

Gestiona Varios Borradores Fácilmente En Jekyll – Guía

A pesar de ser sólo un sitio web con paginas estáticas, podemos utilizar un plugin en **Jekyll** para hacer esta tarea de forma mas fácil. Este es tan solo la continuación del anterior articulo, donde hablamos sobre [realizar un borrador en jekyll](#), como también, conocer la carpeta donde se almacenan los borradores en [jekyll](#).



Suena fácil ¿no? no debería ser un problema si usted sólo administra 1 o 2 borradores. Pero, cuando usted tiene 5 o mas borradores para manejar, el cambio de nombre de cada archivo, y especificar la fecha correcta a pulso puede volverse un dolor de cabeza. Vamos a echar un vistazo de cómo podemos simplificar el flujo de trabajo con un plugin de **Jekyll**.

Primeros Pasos

Antes de comenzar, vamos a crear una nueva carpeta llamada **_plugins**; Se requiere esta carpeta para almacenar y ejecutar los **plugins de Jekyll**. También tenemos que crear un nuevo archivo llamado **publisher.rb** en ella; técnicamente, puede nombrar libremente ese archivo con el nombre que desee.

Nosotros vamos a usar un **plugin en Jekyll** creado por [Jeffrey Sambells](#). Este plugin se encargará los cambios cuando se publica un **post** en Jekyll. Se cambiará el nombre del archivo correctamente, junto con la fecha.

El siguiente, es el código fuente del plugin, disponible en la [página Gist](#). Copia este código y pega en el archivo **publisher.rb** creado con anterioridad. Si te molesta ir página tras página, el código sería el siguiente:

```
[c]
```

```
module Jekyll
  class PostPublisher < Generator
    safe false

    def replace(filepath, regexp, *args, &block)
      content = File.read(filepath).gsub(regexp, *args, &block)
      File.open(filepath, 'wb') { |file| file.write(content) }
    end

    def generate(site)
      @files = Dir["_publish/*"]
      @files.each_with_index { |f,i|
        now = DateTime.now.strftime("%Y-%m-%d %H:%M:%S")
        replace(f, /^date: unpublished/mi) { |match| "date: \"" + now + "\""}
        now = Date.today.strftime("%Y-%m-%d")
        File.rename(f, "_posts/#{now}-#{File.basename(f)}")
      }
    end
  end
end
```

```
[/c]
```

Si su servidor Jekyll se está ejecutando, reinícielo para que el plugin funcione.

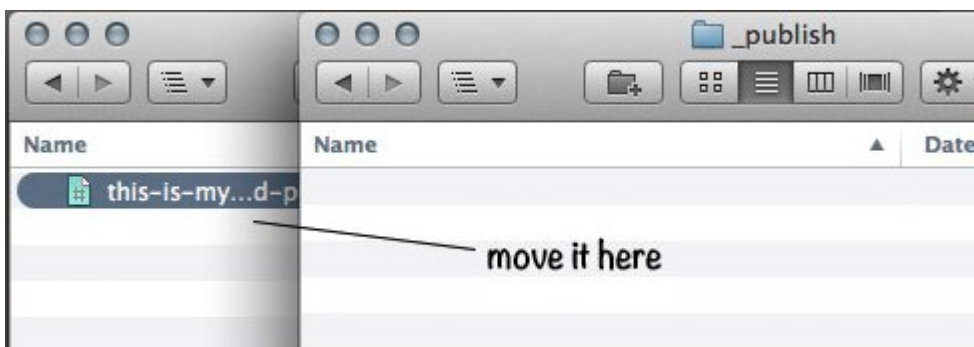
Usando El Plugin En Jekyll

Para utilizar el **plugin**, tenemos que crear una nueva carpeta llamada **_publish**. Vamos a mover nuestros borradores posteriores a esta carpeta, una vez que estamos dispuestos a publicarlo. Sin embargo, antes de hacerlo, establezca el **date** de los borradores en **unpublished**, justo de la siguiente manera:

```
[html]
-
layout: post
title: "This is My Second Post"
date: unpublished
-

[/html]
```

Ahora, mueva el proyecto a la carpeta **_publish**.



Como se mencionó, **Jekyll** moverá automáticamente a la carpeta **_posts** los artículos dependiendo de la fecha que se haya establecido, y luego publicara el artículo por usted.

Finalizando

Hemos visto que **Jekyll** amplía sus funcionalidades con los **plugins**, en este caso, un plugin para borradores. Puedes encontrar más plugins para **Jekyll** en la siguiente página: [Plugins Disponibles](#).

En el próximo artículo, le mostraremos cómo publicar su blog en Jekyll para un [servidor web](#) a través de **FTP**. Estén atentos!