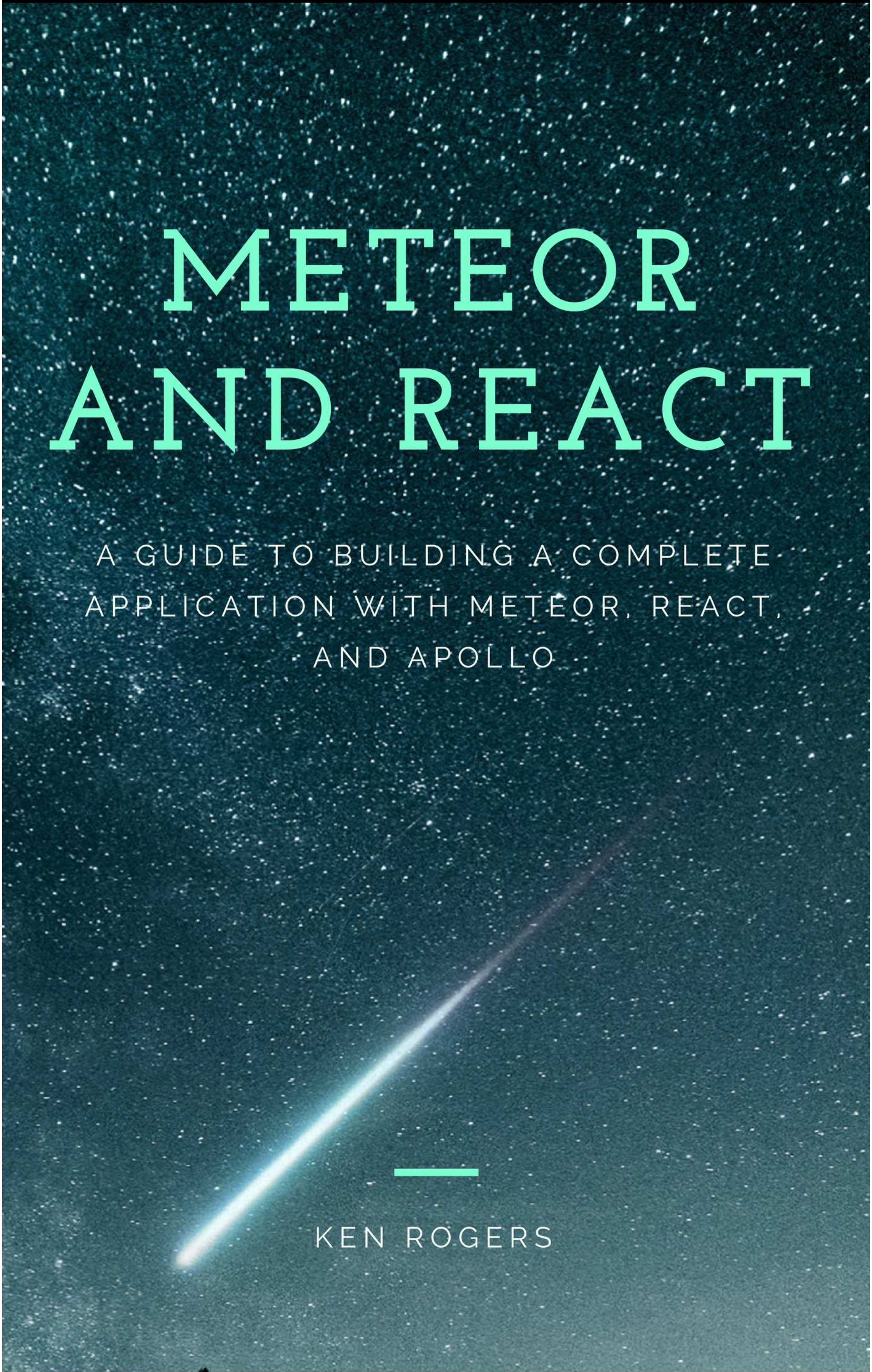

METEOR AND REACT

A GUIDE TO BUILDING A COMPLETE APPLICATION WITH METEOR, REACT, AND APOLLO

KEN ROGERS

Meteor and React

Build a Complete Application Using Meteor 1.5 and React

Ken Rogers

Contents

Preface

This book is my attempt at creating a go-to resource for learning how to develop applications with Meteor and React. I wanted to make it accessible to everyone, which is why I am making the HTML version free.

Meteor and React are fantastic platforms, and work very well together, but it can be a bit hard to get going with them. I felt a comprehensive start-to-finish tutorial was missing, so I made one.

I'd like to thank Michael Hartl of the Rails Tutorial for paving the way for this style of book writing and marketing, and for creating Softcover, the software that allowed me to make this book. If it wasn't for him, this whole thing would have taken way more time and probably wouldn't have turned out very well.

I'd also like to thank MDG and Facebook for making great tools like Meteor, Apollo, React, and React Router without which this book would not exist.

Finally, I'd like to thank Sacha Greif for making the Vulcan toolkit, which has turned Meteor development into everything I wanted it to be.

I really think this new version is going to be a fantastic resource for learning how to build apps with these two frameworks, and I can't wait to hear what you think and see what you make.

Let's get started.

Chapter 1

Introduction

Welcome to Meteor and React! I'm super excited to finally be getting it out to the world. I really think it's a great way to learn these two great JavaScript frameworks.

1.1 What to expect

In this first chapter I'll be going through some details about what we'll be learning in this version of the book, as well as some stuff you need to know about the book itself.

First off, I'm going with a Rails Tutorial style format for this book. That means that there will be a free HTML version with the option to pay to get the PDF, ePub and MOBI versions, as well as the option to upgrade to the screencasts.

I'll be releasing the book and screencasts on a chapter by chapter basis. The book will start off at a discounted price and then raise to the normal price after everything is finished.

If you decide to buy before it's finished, you'll get it at a cheaper price and get the content as it is completed.

The HTML version is free and will always be free, and I'll be releasing the downloadable ebook and screencasts starting off at a very discounted rate, and then increasing with every chapter release.

The book will be finished first, followed by the screencasts. I'm hoping to hit a schedule of one chapter per week until the book is finished.

No matter what time you purchase the package, you'll be sent each chapter as it is finished.

1.2 Prerequisites

This book is about at an intermediate level. You should be familiar with both Meteor and React, but don't need to be an expert in either of these. Go through the Meteor Tutorial and check out the React documentation to get an introduction to the two technologies we'll be using.

You should also be familiar with GraphQL. The basics are fine. As long as you know what it is and what it is used for, you're good to go.

You should be familair with web development in general, otherwise you may have some trouble getting through this book. Overall though, I've tried to write it in a way that is accessible to most users, while still being useful.

1.3 What we are learning

This book will cover the process of building a complete app from start to finish using Meteor and React. Specifically, we'll be using the Vulcan tool kit.

Vulcan is a framework created by Sacha Greif that sits on top of Meteor, React, React Router, and Apollo.

It brings all of these technologies together cohesively and makes the menial tasks much easier. It's also designed with flexibility in mind so you can swap out parts if you like.

The app we'll be building is called Food Plannr (cause all the cool kids spell things wrong now). This is a meal planning app that will be used to plan out meals and shopping lists. I cook for my family and always have trouble planning out good meals and shopping for the right stuff. Using this app, we can plan out our meals for the week by searching for recipes online and importing the ingredients and meals into our planner. The app will then take our recipes and distribute them throughout the week, and we can adjust if we like. It will also create a shopping list for us. We'll be covering all of the concepts necessary to understand how to build applications with Meteor and React using Vulcan, from initial project setup all the way to deployment, including a free option for deployment.

Each chapter will introduce a new piece to the application, and will include working example code so you can compare your work and see where your bugs are.

There will be exercises at the end of each chapter to help you solidify the knowledge you'll learn by working through the book. Sometimes these will be questions you should answer and sometimes they will be additions or modifications to the application.

I will not be providing the solutions to these exercises.

The exercises are meant to be a supplemental part of the book designed to help you cement the knowledge in your head. I want you to be able to solve these challenges on your own. You'll never learn more in programming than you will when you push through and solve a problem and get something working.

If you are really, really stuck on one of the exercises, email me and I'll do my best to help you out.

1.4 Teaching philosphy

I've recently been thinking and reading a lot about how to effectively teach things. There seems to be a fundamental lack of effective teaching methods in web development. There's no shortage of books, tutorials, and videos, but it always seems that they fall a little bit short and readers are left without a solid grasp of the information.

There are two things that I think can help solve this issue:

1. Reverse the normal teaching direction. Most teaching is bottom up. Courses start with the small building blocks and build up to the big picture. The main problem with this is that the big picture stuff is interesting,

and the small details can be boring without knowing what they are building up to. That's why I want to flip that around and show the big picture first.

So first we'll be looking at what we're building so you can see the app. Then we'll go through the completed code and see how it's made up. Then we'll actually walk through the process of coding the app.

1. The second concept is actually writing code and working on something yourself. That's why I highly encourage you to type everything you see in the book, don't copy and paste. In addition, work through the exercises at the end of each chapter. These exercises will challenge you to take the concepts you learned and create your own code to apply what you're learning.

I wanted to try this mainly after reading this article from fast.ai describing their approach to teaching machine learning and artificial intelligence.

1.5 Some final notes

I really think this book is going to be a fantastic introduction to developing applications with Meteor and React, and before we get started, I just wanted to let you know of a couple of things.

If you notice any typos or bugs, I would greatly appreciate a heads up. Writing a programming book is a very involved process, and it's easy to let small things slip, but I want to make this a very high-quality, go-to resource for learning Meteor. User feedback is critical in the process, and anything you can do to help would be great. **If you scroll up to the top of the page, you can hit the 'Contact Author' button at any time to send me an email.**

When it comes to support, I'll do my best to help anyone out who is struggling, but I can't always provide one on one help to everyone who needs it. I will do my best to at least respond to every email, even if I can't solve your problem.

The main goal of the example code is to make it so that if you are stuck on a bug you can't fix, you can compare your code to the example code and get past it.

With all that said, let's get going!

Chapter 2

What We Are Building

The app we're building is called Food Plannr.

Food Plannr is an app to help us plan out our meals for the week. I cook for my family, and always have a hard time planning out meals, keeping them healthy and the budget under control, and making sure I buy enough groceries but don't buy too much all at the same time.

Food Plannr is an app that will allow users to log in and plan out their meals for the week.

The app will start by asking the user to create a list of recipes. Each recipe will have a name and a url that will take the user to the recipe.

Whenever a user adds a recipe, it gets saved to their profile so they can reuse it for later weeks.

Every recipe will also have ingredients attached to it. When a recipe is added to a weekly plan, it's ingredients, along with their quantities, will automatically be added to the shopping list.

When we have our recipes for the week all chosen, we can hit a button that will automatically distribute them and take us to a week long calendar view. Here we can adjust the automatic creation if we want.

Otherwise, we can check out our shopping list and go grab the ingredients we need for the week.

When we log in to the app during the week, our dashboard will automatically show our meal plan for that day.

Soon, I'll have this app completed and available for you to download on

GitHub and try for yourself.

Chapter 3

Code Overview

Let's take a dive inside our application's code and see how everything is put together.

Don't worry if you don't understand much of this chapter. The point here is to get you introduced to the way it is set up and get you looking at how Meteor apps are built using Vulcan.

We'll go through every part of this in detail throughout the book, and it will make more and more sense as we go along.

[CHAPTER IN PROGRESS]

Chapter 4

Project Setup

4.1 Hello Vulcan

Vulcan is a tool kit built by Sacha Greif of Discover Meteor fame that sits on top of Meteor, React, React Router, and Apollo. It contains a lot of built in functionality that makes it way easier to build apps using these technologies.

The nice thing about Vulcan is that it is designed to be modular. You can use all of it or only certain parts of it. Since the technologies it uses out of the box are the most popular right now for building Meteor apps, we'll be sticking with those.

Vulcan has a lot of utilities that make using things like higher order components and creating GraphQL schemas very easy out of the box. But, in case we need some extra functionality, it's very easy to customize Vulcan's pieces to do what we need.

This is why I like it so much. We can use it to make the boring parts of coding easier without being limited by a strict framework.

One of my main gripes about developing with Meteor recently is that is is supposed to be an all in one platform, and for the most part it can be, but if you want to use the most popular and recommended tools it can still feel a bit abstract to easily put them all together.

Vulcan fixes that.

Vulcan turns Meteor development into what I always wanted it to be.

We get reactivity and JavaScript on both the server and client. And with Vulcan we can now use that with the benefits of using React on the front end and GraphQL for our data.

Vulcan makes it super easy to do so while abstracting away the super monotonous parts.

One thing that always drove me nuts was having to manually create container components and UI components. It was the same thing every single time and always seemed to give me trouble.

Vulcan has higher order components built right in, and makes it super easy to use best practices without the monotony.

4.2 Installing Vulcan

Let's get started by installing Vulcan by cloning the repo.

```
git clone https://github.com/VulcanJS/Vulcan.git foodplannr
```

Now we can `cd` into that directory and run `npm install` or `yarn` if you use the yarn package manager.

Once everything installs you can run `npm start` or `yarn start` to run the app.

If you look at the `packages.json` file you can see what the `start` command does. It's just a shortcut for running the default `meteor` command using the settings file for our project.

If you open up this file you'll see a ton of different options. We'll go through this file in a moment.

If everything worked correctly you'll see the default Vulcan movies app displaying at `localhost:3000`.