Getting Started with Gulp

Gulp is a fast and powerful JavaScript task runner that is built upon the node.js platform. It allows you to create tasks that can automate common workflow processes. Gulp.js is comparatively new and has been developed keeping in mind the issues that developers have encountered with Grunt.

Getting Started with Gulp introduces you to the world of building and testing JavaScript applications using gulp. This book is a hands-on guide to get you up to speed with gulp. It will instill you with the ability to automate several common development tasks to vastly improve your development workflow. It will get you started with gulp quickly and efficiently, providing you with core concepts of gulp, node.js, and npm.

By the end of this book, you will be able to create your very own gulp environments from scratch, create and maintain tasks and project builds, and automate your workflow with plugins and custom tasks.

Who this book is written for
If you are a developer who is new to build systems and task runners but have prior experience with web development, then this book is the right choice for you. You need to have some basic knowledge of HTML, CSS, and JavaScript.

What you will learn from this book
- Create files and directories and run applications using the command line
- Get to know the basics of node.js and npm and how they relate to gulp
- Write basic tasks that will concatenate, minify, compress, and preprocess your files
- Understand the internal working and structure of gulpfiles and gulp tasks
- Install gulp plugins and add them as project dependencies
- Gain a solid understanding of gulp to write your own custom tasks from scratch
- Discover ways to add additional functionality to improve your tasks

Create powerful automations with gulp to improve the efficiency of your web project workflow

Foreword by Eric Schoffstall, Creator of Gulp

Travis Maynard

Packt Publishing

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 "Introducing Gulp"
- A synopsis of the book’s content
- More information on Getting Started with Gulp

About the Author

**Travis Maynard** is a frontend web developer who focuses on minimalism and simplicity. He is currently creating software at The Resumator to help make the hiring process a more successful and enjoyable experience.

Prior to his work at The Resumator, he created efficient web frameworks and user experiences at West Virginia University to serve their education and marketing efforts.

In his spare time, he actively contributes to the open source software community and occasionally writes about the web on his website ([http://travismaynard.com](http://travismaynard.com)).

I would like to thank the gulp team for creating and supporting such a great tool. It has changed the lives of many developers and made our workflows far more efficient and enjoyable. More specifically, I would like to thank the team members Eric Schoffstall and Blaine Bublitz for their help with code reviews and mentoring me throughout my initial writings on gulp.

A big thanks to Sonali Vernekar, Arwa Manasawala, Neeshma Ramakrishnan, and Bharat Patil along with everyone at Packt Publishing for giving me the opportunity to write this book and mentoring me throughout the writing process.

I would also like to thank Alaina Maxwell for supporting me throughout the many late nights of writing, and for reviewing and reading through page upon page of content to offer suggestions that undoubtedly make this book far more enjoyable to read.

Last, but most certainly not least, I would like to thank you, the reader, for choosing this book as your introductory guide to gulp. I hope that it serves you well, and what you learn will benefit you for many years to come.
Getting Started with Gulp

I wrote this book to provide developers with a simple and inviting way to learn about gulp and the tools that are needed to use it. My goal is to keep the content simple and to remain aware of the intimidations that I experienced while learning gulp myself. With this in mind, I wanted to create content that never assumed too much from the reader, but also kept a steady pace for a more experienced reader to keep them engaged and ensure they learn the concepts actively.

What This Book Covers

Chapter 1, Introducing Gulp, focuses on helping you understand the languages and tools that you will use. You will learn how to use gulp to perform automated tasks for your development projects.

Chapter 2, Getting Started, focuses on getting your local environment set up by installing any software that is needed to move forward. You will learn how to use a command-line interface and take a look at the anatomy of a gulpfile.

Chapter 3, Performing Tasks with Gulp, covers how to create a set of base tasks that you will build upon in the following chapters. These base tasks include concatenation, minification, and preprocessing of your project files.

Chapter 4, Using Node.js Modules for Advanced Tasks, explores when and why to use node.js modules instead of gulp plugins in our tasks. You will learn how to create new tasks to run a static server, keep your project in sync across devices, and take advantage of node.js' module definitions in your client-side code.

Chapter 5, Resolving Issues, covers how to improve your tasks by adding better error handling, ordering your source files, and cleaning up your compiled code. Additionally, you will learn how to set up task dependencies, generate source maps, and use an external configuration file.
Development always starts off in a simple way. You come up with a great idea and then plan out how to build it. Quickly, you scaffold your project structure and organize everything to perfection. As you progress, your small idea starts to grow into a much larger application. You soon realize that your project has become heavy and bloated, and to remedy this, you perform a series of little mundane operations each time you modify your code to keep it small and efficient. Suddenly, all of these repetitive tasks seem to pull you down at the height of your coding victory! You tell yourself that there must be a better way.

The good news is that you are absolutely right. The solution to this development obstacle lies in utilizing build systems. Build systems are some of the most valuable tools in a developer's toolbox, and if you've never used one before, you're soon going to wonder how you ever worked without one.

In software development, build systems such as Make were initially used to compile code into executable formats for use in an operating system. However, in web development, we have a completely different set of practices and operations to contend with. Over the past few years, the growth of the Web has led to an increasing interest in using build systems to more capably handle the growing complexities of our applications and project workflows.

As developers, it is important for us to anticipate these growing complexities. We must do all that we can to improve our workflows so that we can build efficient projects that allow us to focus on what we do best: write great code.
In this book, we are going to explore gulp, one of the most popular JavaScript build systems available today. Instead of dropping you right into the code—abandoning you to sink or swim on your own—we will break apart the learning process into simple, understandable chunks that can be easily consumed and referenced if you get hung up at any point. All that you need to follow the instructions in this book is a general understanding of web development and how to write basic HTML, CSS, and JavaScript.

The first step toward using build systems is quite often viewed as the most intimidating, and understandably so. For years, I viewed the command line as a tool that was only beneficial to programmers and system administrators. I even resisted learning about node.js because I feared the amount of time and dedication required to study it would be greater than how much I could actually benefit from it.

These feelings of intimidation and resistance are completely normal and are felt by many developers just like you. We are overwhelmingly exposed to new tools and frameworks on a daily basis. It is our responsibility as developers to evaluate these tools to determine their overall value based on the time investment required to implement them into our projects. When it comes to some tools, developers simply don't dig deep enough to identify the parts that might be useful to them.

I've come to realize that these things aren't as complicated as we sometimes make them, but many developers are still psyching themselves out before they even really get started. It's important to remember that these tools are not too complicated or large for you to learn and use within a reasonable amount of time. They may be foreign at first, but they are not beyond your grasp and understanding.

What is gulp?

Gulp is a streaming JavaScript build system built with node.js; it leverages the power of streams and code-over-configuration to automate, organize, and run development tasks very quickly and efficiently. By simply creating a small file of instructions, gulp can perform just about any development task you can think of.

Gulp uses small, single-purpose plugins to modify and process your project files. Additionally, you can chain, or pipe, these plugins together into more complex actions with full control of the order in which those actions take place.

Gulp isn't alone though; it is built upon two of the most powerful tools available in the development industry today: node.js and npm. These tools help gulp perform and organize all of the wonderful things that it empowers us to do.
What is node.js?

Node.js, commonly referred to as node, is a powerful JavaScript platform that is built on top of Google Chrome's JavaScript runtime engine, V8. This gives us the ability to write JavaScript code on a server, or in our case, on our local machine. Using node, we now have the opportunity to write both the backend and frontend of a web application entirely in JavaScript. For the purposes of this book, we will only be using it as a means to run local tooling applications.

Node.js ships with npm, a companion package manager that facilitates the installation, storage, and creation of modular components that you can use to create applications. Together, these two tools are the engine behind how gulp operates and organizes its plugin ecosystem.

As I mentioned in the introduction, new tools such as node.js can bring about overwhelming thoughts or feelings of intimidation. This is especially true for those who focus entirely on the frontend side of development. However, when it comes to frontend, often the hardest part is just convincing yourself to get started. Sometimes, all you need is a simple project that can help build your confidence. In the following chapters, this is exactly what we are going to focus on, and soon all of that intimidation will melt away.

Why use gulp?

There are many uses for gulp, but as a newcomer, it might be difficult for you to identify how you can use it to improve your project workflow. With the ever-growing number of tools and frameworks, it becomes difficult to set aside enough time to research and choose the right one for your project or team. To better understand the benefits of gulp, let's identify a few of the defining reasons why to use it and what sets it apart from similar tools.

Project automation

First and foremost, the ability to automate your workflow is incredibly valuable. It brings order to the chaotic amount of tasks that need to be run throughout development.

Let's imagine that you recently developed a big application, but instead of being able to allow the necessary time to put together a proper build system, you were pressured into completing it within an incredibly short timeframe.
Introducing Gulp

Here's an example of this: For the past few days, your boss has been gathering feedback from users who claim that slow load times and performance issues are preventing them from getting their work done and damaging their user experience. It has become so frustrating that they have even threatened to move to another competing service if the performance doesn't improve soon.

Due to the short deadline, the sacrifices that were made during development have actually caused problems for your users, and the maintenance needed to resolve those problems has now become a large burden on you and your team.

Naturally, your boss is rather upset and demands that you figure out a way to correct these issues and deliver a more performant service. Not only that, your boss also expects you to have a sustainable solution so you can provide this across all of your team's future projects as well. It's quite a burden, especially with such short notice. This is a perfect example of where gulp can really save the day.

To deliver better load times in your application, you would need to compress your overall file sizes, optimize your images, and eliminate any unnecessary HTTP requests.

You could implement a step in your workflow to handle each of these manually, but the problem is that workflows often flow forward and backward. No one is infallible, and we all make mistakes. A big part of our job is to correct our mistakes and fix bugs, which requires us to take a step back to resolve any issues we run into during development.

If we were to plan out a step in our workflow to handle these items manually, it would become a huge burden that would most likely take up much of our time. The only practical way to handle optimizations like these is to automate them as an ongoing workflow step. Whether we are just starting, finishing up, or returning to our code to fix bugs, our optimizations will be handled for us.

While things like these should usually be part of your initial project setup, even as an afterthought, gulp makes resolving these issues incredibly easy. Also, it will set you up with a solid base that you can include in future projects.

There are many additional tasks that we can add to our list of automations. These include tasks such as CSS preprocessing, running an HTML server, and automatically refreshing your browser window upon any changes to your code. We will be covering all of those and more in the upcoming chapters.
Streams
At the heart of gulp is something called streams, and this is what sets it apart from other JavaScript build systems. Streams were originally introduced in Unix as a way to "pipe" together small, single-purpose applications to perform complex, flexible operations. Additionally, streams were created to operate on data without the need to buffer the entire file, leading to quicker processing. Piping these small applications together is what is referred to as a pipechain. This is one of the core components of how we will organize and structure our tasks in gulp.

Like Unix, node.js has its own built-in stream module. This stream module is what gulp uses to operate on your data and perform tasks. This allows developers to create small gulp plugins or node modules that perform single operations and then pipe them together with others to perform an entire chain of actions on your data. This gives you full control over how your data is processed by allowing you to customize your pipechain and specify how and in what order your data will be modified.

Code over config
Another reason why many developers find gulp to be a more natural alternative to other JavaScript build systems is because the build file you create is written in code, not config. This may be a matter of personal preference, but I know that this was a fundamental reason why I chose to use gulp over other build systems.

As I mentioned before, by learning more about gulp, you are also learning the basics of node.js, simply because you're writing code for a node.js application. With a build system that uses a config file, you're missing out on the value of learning the core code syntax of the platform you are using.
Summary

In this chapter, we learned about the importance of build systems in software development and the growth of interest for their usage in modern web development workflows.

As we introduce new tools such as preprocessors and JavaScript libraries, we should have a way to properly organize those files into an efficient workflow and build them for production-ready releases.

We discussed the tools that we will be using throughout the rest of the book and how they all work together and interact with one another to provide us with a solid build system solution that we can use for our projects.

With a basic understanding of these tools and their uses, we can now begin to learn how to set up our local environment for gulp. In the upcoming chapter, we will learn about our command-line application, install our software, and prepare our project to begin writing code.
Where to buy this book

You can buy Getting Started with Gulp from the Packt Publishing website.
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.
Click here for ordering and shipping details.