

Gestor de dependencias Bower



En este post vamos a hablar de la gestión de dependencias para los desarrolladores frontend. Ya hablamos de la gestión de dependencias para backend en un [artículo anterior sobre Composer](#). En este artículo veamos como pudíamos usar esa herramienta para gestionar las dependencias de paquetes PHP. El problema que surge con Composer es que no siempre es adecuado para gestionar dependencias cuando se trata de frameworks JavaScript o CSS (en realidad, no es ese su objetivo, sino que está preparado para librerías y frameworks PHP). Bower es el complemento perfecto para tener en nuestro equipo todo lo que necesitemos en lo que a gestión de dependencias se refiere.

Bower nos va a permitir instalar (y mantener actualizados con mucha facilidad) frameworks como [jQuery](#), [jQueryUI](#) o [Bootstrap](#), entre otros, así como las dependencias secundarias que estos puedan llegar a necesitar.

REQUISITOS PREVIOS E INSTALACIÓN

Para poder instalar Bower en nuestro equipo debemos tener previamente instalados Node.js y Git. Podemos [descargar Node.js de su página oficial](#). En un artículo posterior hablaremos de qué es, y para que nos puede servir Node.js. De momento, bajamos la última versión (la 6.4.0 en el momento de escribir esto) y la instalamos. La instalación es muy sencilla. Doble clic en el archivo node-v6.4.0-x64.msi que se ha descargado y la típica instalación de las aplicaciones de Windows (aceptar, siguiente, siguiente...). No cambies las opciones durante la instalación. Una vez terminada, puedes, si lo deseas, crearte un acceso directo al escritorio aunque, por ahora al menos, no te hace falta, ya que su uso será transparente para nosotros en lo que se refiere a este artículo. En cuanto a Git, ya lo conoces un poco si has leído los artículos que publicamos al respecto. Si no, para saber lo que necesitamos, y cómo instalarlo, puedes leer [este artículo](#).

Y ahora vamos a instalar Bower, que ya toca. Para ello abre una consola de Git-bash en tu ordenador. Te sugiero que la abras en la carpeta de localhost, ya que esta será la raíz de todos nuestros proyectos web. Si tienes instalado XAMPP, tu carpeta de localhost es `C:/xampp/htdocs/`. Si tienes instalado otro sistema de servidor local, la ruta será diferente. Si no tienes servidor local este es el momento de instalar uno. Puedes ver cómo hacerlo en [este artículo](#).

En la consola de Git-bash teclea lo siguiente:

```
npm install -g bower
```

Veamos que hace esto. La instrucción `npm` llama al gestor de paquetes de [Node.js](#). El comando `install` es la orden que le damos a `npm` indicándole que va a instalar "algo". El parámetro `-g` se refiere a **global**, para que lo que vayamos a instalar esté disponible en cualquier parte del equipo, y para todos los usuarios del mismo. Por último, incluimos `bower`, que es el nombre de la aplicación que queremos que `npm` nos instale.

Pulsamos la tecla <INTRO> y `npm` se hace cargo de instalarnos Bower para que podamos usarlo desde cualquier parte del equipo. Veremos en la consola unas indicaciones sobre el proceso de instalación. Dado que salen algunos caracteres con un aspecto algo "extraño", puede parecer que ha fallado algo. No te preocupes. No ocurre nada. Se está descargando e instalando Bower. Tras unos segundos, verás que tu consola muestra un aspecto mucho más tranquilizador informándote de que la instalación ha concluido, como se muestra a continuación:

```
Jose@Jose MINGW64 /c:/xampp/htdocs
$ npm install -g bower
C:\Users\Jose\AppData\Roaming\npm -> C:\Users\Jose\AppData\Roaming\npm\node_modules\bower\bin\
C:\Users\Jose\AppData\Roaming\npm -- bower@1.7.9
Jose@Jose MINGW64 /c:/xampp/htdocs
$
```

ALGO MÁS SOBRE npm

Con lo que hemos hecho ya tenemos el gestor de dependencias Bower instalado y listo para su uso. Sin embargo, vamos a comentar algo más sobre npm. Aunque este no es un artículo específico sobre esta herramienta, si hay algunas cosas que quiero que "tengas a

mano".

Lo primero es sobre la actualización. Si queremos, por ejemplo, actualizar Bower a la última versión, teclearemos:

```
npm update -g bower
```

Si queremos desinstalarlo, podemos usar:

```
npm uninstall -g bower
```

Existen muchas más instrucciones [npm](#). Podemos ver en consola la lista detallada tecleando:

```
npm -l
```

En todo caso, como te he comentado. Esto ya es tema para otro artículo.

USANDO BOWER

Ya tenemos Bower instalado en nuestro equipo y, por supuesto, la forma de aprender a usarlo es usándolo. Como ya sabemos, Bower se va a ocupar de las dependencias que necesitemos para el frontend de un proyecto web. Para ello vamos a crear en nuestro localhost una carpeta donde alojaremos nuestro proyecto de pruebas. La llamaremos [pruebas_bower](#). Para ello, teclea, en la consola Git-bash lo siguiente:

```
mkdir pruebas_bower
```

Ahora, entra en el directorio así:

```
cd pruebas_bower
```

El prompt de tu consola tiene ahora el aspecto que ves a continuación, indicando que ya estamos dentro del directorio adecuado.

```
José@Jose MINGW64 /c/xampp/htdocs/pruebas_bower
$
```

En nuestro proyecto queremos instalar Bootstrap, jQuery y jQueryUI, que serán las herramientas que hemos decidido que vamos a usar en la aplicación que desarrollaríamos en este ejemplo. Empezaremos por Bootstrap, tecleando, en la línea de mandatos, lo siguiente:

```
bower install bootstrap
```

Tras unos segundos vemos la respuesta de Bower en la consola, así:

```
José@Jose MINGW64 /c/xampp/htdocs/pruebas_bower
$ bower install bootstrap
bower bootstrap#*          cached https://github.com/twbs/bootstrap.git#3.3.7
bower bootstrap#*          validate 3.3.7 against https://github.com/twbs/bootstrap.git#*
bower jquery#1.9.1 - 3      cached https://github.com/jquery/jquery-dist.git#3.1.0
bower jquery#1.9.1 - 3      validate 3.1.0 against https://github.com/jquery/jquery-dist.git#1.9.1
bower bootstrap#3.3.7       install bootstrap#3.3.7
bower jquery#1.9.1 - 3      install jquery#3.1.0

bootstrap#3.3.7 bower_components\bootstrap
└─ jquery#3.1.0
jquery#3.1.0 bower_components\jquery
```

Esto nos dice que se ha instalado correctamente Bootstrap 3.3.7, que es la última versión disponible en el momento de escribir esto. Si hubiéramos deseado una versión anterior, deberíamos haberlo especificado (en seguida veremos cómo hacerlo). Pero nos dice más cosas. Como resulta que la parte JavaScript de Bootstrap (la que se ocupa de modales y otros complementos) requiere para funcionar de jQuery, Bower también ha instalado ya jQuery, por lo que no vamos a necesitar instalarlo aparte. Esa es la tarea de un gestor de dependencias, y Bower la realiza estupendamente bien.

Si ahora miras en la carpeta de tu proyecto, verás que tienes una carpeta llamada [bower_components](#) y, dentro de ella, tienes otras dos: [bootstrap](#) y [jquery](#), con las últimas versiones disponibles de estos frameworks.

Ahora vamos a instalar jQueryUI, que hemos dicho que también vamos a querer disponer de él para nuestro proyecto. Como bootstrap no depende específicamente de jQueryUI, no ha sido instalado antes, así que debemos instalarlo aparte, así:

```
bower install jqueryui
```

En nuestro proyecto, dentro de la carpeta [bower_components](#), tenemos ahora también la carpeta [jquery-ui](#).

Ahora ya sólo tenemos que incluir en nuestras páginas las correspondientes llamadas a hojas de estilo y códigos JavaScript de las dependencias que hemos instalado, cómo vemos en el siguiente ejemplo de lo que podría ser una página de nuestro proyecto.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
```

```
<title>Página de mi proyecto</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="bower_components/bootstrap/dist/css/bootstrap.min.css">
<link rel="stylesheet" href="bower_components/jquery-ui/themes/base/base.css">
</head>
<body>
<!-- Código de la página -->

<script src="bower_components/jquery/dist/jquery.min.js"></script>
<script src="bower_components/bootstrap/dist/js/bootstrap.min.js"></script>
<script src="bower_components/jquery-ui/jquery-ui.min.js"></script>
</body>
</html>
```

En el ejemplo hemos usado el tema base de jQueryUI, pero podríamos haber usado otro si lo deseáramos. Solo es cuestión de incluir el archivo CSS adecuado. También debes tener en cuenta que este ejemplo está basado en que la página HTML de muestra está en la raíz del proyecto. Si estuviera en otra ruta deberías tener esto en cuenta al referenciar hojas de estilo y códigos JavaScript.

OTRA FORMA DE INSTALAR PAQUETES

Vamos a conocer una forma más estructurada de usar Bower. Lo primero que vamos a hacer es eliminar de nuestro directorio de proyecto la carpeta `bower_components` y cualquier otra cosa que hayamos añadido, para dejarlo vacío y empezar de nuevo de cero. En este segundo ejemplo sólo instalaremos Bootstrap y Font Awesome, por cambiar un poco.

Lo que vamos a hacer es crear un archivo llamado `bower.json` para especificar en este formato las dependencias necesarias para nuestro proyecto. Para ello tecleamos:

`bower init`

Al cabo de unos segundos Bower empezará a preguntarnos algunos datos generales acerca de nuestro proyecto. Podemos teclearlos o pulsar la tecla `<INTRO>` sin más, ya que no todos son imprescindibles. Lo primero que nos pregunta es el nombre del proyecto, y nos sugiere, enter paréntesis, el nombre de la carpeta en la que estamos. Podemos poner otro nombre, o no poner ninguno.

Puedes ir pulsando `<INTRO>` en cada dato que te pregunte Bower, hasta que llegues a la pregunta que te muestro a continuación:
`set currently installed components as dependencies? (y/N)`

La respuesta en mayúscula (la `N`, en este caso) es la que se tomará por defecto si pulsas directamente `<INTRO>`. La pregunta se refiere a que si existe ya previamente una carpeta `bower_components` con algún componente preinstalado de antes, podemos hacer que sea una dependencia de este proyecto o no. Cómo no hay esta carpeta, porque hemos empezado de nuevo en limpio, podemos aceptar la respuesta por defecto. La siguiente pregunta es:

`add commonly ignored files to ignore list? (Y/n)`

En este caso, la respuesta por defecto es `Y`. Esto hará que los ficheros que normalmente no se usan para un proyecto en producción sean ignorados. Para que nos aclaremos (y esto es importante que lo sepas). Bower no instala los paquetes que le pides de la página oficial de cada fabricante, sino del repositorio GitHub que dicho fabricante tiene para su producto. Algunos ficheros que aparecen en GitHub, como `README.md`, no son, en absoluto, necesarios para nuestro proyecto, así que aceptando esta opción, Bower los ignorará. Además ignorará otros archivos y carpetas que nuestro proyecto no necesita.

Y la siguiente pregunta:

`would you like to mark this package as private wich prevents it from being accidentally published to the registry? (Y/n)`

Esto tiene que ver con los paquetes que se suben a repositorios privados de Github. Para las dependencias de nuestros proyectos no nos afecta. Acepta la opción por defecto (`Y`). Si, por defecto te diera la opción `N`, pulsa `Y` antes de pulsar `<INTRO>`.

A continuación nos muestra un archivo en formato JSON y nos pregunta si nos parece correcto:

`Looks good? (Y/n)`

Lo aceptamos como bueno y nos encontramos que en nuestra carpeta raíz del proyecto hay un archivo llamado `bower.json`, cuyo código (puedes verlo abriéndolo con un editor de texto plano) es el siguiente:

```
{
  "name": "pruebas_bower",
  "authors": [
    "José López Quijado <jquijado@eldesvandejose.com>"
  ]
}
```

```
],  
"description": "",  
"main": "",  
"license": "MIT",  
"homepage": "",  
"private": true,  
"ignore": [  
  "**/*.*",  
  "node_modules",  
  "bower_components",  
  "test",  
  "tests"  
]  
}
```

Hasta ahora, esto es lo único que tenemos en nuestra carpeta del proyecto. Ahora vamos a hacer que Bower instale las dependencias que queremos (recuerda que, para este ejemplo hemos hablado de usar Bootstrap y Font Awesome, y que Bower instalará para Bootstrap, además, jQuery). Lo vamos a hacer de una forma parecida a la que usamos en el ejemplo anterior, pero añadiendo la opción `--save`. Esto hará que las dependencias se hagan constar en el archivo `bower.json` que hemos creado. La instrucción necesaria podría ser la siguiente:

```
bower install --save bootstrap fontawesome
```

Al igual que antes, se ha creado la carpeta `bower_components`, con todo lo que necesitamos. Pero, además, se ha modificado el archivo `bower.json`. Su contenido, ahora, es el siguiente:

```
{  
  "name": "pruebas_bower",  
  "authors": [  
    "José López Quijado <jquijado@eldesvandejose.com>"  
  ],  
  "description": "",  
  "main": "",  
  "license": "MIT",  
  "homepage": "",  
  "private": true,  
  "ignore": [  
    "**/*.*",  
    "node_modules",  
    "bower_components",  
    "test",  
    "tests"  
  ],  
  "dependencies": {  
    "bootstrap": "3.3.7",  
    "font-awesome": "fontawesome#4.6.3"  
  }  
}
```

Observa las líneas resaltadas. Bajo el epígrafe `dependencies` (dependencias) se han anotado las dependencias que le hemos pedido al proyecto. No consta jQuery porque no se lo hemos pedido específicamente, aunque, por supuesto, Bower lo ha descargado, ya que Bootstrap lo necesita.

Y ¿para qué nos sirve esto? Bien. Suponte que ahora vas a desplegar tu proyecto en otro servidor. No necesitas llevarte toda la carpeta `bower_components`. En este caso tiene poco contenido, pero podría tener mucho más. Te basta con llevarte el archivo `bower.json`. Una vez copiado este en el servidor de destino, tecleas en la consola:

bower install

De este modo, Bower lee en [bower.json](#) cuales son las dependencias, las descarga y las instala.

MÁS SOBRE BOWER

Lo mismo que nos sirve para instalar dependencias, Bower nos sirve para desinstalarlas. Suponte que, en el ejemplo anterior, decide que, en realidad, no vas a necesitar Font Awesome. Tecleas lo siguiente:

```
bower uninstall --save font-awesome
```

Observa que nos referimos al paquete [fontawesome](#) como [font-awesome](#) (con un guion medio), ya que así es como Bower ha nombrado la carpeta dentro de [bower_components](#) y la dependencia en [bower.json](#), y si no, no lo podrá desinstalar. Como hemos usado la opción `--save`, no solo elimina la carpeta, sino que actualiza el fichero [bower.json](#).

De cada dependencia que podemos instalar usando Bower no hay una sola edición disponible en Github. Podemos ver todas las que hay, para elegir. Para ello empleamos la instrucción [search](#). Por ejemplo, si te interesan opciones sobre Bootstrap, puedes teclear:

```
bower search bootstrap
```

La salida de Bower tendrá un aspecto similar al siguiente:

```
josé@Jose MINGW64 /c:/xampp/htdocs/pruebas_bower
$ bower search bootstrap
Search results:

bootstrap https://github.com/twbs/bootstrap.git
bootstrap-bo https://github.com/brentoneill/bootstrap.git
x-bootstrap https://github.com/x-element/x-bootstrap.git
dd-bootstrap https://github.com/DeloitteDigitalUK/DDBootstrap.git
bootstrap-pi https://github.com/guilhermegeek/bootstrap-pi.git
bootstrap-ie https://github.com/ddouble/bsie.git
bootstrap-ui https://github.com/visionappscz/bootstrap-ui.git
bootstrap-tl https://github.com/tomatolabs/bootstrap.git
ng-bootstrap https://github.com/igroup/ng-bootstrap.git
hg-bootstrap https://github.com/HackerGaucho/hg-bootstrap.git
cd-bootstrap https://github.com/jairorudas/cd-bootstrap.git
qk-bootstrap https://github.com/fran150/qk-bootstrap.git
ui-bootstrap https://github.com/angular-ui/bootstrap.git
bootstrap-gap https://github.com/evangraaff/bootstrap-gap.git
bootstrap-cc https://github.com/chibicode/bootstrap-cc.git
ui-bootstrap https://github.com/angular-ui/bootstrap.git
bem-bootstrap https://github.com/matmucharapna/bem-bootstrap.git
nr-bootstrap https://github.com/artbelov/nr-bootstrap.git
bootstrap-ui https://github.com/angular-ui/bootstrap.git
bootstrap-dv https://github.com/JKAussieSkater/Bootstrap-DV.git
mk-bootstrap https://github.com/geometrybase/mk-bootstrap.git
boost-bootstrap https://github.com/cristiangrojas/boost-bootstrap.git
vue-bootstrap https://github.com/demohi/vue-bootstrap.git
xui-bootstrap https://github.com/spxis/xui-bootstrap.git
ng2-bootstrap https://github.com/valor-software/ng2-bootstrap.git
bootstrap.css https://github.com/bowerjs/bootstrap.git
mmd-bootstrap https://github.com/mynewsdesk/mmd-bootstrap.git
bootstrap-tag https://github.com/gammasoft/bootstrap-tag.git
bootstrap-bower https://github.com/diegotoral/bootstrap-bower.git
xdf-bootstrap https://github.com/xdfcn/bootstrap.git
```

Esta manera de buscar paquetes puede ser, en ocasiones, un engorro, sobre todo si sabemos que necesitamos instalar tal o cual paquete que vimos una vez y creemos que es lo que nos hace falta, pero no recordamos exactamente el nombre, o como se escribía...

Uffff. Afortunadamente, existe un buscador específico para dependencias que pueden instalarse con Bower. La dirección es

<https://bower.io/search/>. Úsala y verás que es una gran ayuda.