

| | |
|----------|--|
| عنوان: | Gulp #2 |
| نویسنده: | م محمد شریفی |
| تاریخ: | ۱۰:۴۵ ۱۳۹۴/۰۶/۱۷ |
| آدرس: | www.dotnettips.info |
| گروه‌ها: | JavaScript, Twitter Bootstrap, bower, Gulp |

در [قسمت قبلی](#) بحث کردیم که گالپ چیست و چه کاربردی دارد و در نهایت آن را بر روی سیستم خود نصب کردیم. در این مقاله و مقالات بعد می‌خواهیم کار خود را با راه اندازی یک workflow برای بوت استرپ، روند شخصی سازی آن را بسیار آسان و لذت بخش‌تر کنیم. امیدوارم که برای ادامه‌ی این بحث هیجان انگیز آماده باشید!

ساخت پروژه گالپ

ابتدا یک پوشه‌ی دلخواه به نام **project** را درست کنید. سپس خط فرمان خود را به این مسیر تغییر دهید و در نهایت دستور زیر را وارد کنید:

```
npm init
```

این دستور برایمان یک فایل *package.json* می‌سازد تا هم مشخصات پروژه را مثل نام، ورژن، نام توسعه دهنده، مخزن و ... مشخص کنیم و هم وابستگی‌های آن را، تا توسعه دهندگان دیگر، هنگام استفاده از پروژه، با مشکل مواجه نشوند و فقط با اجرای دستور `npm install` تمام وابستگی‌های پروژه را نصب کنند. حتما مشاهده کرده‌اید که این دستور چند سوالی را از شما می‌پرسد. برای نمونه من آنها به این صورت پاسخ می‌دهم و در نهایت از من یک تایید می‌گیرد که **yes** را می‌زنم.

```
name: (Gulp-RTLbootstrap-fontawesome)
version: (1.0.0)
description: An Awesome workflow
entry point: (index.js) index.html
test command: test
git repository: https://github.com/mmdsharifi/gulp-rtlBootstrap-fontawesome.git
keywords: gulp, rtlbootstrap, persian bootstrap
author: Mohammad Sharifi
license: (ISC) MIT
```

الان فایل *package.json* درست شده و چون ما در این پروژه می‌خواهیم از گالپ استفاده کنیم، یکی از وابستگی‌های پروژه‌ی ما گالپ خواهد بود. بدین معنی که اگر بخواهیم پروژه را توسعه دهیم و گالپ نصب نشده باشد، با مشکل مواجه می‌شویم.

نصب گالپ

در خط فرمان دستور زیر را وارد کنید تا گالپ در این پروژه نصب شود.

```
npm install gulp --save-dev
```

تفاوتی آن با دستوری که در مقاله‌ی قبلی اجرا کردیم، این است که این دستور فقط گالپ را در مسیر جاری در فولدر `node_modules` نصب می‌کند و **--save-dev** آن را به وابستگی‌های توسعه‌ی پروژه در فایل *package.json* اضافه می‌کند. گام بعدی، ساخت فایلی است که گالپ به آن نیاز دارد تا با استفاده از آن، تسک‌ها و کارهایی را که برایش نوشته‌ایم، از آن بخواند و اجرا کند.

ایجاد فایل *gulpfile.js*

این فایل را یا به صورت دستی ایجاد کنید یا با خط فرمان با دستور ذیل:

```
touch gulpfile.js
```

و حالا آن را در ویرایشگر مورد علاقه‌ی خود باز کنید. من از ویرایشگر [Atom](#) استفاده می‌کنم. اما notepad هم کفایت می‌کند.

نوشتن اولین تسک گالپ

در ویراشگر خط زیر را می‌نویسیم:

```
var gulp = require('gulp');
```

require به Node می‌گوید که به فولدر node_modules برای پکیج gulp نگاه کند. زمانیکه آن را پیدا کرد، آن را به متغیر gulp انتساب می‌دهیم تا از تابع‌های گالپ بتوانیم استفاده کنیم. حال می‌خواهیم اولین تسک خود را بنویسیم:

```
gulp.task('task-name', function() {
  // Stuff here
});
```

گالپ دارای [۴ تابع](#) task, src, dest, watch است که با آنها آشنا خواهیم شد. task یک کار را برای گالپ تعریف می‌کند و [سه پارامتر](#) دارد. اولی نام تسک، دومی (اختیاری) وابستگی این تسک (بدین معنا که اول باید این وظیفه اجرا شود، سپس تسک جاری) و در نهایت تابع درون تسک‌ها را می‌نویسیم. برای مثال:

```
gulp.task('hello', function() {
  console.log('Hello Gulp !');
});
```

فایل را ذخیره کنید. می‌خواهیم به گالپ بگوییم که این وظیفه را انجام دهد. کافی است **gulp hello** را در خط فرمان وارد کنیم. نتیجه به صورت زیر خواهد بود:

```
mmdsharifi@mmdsharifi ~/Work/Gulp-RTLbootstrap-fontawesome $ gulp hello
[09:10:44] Using gulpfile ~/Work/Gulp-RTLbootstrap-fontawesome/gulpfile.js
[09:10:44] Starting 'hello'...
hello Gulp !
[09:10:44] Finished 'hello' after 83 μs
```

البته که تسک‌هایی که برای گالپ می‌نویسیم، کاراتر از این است؛ برای مثال:

```
gulp.task('task-name', function () {
  return gulp.src('source-files') // Get source files with gulp.src
    .pipe(aGulpPlugin()) // Sends it through a gulp plugin
    .pipe(gulp.dest('destination')) // Outputs the file in the destination folder
});
```

با استفاده از متد src به گالپ می‌گوییم که مسیر مبدأ فایل‌ها، برای انجام تسک کجا است و dest هم بعد از انجام تسک، فایل‌های خروجی را به مقصد مشخص می‌برد. متد pipe یک تابع ند جی اس است که مطابق [مستندات](#) خودش متدی است که تمام [جریان](#)‌های قابل خواندن را واکنشی می‌کند و به مسیری [که به صورت آرگومان به عنوان مقصد] داده شده، هدایت می‌کند. شاید در ابتدا نوشتن تسک برایتان کمی پیچیده باشد، اما بعد از ساخت اولین پروژه با گالپ، خواهید دانست که تسک نوشتن برای گالپ کاری بسیار آسان و شیرین است!

<https://github.com/mmdsharifi/gulp-rtlBootstrap-fontawesome> در گیت هاب :

نام کامیت این قسمت: Init commit

در مقاله بعدی gulp را در کنار [bower](#) بکار خواهیم برد. بهتر است مطالعه‌ای در مورد bower نیز انجام دهید. (پیشنهاد: [+ و +](#))

نظرات خوانندگان

نویسنده: محمود راستین
تاریخ: ۱۹:۹ ۱۳۹۴/۰۷/۰۲

تشکر بابت این مقاله. فقط یک نکته رو بیان کنم که فرمان برای پلتفرم ویندوز نیست. اگر دستور :

```
touch gulpfile.js
```

رو صادر کنیم با خطای زیر رو به رو میشیم :

```
'touch' is not recognized as an internal or external command, operable program or batch file.
```

برای ایجاد فایل خالی در ویندوز باید اینکارو بکنیم :

```
echo $null >> gulpfile.js
```

این دستور همانند دستور touch هستش. با این دستور فایل gulpfile.js در مسیر پروژه ساخته میشه.

نویسنده: م محمد شریفی
تاریخ: ۹:۲۳ ۱۳۹۴/۰۷/۰۳

خواهش می‌کنم.

بله درسته دستور touch در لینوکس به درستی کار می‌کند.

سپاس از اشتراک گذاری این نکته.