Clonar un subdirectorio de un repositorio en Git

<https://www.delftstack.com/howto/git/git-clone-subdirectory/>

Editado por Gustavo García, 20 de septiembre de 2022.

Este tutorial le enseñará cómo clonar un subdirectorio de un repositorio de Git.

Git, un sistema de control de versiones, mantiene el historial de los cambios realizados en el directorio de un proyecto. En un entorno típico de desarrollo de proyectos, se segregarían diferentes módulos de proyectos en diferentes subdirectorios. Entonces, se puede clonar solo el subdirectorio específico de un módulo del proyecto. Podemos usar la función the sparse-checkout  proporcionada por Git para tales fines.

Ahora ilustraremos esto con un ejemplo.

Git se usa en un entorno de desarrollo colaborativo para realizar un seguimiento de los cambios realizados en los archivos en el directorio del proyecto. Uno mantendría diferentes módulos de proyecto en subdirectorios separados dentro del directorio principal del proyecto en el entorno de desarrollo del proyecto.

El repositorio de Git rastrearía este directorio principal del proyecto y, por lo tanto, uno puede clonar o verificar todo el directorio del proyecto. Por lo general, diferentes equipos trabajarían en diferentes módulos de proyecto en un proyecto grande.

Por lo tanto, en tales casos, no sería necesario consultar todo el directorio del proyecto. Bastaría con clonar solo el subdirectorio del módulo del proyecto en el que está trabajando un equipo en particular.

La clonación solo de subdirectorios específicos (es decir, un subconjunto del repositorio Git del proyecto, se denomina sparse checkout). Podemos clonar solo los subdirectorios que queremos del directorio del proyecto en el repositorio de Git en un sparse checkout.

Supongamos que tenemos un directorio de proyectos llamado my\_project. En el directorio del proyecto my\_project, hay subdirectorios para diferentes módulos llamados frontend, backend, documentation, etc.

Solo queremos pagar o clonar el subdirectorio del módulo frontend. Por lo tanto, ahora primero crearemos un directorio para el repositorio, de la siguiente manera:

$ mkdir my\_project

$ cd my\_project

Después de crear el directorio del proyecto, ahora inicializaremos el repositorio de Git y agregaremos la URL remota, de la siguiente manera:

$ git init

$ git remote add -f origin https://github.com/johndoe/my\_project.git

Debemos habilitar sparse checkouts  para chequear o clonar solo subdirectorios del directorio principal del proyecto.

Podemos usar el comando git config  para lograr lo mismo. Necesitamos usar el comando git config  de la siguiente manera:

$ git config core.sparsecheckout true

Ahora podemos decirle a Git qué subdirectorios queremos revisar.

Por lo tanto, para habilitar solo el checkout del subdirectorio frontend , debemos incluirlo en el archivo .git/info/sparse-checkout:

$ echo "frontend/" >> .git/info/sparse-checkout

Ahora, podemos obtener los archivos del repositorio remoto de Git:

$ git pull origin main

Ahora, tenemos en el árbol de trabajo solo el subdirectorio frontend  del proyecto principal, my\_project.