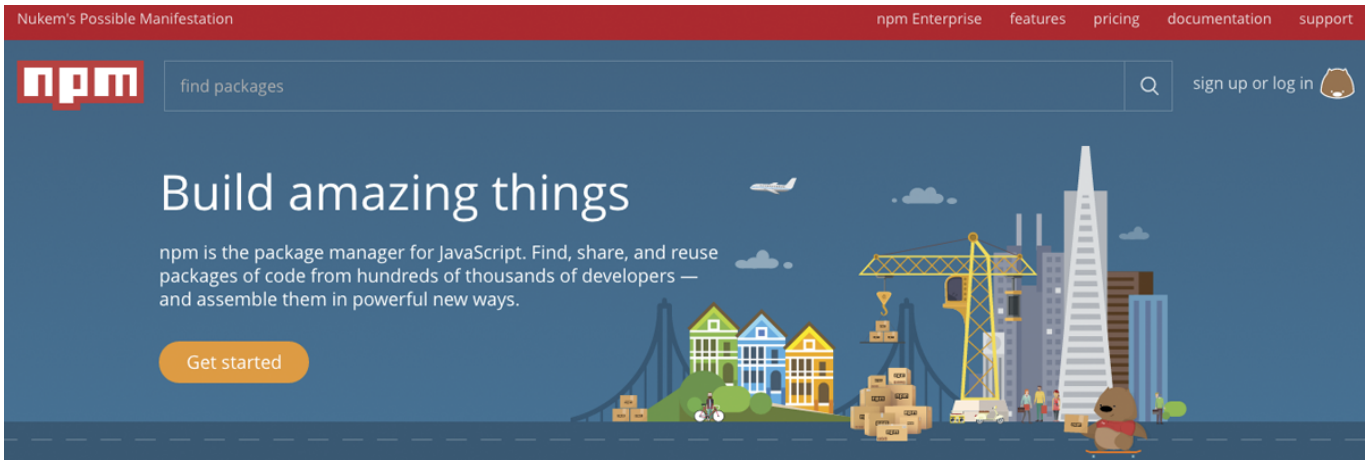
Lección 4: NPM como herramienta para automatización de tareas

Lectura: Usos comunes y buenas prácticas de NPM

Usos comunes y buenas prácticas de npm



Usos comunes:

Si necesitas llevar a cabo operaciones en tu paquete, antes de que sea usado y de manera que no dependa del sistema operativo o la arquitectura de la consola de destino, usa el script **prepublish.**

Éste incluye tareas como:

- Compilar el código de origen de CoffeeScript en JavaScript.
- Crear versiones minificadas del código de origen de JavaScrip
- Atraer recursos remotos que tu paquete usará.

La ventaja de hacer todo esto durante el **prepublish** es que es de inmediato y en un mismo lugar, de esta forma puedes reducir la complejidad y la variabilidad. Esto significa que:

- Puedes depender de **coffee-script** como un **devDependency.** De esta manera tus usuarios no necesitarán tenerlo instalado.
- No necesitas incluir minificadores en tu paquete. Esto reduce el tamaño para tus usuarios.
- No necesitas apoyarte en que tus usuarios tengan curl, wget u otras herramientas del sistema.

Buenas prácticas:

- No salgas con un código de error no nulo, a no ser que así lo quieras. Excepto por los scripts que no están instalados, esto hará que la acción de npm falle y retroceda. Si la falla es mínima o solo va a prevenir algunas características opcionales, es mejor solo imprimir una advertencia y la salida satisfactoriamente.
- Intenta no usar scripts para que hagan lo que npm puede hacer por ti. Lee sobre **package.json** para que veas todas las cosas que puedes especificar y habilitar solo describiendo apropiadamente tu paquete. En general, esto te guiará a un estado más robusto y consistente.
- Inspecciona el ambiente para determinar dónde poner las cosas. Por ejemplo, si el ambiente variable de **npm_config_binroot** está configurado como **/home/user/bin,** entonces no trates de instalar ejecutables en **/usr/local/bin,** ya que el usuario probablemente lo configuró así por una razón.
- No le pongas prefijos a tus comandos de scripts con “sudo”. Si los permisos de raíz son requeridos por alguna razón, entonces fallará con ese error y el usuario deberá ser sudo para ejecutar el comando npm.
- No uses **instalar,** usa el archivo **.gyp** para compilación y usa prepublish para todo lo demás. Casi nunca deberías configurar explícitamente una preinstalación o una instalación de script. Si estás haciendo esto, considera si tienes otra opción, ya que el único uso válido de la instalación o preinstalación de scripts es para la compilación, lo cual debe hacerse en la arquitectura