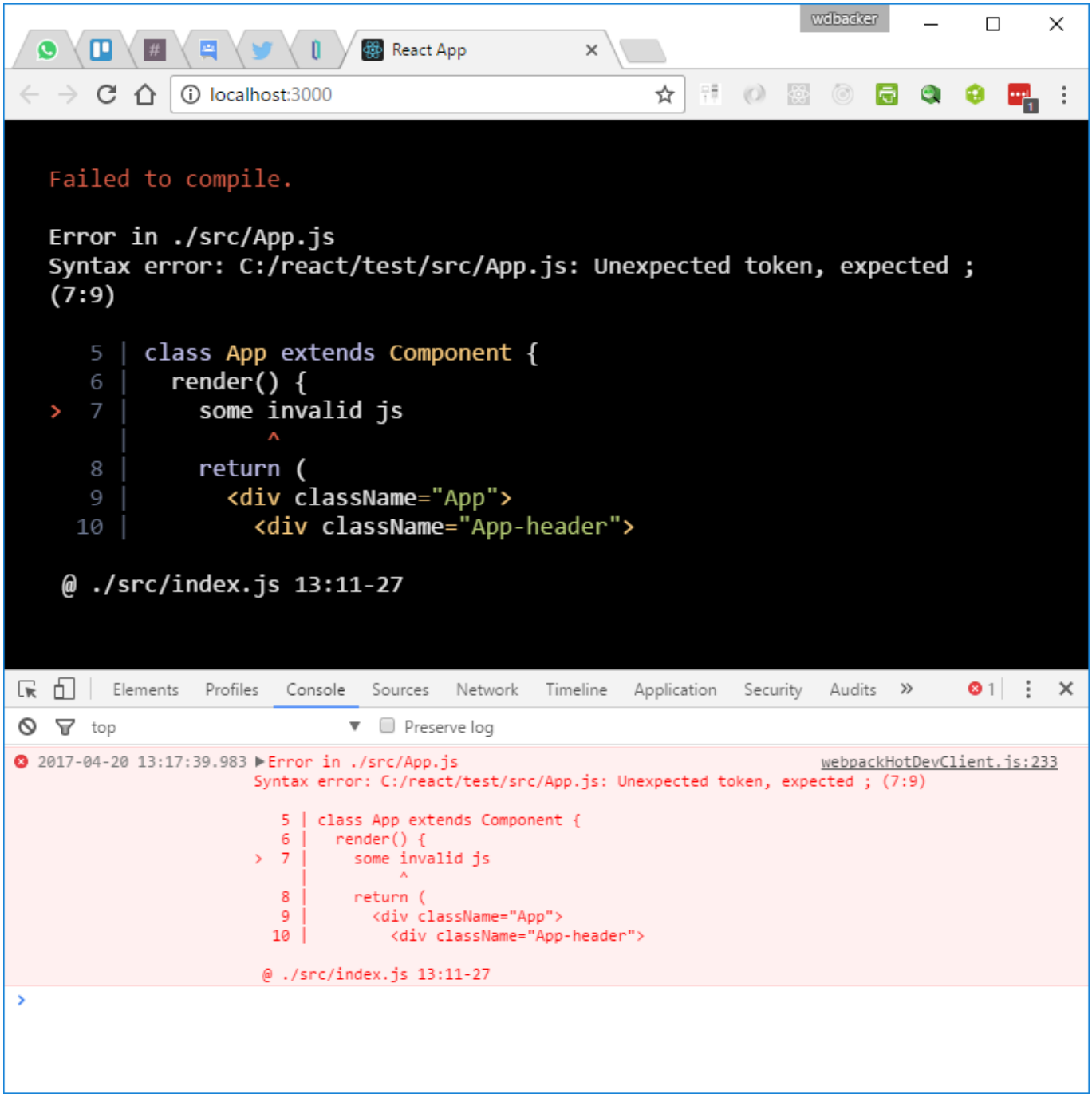


Y mira lo que pasa en tu navegador...



Bastante útil para las depuraciones, ¿ no crees? Tienes un entorno completo para la depuración, desarrollo e implementación con los scripts estándares create-react-app.

Ahora abre también las Devtools de Chrome con Ctrl+Shift+I:



Como puedes ver, el error también se muestra claramente en el espacio de trabajo de Devtools.

También existen algunas otras herramientas que podemos añadir a Atom y Chrome para crear una experiencia de depuración en React que sea mucho más potente.

En tu editor Atom, ve a la pestaña File, después a Settings y selecciona +Install. Busca el paquete correspondiente a "react" en la comunidad e instálalo. Este paquete sabe todo lo que hay que saber sobre la sintaxis JSX en React, con énfasis en la sintaxis.

En Chrome, instala la extensión "React Developer Tools". Cuando se haya instalado, fíjate en Devtools en Chrome: podrás ver que se añadió una pestaña de depuración en React, haz clic sobre ella. Ahora podrás examinar todos los componentes de React (al igual que las modificaciones y actualizaciones), ¡ y depurarlos al nivel de React (no solo en HTML)!

Ahora tenemos un entorno de programación en React completo usando Node.js para desarrollar tu front-end. Si deseas saber más sobre React y aprender sus principios básicos, te sugiero que eches un vistazo a los tutoriales que se encuentran disponibles en la página web de React.

En la parte 2 de esta serie de artículos, configuraremos nuestro back-end en Node.js/Caché (sí, ¡ verás que puedes escribir tu propia aplicación completa de Caché en JavaScript!).

#JavaScript #JSON #Node.js #React #Caché

URL de fuente: https://es.community.intersystems.com/post/nodejs-c%C3%B3mo-crear-una-aplicaci%C3%B3n-web-b%C3%A1sica-con-react-parte-1