

Quick Build System with Gulp

Copyright Blurb

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means including information storage and retrieval systems – except in the case of brief quotations in articles or reviews – without the permission in writing from its publisher, **Sandeep Kumar Patel**.

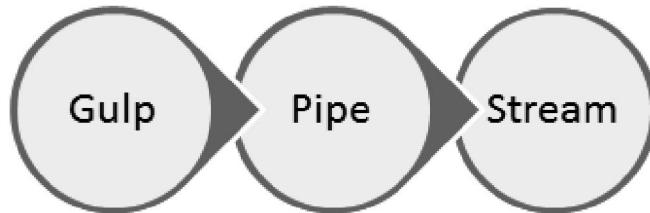
All brand names and product names used in this book are trademarks, registered trademarks, or trade names of their respective holders. I am not associated with any product or vendor in this book.

Published By

Sandeep Kumar Patel.

1

Introducing Gulp System



Gulp is new build system is capable of automate minification, file copy, watching file changes and rerun tasks. Gulp is built on top of the NodeJS and is available as open source. In this chapter, we will learn to getting started with Gulp system.

You can find the Gulp source code in the following link:-
<https://github.com/gulpjs/gulp/>

Gulp Installation

To install Gulp we need to have NPM installed in the machine. Considering NPM is installed in the machine let's continue with the example. We have created a directory named **HelloWorldApp** for this demo. Let's initialize **package.json** file using **npm init** command. The following screenshot shows the terminal with initialization of **package.json** file.

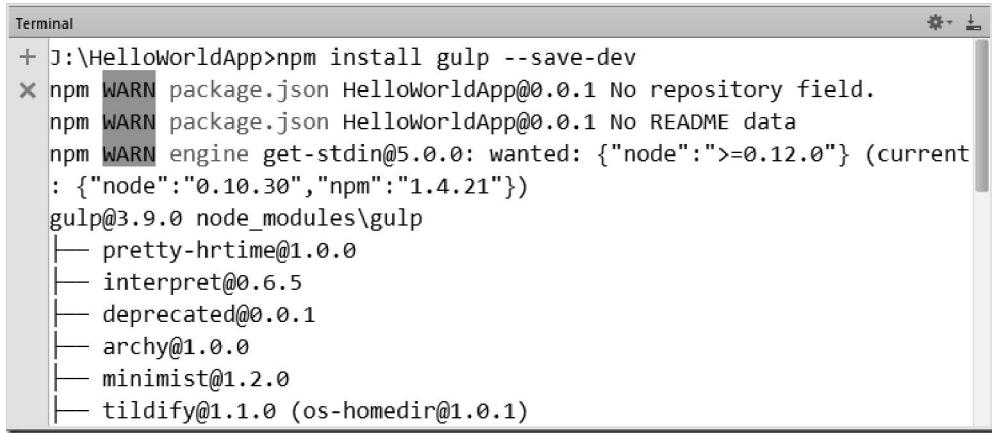
The screenshot shows a terminal window with the following text:

```
Terminal
+ J:\HelloWorldApp>npm init
✖ This utility will walk you through creating a package.json file
.
It only covers the most common items, and tries to guess sane defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package
and
```

Gulp can be installed as local development dependencies using **npm install gulp --save-dev** command. The following screenshot shows the terminal with the installation of Gulp build system.



```
Terminal
+ J:\HelloWorldApp>npm install gulp --save-dev
× npm WARN package.json HelloWorldApp@0.0.1 No repository field.
npm WARN package.json HelloWorldApp@0.0.1 No README data
npm WARN engine get-stdin@5.0.0: wanted: {"node":">=0.12.0"} (current
: {"node":"0.10.30","npm":"1.4.21"})
gulp@3.9.0 node_modules\gulp
|   +-- pretty-hrtime@1.0.0
|   +-- interpret@0.6.5
|   +-- deprecated@0.0.1
|   +-- archy@1.0.0
|   +-- minimist@1.2.0
|   +-- tildify@1.1.0 (os-homedir@1.0.1)
```

Creating Gulp Task

After successful installation of Gulp, the next step is to create **gulpfile.js** file.

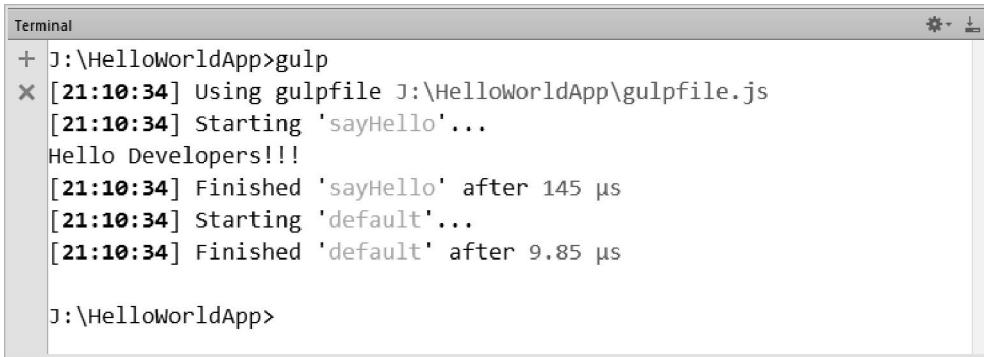
The **gulpfile.js** file contains the code to define the Gulp task. We can call the **gulp** using **require()** function and create a task using **task()** method.

Let's create a task named **sayHello** which is a sample task and prints a text message in console. We can make this task as default task using **default** key word. The code content of **gulpfile.js** are as follows: -

```
var gulp = require('gulp');
gulp.task('sayHello', function() {
  console.log("Hello Developers!!!");
});
gulp.task('default',['sayHello']);
```

Running Gulp Task

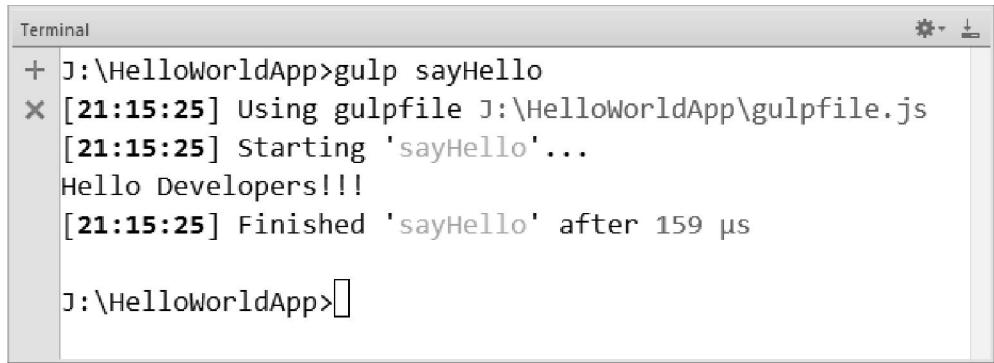
Now we can run the **default** task using Gulp command. The following screenshot shows the terminal with gulp task in execution.



```
Terminal
+ J:\HelloWorldApp>gulp
× [21:10:34] Using gulpfile J:\HelloWorldApp\gulpfile.js
[21:10:34] Starting 'sayHello'...
Hello Developers!!!
[21:10:34] Finished 'sayHello' after 145 µs
[21:10:34] Starting 'default'...
[21:10:34] Finished 'default' after 9.85 µs

J:\HelloWorldApp>
```

We can also execute the **sayHello** task using **gulp sayHello** command. The following screenshot shows the terminal with gulp task in execution.



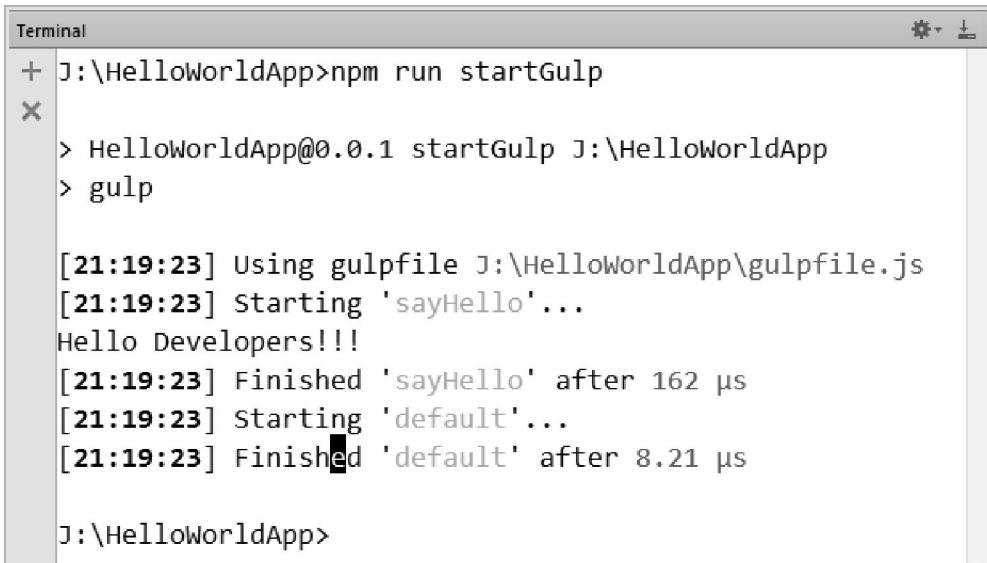
```
Terminal
+ J:\HelloWorldApp>gulp sayHello
✖ [21:15:25] Using gulpfile J:\HelloWorldApp\gulpfile.js
[21:15:25] Starting 'sayHello'...
Hello Developers!!!
[21:15:25] Finished 'sayHello' after 159 µs

J:\HelloWorldApp>
```

We can also call gulp though NPM by adding an entry to **package.json** using script property. The code content of the **package.json** file are as follows:-

```
{
  "name": "HelloWorldApp",
  "version": "0.0.1",
  "description": "Getting Started With Gulp",
  "scripts": {
    "startGulp": "gulp"
  },
  "author": "Sandeep",
  "license": "ISC",
  "devDependencies": {
    "gulp": "^3.9.0"
  }
}
```

Now we can execute the gulp task using **npm run startGulp** command. The following screenshot shows the terminal with Gulp task in execution using NPM module.

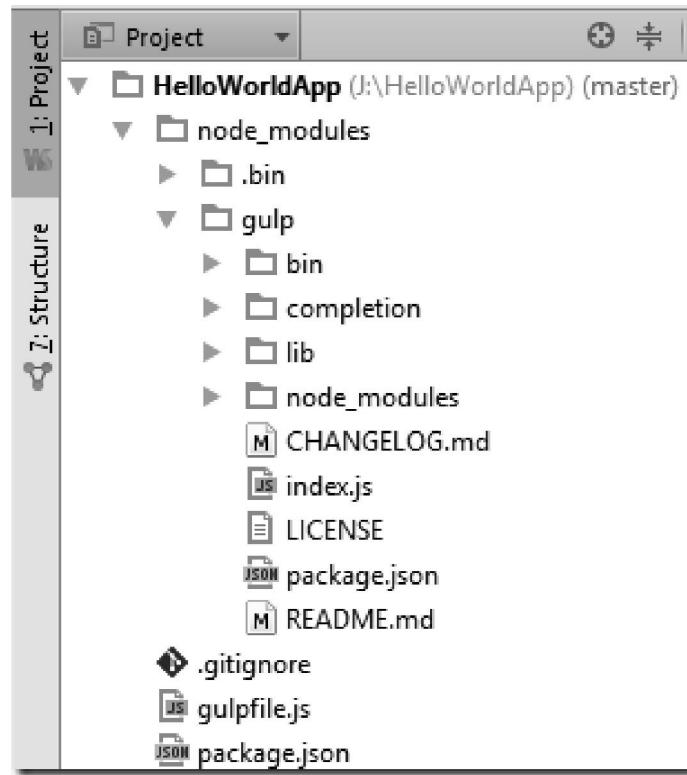


```
Terminal
+ J:\HelloWorldApp>npm run startGulp
✖
  > HelloWorldApp@0.0.1 startGulp J:\HelloWorldApp
  > gulp

[21:19:23] Using gulpfile J:\HelloWorldApp\gulpfile.js
[21:19:23] Starting 'sayHello'...
Hello Developers!!!
[21:19:23] Finished 'sayHello' after 162 µs
[21:19:23] Starting 'default'...
[21:19:23] Finished 'default' after 8.21 µs

J:\HelloWorldApp>
```

The following screenshot shows the updated project structure for **HelloWorldApp** directory. We can see a **gulp** directory is created inside the **node_modules** directory.



Summary

In this chapter we have introduced with the Gulp build system. We have also learnt to install Gulp and created a sample task. In the coming chapters we will learn more about Gulp.

You can find the complete code for this chapter in the following link:-

<https://github.com/saan1984/HelloWorldApp>