La aplicación de los doce factores

II Dependencias

Declarar y aislar explícitamente dependencias

La mayoría de los lenguajes de programación ofrecen un sistema de empaque para distribuir bibliotecas de soporte, como CPAN para Perl o Rubygems para Ruby. Las bibliotecas instaladas a través de un sistema de empaque pueden instalarse en todo el sistema (conocidas como "paquetes de sitio") o en el directorio que contiene la aplicación (conocido como "venta" o "agrupación").

Una aplicación de doce factores nunca se basa en la existencia implícita de paquetes en todo el sistema. Declara todas las dependencias, completa y exactamente, a través de un manifiesto de *declaración de dependencia* . Además, utiliza una herramienta de *aislamiento de dependencias* durante la ejecución para garantizar que no se filtren dependencias implícitas del sistema circundante. La especificación de dependencia completa y explícita se aplica de manera uniforme tanto a la producción como al desarrollo.

Por ejemplo, Bundler para Ruby ofrece el `Gemfile`formato de manifiesto para la declaración de dependencia y `bundle exec`para el aislamiento de dependencia. En Python hay dos herramientas separadas para estos pasos: Pip se usa para la declaración y Virtualenv para el aislamiento. Incluso C tiene Autoconf para declaración de dependencia, y el enlace estático puede proporcionar aislamiento de dependencia. No importa cuál sea la cadena de herramientas, la declaración de dependencia y el aislamiento siempre deben usarse juntos: solo uno u otro no es suficiente para satisfacer los doce factores.

Una ventaja de la declaración explícita de dependencia es que simplifica la configuración para los desarrolladores nuevos en la aplicación. El nuevo desarrollador puede verificar la base de código de la aplicación en su máquina de desarrollo, requiriendo solo el tiempo de ejecución del lenguaje y el administrador de dependencias instalados como requisitos previos. Podrán configurar todo lo necesario para ejecutar el código de la aplicación con un *comando de construcción* determinista . Por ejemplo, el comando de compilación para Ruby / Bundler es `bundle install`, mientras que para Clojure / Leiningen lo es `lein deps`.

Las aplicaciones de doce factores tampoco dependen de la existencia implícita de ninguna herramienta del sistema. Los ejemplos incluyen el desplazamiento a ImageMagick o `curl`. Si bien estas herramientas pueden existir en muchos o incluso en la mayoría de los sistemas, no hay garantía de que existan en todos los sistemas donde la aplicación pueda ejecutarse en el futuro, o si la versión encontrada en un sistema futuro será compatible con la aplicación. Si la aplicación necesita convertirse en una herramienta del sistema, esa herramienta debe enviarse a la aplicación.

Turco (tr) | 日本語 (ja) | Français (fr) | Italiano (it) | Slovensky (sk) | Ελληνικά (el) | Español (es) | Polski (pl) | Русский (ru) | Portugués brasileño (pt_br) | Inglés (en) | 한국어 (ko) | 简体 中文 (zh_cn) | Українська (reino unido) | Deutsch (de) | ภาษา ไทย (th)
«Anterior
Próximo "
Turco (tr) | 日本語 (ja) | Français (fr) | Italiano (it) | Slovensky (sk) | Ελληνικά (el) | Español (es) | Polski (pl) | Русский (ru) | Portugués brasileño (pt_br) | Inglés (en) | 한국어 (ko) | 简体 中文 (zh_cn) | Українська (reino unido) | Deutsch (de) | ภาษา ไทย (th)
Escrito por Adam Wiggins
Última actualización 2017
Código fuente
Descargar ePub Book
Política de privacidad