

Manejo de paquetes de Node.js

Desarrollo de Aplicaciones Web Avanzado

Capacidad Terminal

- Implementan aplicaciones web utilizando la plataforma Node.js

Competencia específica de la sesión

- Al finalizar la sesión, los estudiantes serán capaces de identificar la importancia de NPM en node.js

Conclusiones (después de años de programación)

- Software es terrible
- La gente es terrible
- Hay muchos escenarios
- Nada funcionará al 100% en todos los escenarios
- Es probable que algo no funcione nunca del todo
- Nuestras vidas son perturbaciones sin sentido en un vórtice constante de caos y entropía



Para instalar las aplicaciones, podemos copiar los archivos que la componen en los directorios adecuados, buscar las dependencias en la versión adecuada, ver si ya están instaladas en nuestro sistema, en las rutas que la nueva aplicación espera, si no lo están hacerlo...

Instalar Nhibernate (antes)

- Buscar la página de descarga correspondiente.
- Elegir la versión correcta para nuestra versión de .NET
- Buscar posibles dependencias (log4net, Castle.DynamicProxy, etc.)
- Asegurarse que las dependencias son las versiones exactas
- Revisar que no colisionen con otra previamente instalada (especialmente si había algún assembly)
- Copiar todo a .\lib en la fuente del proyecto
- Agregarlo al sistema de control de código fuente



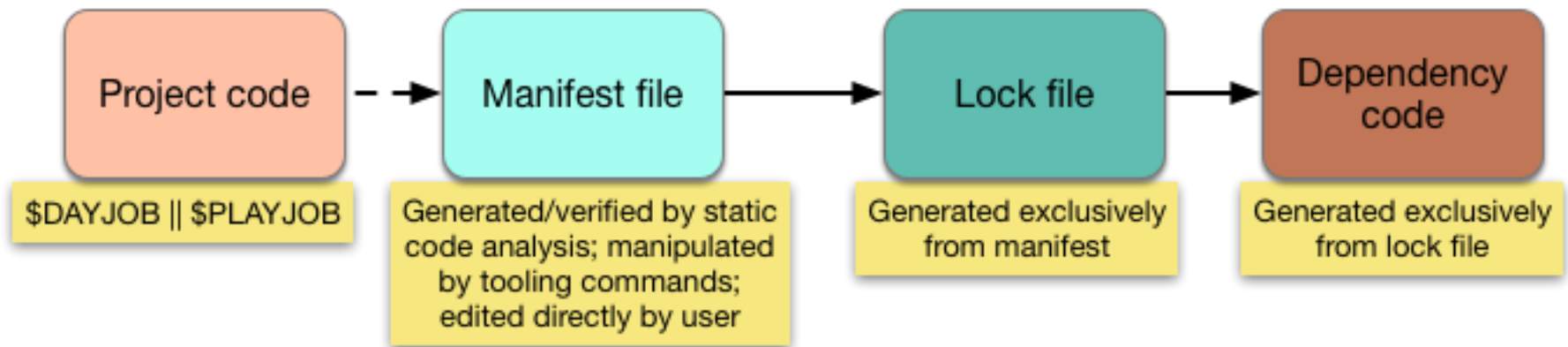
¿Qué es un paquete?

- Los paquetes como su nombre indica son paquetes de software que incluyen los ficheros, en algunos casos muchos, que componen una aplicación. Estos paquetes pueden incluir ficheros fuente o bien binarios, número de versión, así como las referencias a las librerías comunes que necesite.

Sistema de gestión de paquetes

- También conocido como **gestor de paquetes**, es una colección de herramientas que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software. El término se usa comúnmente para referirse a los gestores de paquetes en sistemas Unix-like, especialmente GNU/Linux, ya que se apoyan considerablemente en estos sistemas de gestión de paquetes.

Funcionamiento de Gestor de Paquetes



Simplicidad de uso

```
C:\>npm install math
```

NPM

- NPM es el gestor de paquetes de Node.js. Podemos instalar miles de módulos desde www.npmjs.com/.
- Se pueden instalar módulos de dos formas, globalmente o localmente.

NPM

- Los módulos globales están disponibles para cualquier proyecto en nuestro ordenador.
- Los módulos locales se instalan por proyecto, permitiéndonos tener diferentes versiones según nuestras necesidades. De hecho, la instalación por defecto se realiza de forma local.

package.json

- Todos los paquetes de NPM contienen un archivo, generalmente en la raíz del proyecto, llamado **package.json**
- Este archivo contiene toda la metadata pertinente al proyecto. Es el que le informa a NPM las dependencias del paquete, versión, contribuidores y demás datos necesarios para que funcione el proyecto.

```
{  
  "name" : "barebones",  
  "version" : "0.0.0",  
}
```


package.json

- Análisis interactivo
- <http://browsenpm.org/package.json#version>

Funcionalidad de package.json

- No solamente package.json es un manifiesto de nuestro proyecto, sino que también nos permitirá mantener las versiones del mismo, volverlo modular o incluso, reconstruirlo sin necesidad de las librerías contenidas en este.
- Es por esta característica que al momento de versionar o añadir a un repositorio, el proyecto será más ligero conteniendo solamente lo necesario.

Advertencias



Complejidad

- Al ser tan sencillo añadir nuevas dependencias a un proyecto, es habitual encontrar proyectos con decenas de dependencias externas, cada una de ellas con sus propias dependencias. Esto no es algo malo en si mismo, a fin de cuentas, ¿para qué reinventar la rueda?, pero no hay que olvidar que **cada dependencia que asumes en un proyecto es una dependencia que tienes que mantener.**

Complejidad

- No vale la pena añadir toda una librería por solamente una función de diez o veinte líneas.



- Cuando aparece un problema, no sirve de nada decir “el problema no está en mi código sino en la versión 1.9.10 de jQuery, yo no tengo la culpa”. De cara a tus usuarios y tus clientes, **lo que no funciona es tu aplicación**, esa por la que ellos han pagado. Al final tienes que responder no sólo por tu código sino por el de todas las dependencias que estás utilizando.

Bibliografía

- Official docs from NPM <https://docs.npmjs.com/>
- Sistema de gestión de paquetes
https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_paquetes
- Análisis interactivo de package.json
<http://browsenpm.org/package.json#version>

GRACIAS POR SU ATENCIÓN