EBook Gratis

APRENDIZAJE github

Free unaffiliated eBook created from
Stack Overflow contributors.

#github

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: github

It is an unofficial and free github ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official github.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con github

Observaciones

Esta sección proporciona una descripción general de qué es github, y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de github, y vincular a los temas relacionados. Dado que la Documentación para github es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Instalación o configuración

GitHub es una gran colección de repositorios Git. En otras palabras, ¡puedes pensar en GitHub como una colección de muchos proyectos!

Creando una cuenta

- Visita la página principal de GitHub aquí.
- ¡Elija un nombre de usuario, ingrese su dirección de correo electrónico y elija una contraseña segura y listo!

Herramientas utiles

Para los principiantes de Git / GitHub, entender cómo funciona el control de versiones puede ser confuso al principio. Existe una versión GUI de GitHub que puedes descargar y usar. GitHub Desktop es solo esa herramienta.

Creando tu primer repositorio

Puedes pensar en un repositorio como un proyecto. Puede crear un repositorio en línea o fuera de línea. Siga los pasos a continuación:

En línea

1. Primero inicia sesión y ve a tu perfil.
2. Vaya a la pestaña "Repositorios" cerca de la parte superior de la página
3. ¡Presiona el botón verde "Nuevo" y estás listo para retumbar!

Desconectado

1. Descarga e instala git (elige el sistema operativo que estás ejecutando)
2. Después de la descarga y la instalación, puede utilizar la herramienta de línea de comandos o puede descargar un cliente GUI.
3. Después de la instalación, crea una cuenta en github
4. Desde la parte superior derecha, haga clic en el signo + y elija crear un nuevo repositorio o importar un archivo existente en.
5. Si elige uno nuevo, ingrese el nombre del repositorio y elija tenerlo público o privado.
6. Haga clic en: Crear repositorio

NB Los repositorios privados no están disponibles para usuarios gratuitos.

Archivo readme

Si su proyecto no tiene README.md, GitHub puede analizar README.rdoc para mostrar los detalles. Si tiene ambos, usará README.md, ignorando silenciosamente rdoc.

Un archivo README puede incluir-

Título del Proyecto

Describe brevemente sobre tu proyecto. También puede proporcionar el enlace al sitio web del proyecto, las insignias, la comunidad y la información de contacto (es decir, correo electrónico, sitio social).

Descargar

Enlace de archivo ejecutable (ejecutable o minimizado o archivo de instalación). También puede haber enlaces a versiones anteriores.

Instalación

Cómo se puede utilizar tu trabajo. Puede incluir los requisitos previos, la configuración, las bibliotecas de terceros, el uso, las precauciones, etc.

Demostración

Puede incluir ejemplo de código, archivo gif, enlace de video o incluso capturas de pantalla.

Autores

Nombres de autores, información de contacto, etc.

Expresiones de gratitud

Lista de personas o comunidad ayudada e inspirada a lo largo del proyecto.

Contribuyendo

Instrucciones para contribuir (es decir, agregar características, informar errores, enviar parches) al proyecto. Puede incluir enlace de documentación también.

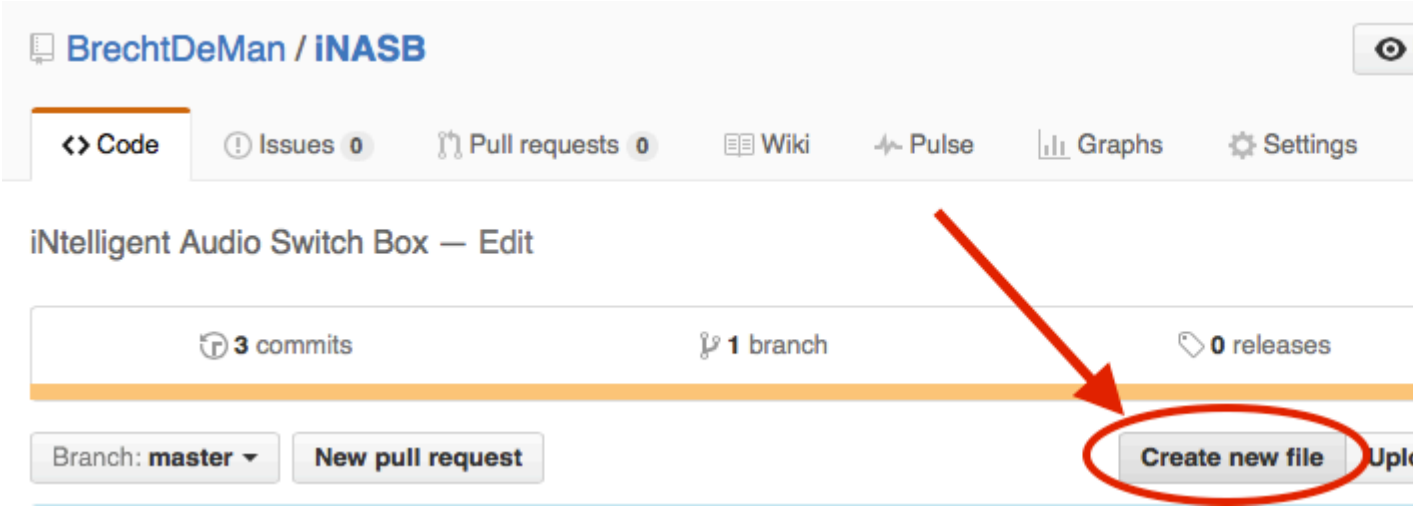
Licencia

Dar una breve introducción sobre su licencia. También puede dar un enlace al sitio de la licencia.

Archivo de licencia

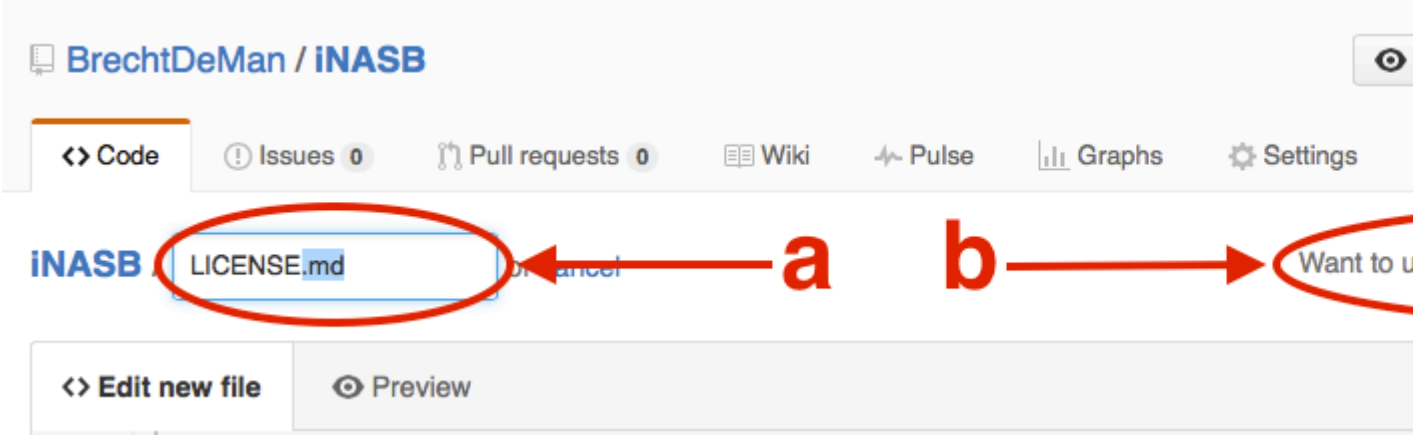
GitHub lo ayuda a agregar rápidamente una licencia a su repositorio, como una alternativa para agregar su propio archivo de texto / markdown.

1. En su repositorio, haga clic en 'Crear nuevo archivo'

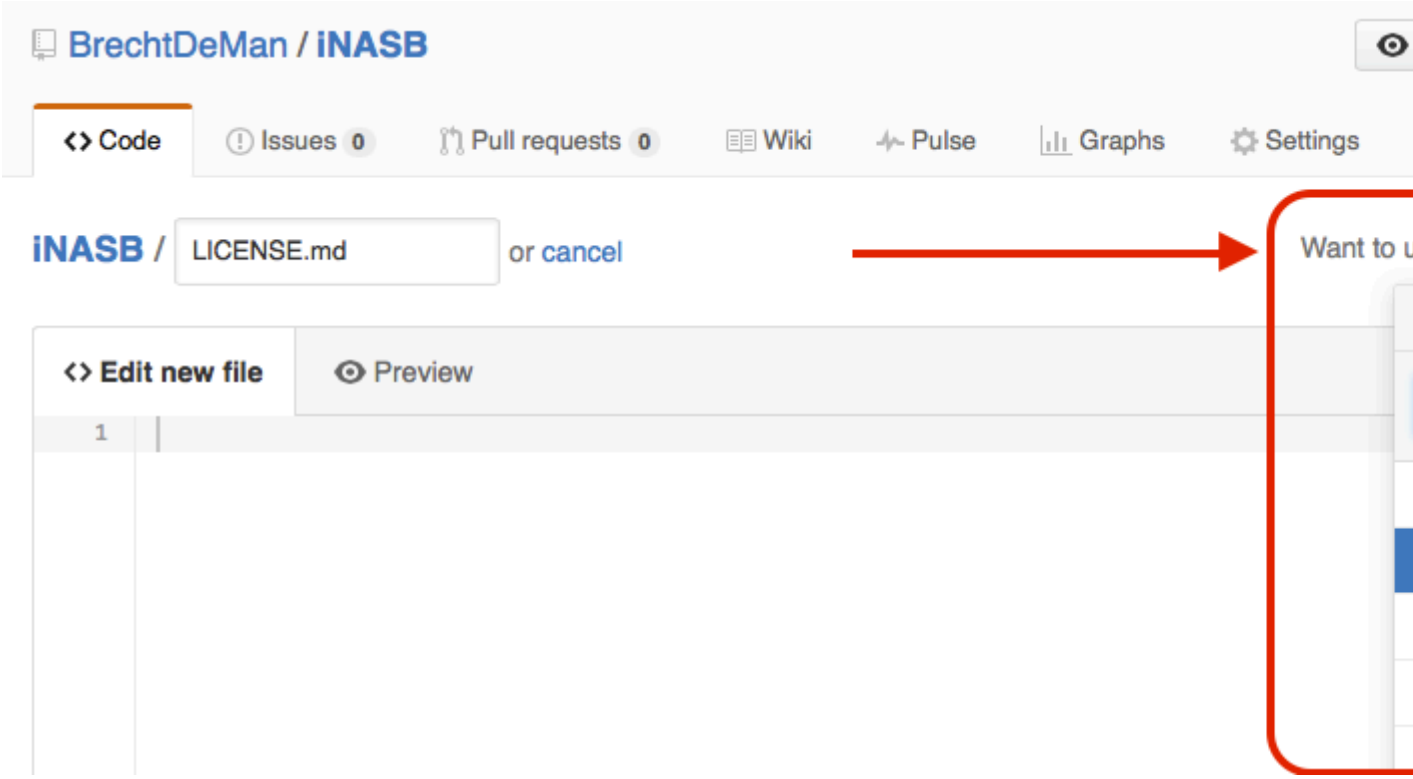


2. En la próxima página:

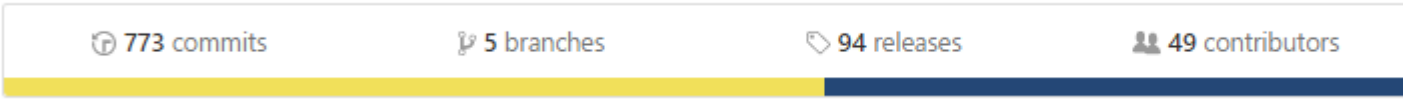
    1. Escriba `LICENSE.md` o `LICENSE.txt` como el nombre de archivo del nuevo archivo.
    2. *¿Quieres usar una nueva plantilla?* Aparecerá el diálogo.



3. Elija su licencia preferida.



4. La licencia que podías ver en los detalles del repositorio:



De preguntas y respuestas: cómo agregar una licencia a un proyecto existente de Github

GitHub Con sabor Markdown

GitHub expande la sintaxis de Markdown para proporcionar nuevas funciones útiles.

Encabezamiento

```
# Header1
## Header2
### Header3
#### Header4
##### Header5
###### Header6
H1
===
H2
---
```

Header1

Header2

Header3

Header4

Header5

Header6

H1

H2

Énfasis

```
*Italic1* _Italic2_
**Bold1** __Bold2__
***Bold_Italic***
~~Strikethrough~~
```

Italic1 Italic2
Bold1 Bold2
Bold_Italic
~~Strikethrough~~

Linea horizontal

```
---
***
___
```

Lista

```
unordered list:

* item-1
  * sub-item-1
  * sub-item-2
- item-2
  - sub-item-3
  - sub-item-4
+ item-3
  + sub-item-5
  + sub-item-6


ordered list:

1. item-1
  1. sub-item-1
  2. sub-item-2
2. item-2
  1. sub-item-3
  2. sub-item-4
3. item-3
```

unordered list:

- item-1
  - sub-item-1
  - sub-item-2
- item-2
  - sub-item-3
  - sub-item-4
- item-3
  - sub-item-5
  - sub-item-6

ordered list:

1. item-1
   i. sub-item-1
   ii. sub-item-2
2. item-2
   i. sub-item-3
   ii. sub-item-4
3. item-3

Mesa

```
Table Header-1 | Table Header-2 | Table Header-3
:--- | :---: | ---:
Table Data-1 | Table Data-2 | Table Data-3
TD-4 | Td-5 | TD-6
Table Data-7 | Table Data-8 | Table Data-9
```

Table Header-1	Table Header-2	Table Header-3
Table Data-1	Table Data-2	Table Data-3
TD-4	Td-5	TD-6
Table Data-7	Table Data-8	Table Data-9

Código

````
inline code- `int i=0`

block code-
``` C
for(int i=0; i<10; i++){
    printf("Hallow World! \n");
}
```
````

inline code- `int i=0`

block code-

```
for(int i=0; i<10; i++){
    printf("Hallow World! \n");
}
```

Citar

```
> Stay hungry; stay foolish.
>> Quality is better than quantity.
>>> Life is not fair; get used to it.
```

> Stay hungry; stay foolish.
>
> > Quality is better than quantity.
> >
> > > Life is not fair; get used to it.

Enlazar

```
https://github.com
[GitHub](https://github.com)
[GitHub](https://github.com "github website")
[GitHub][1]

[1]: https://github.com
```

https://github.com
GitHub
GitHub
GitHub

Imagen

```
![GitHub Logo](https://assets-cdn.github.com/images/icons/emoji/octocat.png "GitHub")
```

Listas de tareas

```
- [x] completed item
- [ ] incomplete item
```

- [x] completed item
- [ ] incomplete item

Emoji

```
:octocat: :+1: :book: :ghost: :bulb: :imp:
```

🐙👍📖👻💡😈

Para todos los emojies de GitHub visita- Hoja de trucos Emoji .

Referencias SHA

Cualquier referencia a un hash SHA1 de una confirmación se convertirá en un enlace a la confirmación en GitHub:

```
e7909ea4fbb162db3f7f543d43c30684a3fb745f
```

Write Preview

`e7909ea`

Solicitud de extracción y referencias de problemas

Cualquier referencia a una solicitud de extracción o un problema se vinculará automáticamente a esa solicitud de extracción o problema.

Esto se puede hacer colocando un `#` delante del número de solicitud de emisión / problema.

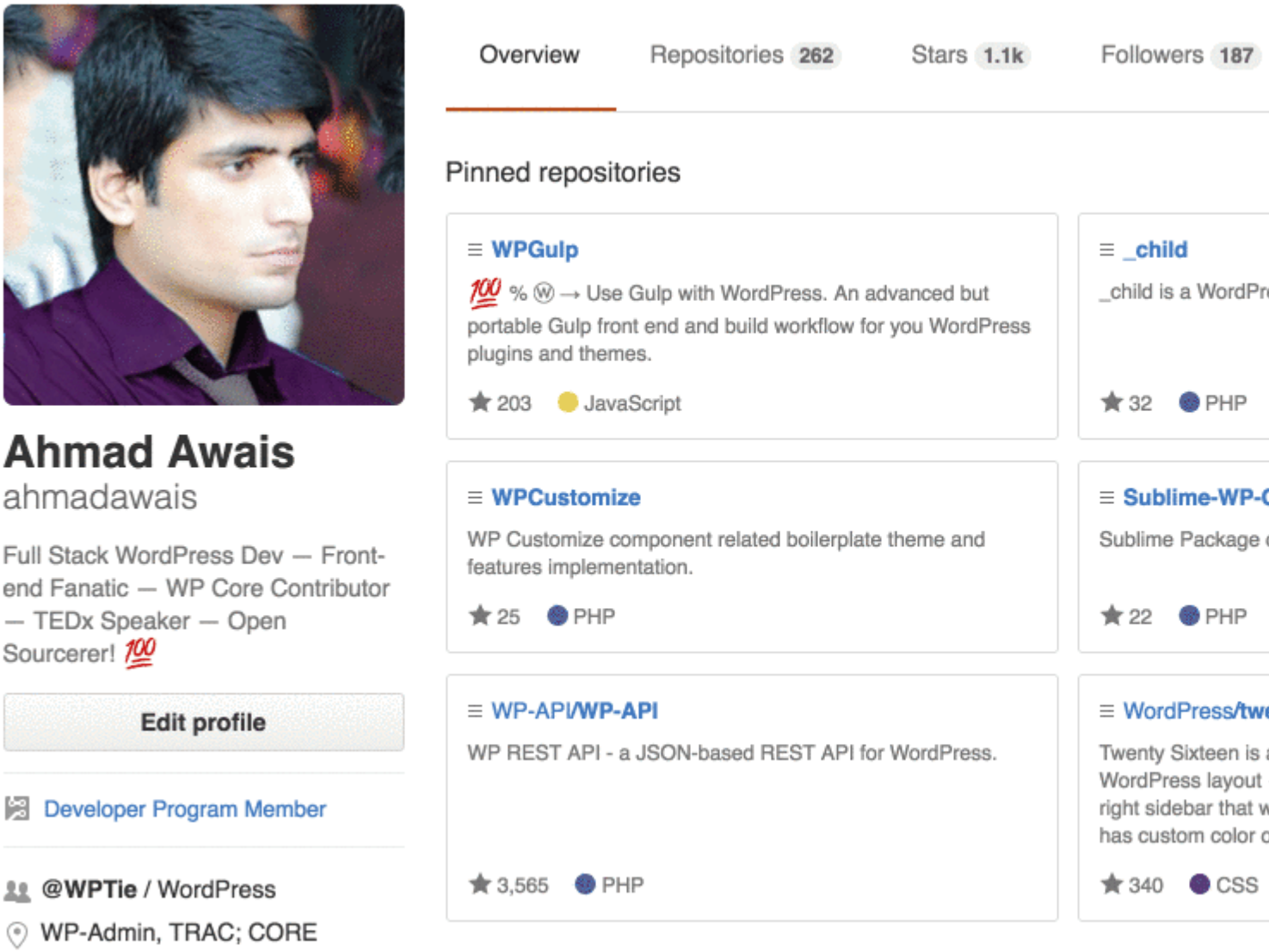
Lea Empezando con github en línea: https://riptutorial.com/es/github/topic/1214/empezando-con-github

Capítulo 2: ¿Cómo crear etiquetas de GitHub personalizadas?

Examples

¡Crea etiquetas personalizadas de GitHub!

Aquí hay un GIF rápido para que el proceso sea lo más fácil posible.



Las etiquetas se pueden aplicar a los problemas y las solicitudes de extracción para indicar la prioridad, la categoría o cualquier otra información que considere útil.

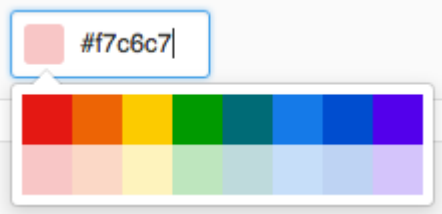
En GitHub, navegue a la página principal del repositorio.

1. Bajo el nombre de su repositorio, haga clic en Problemas o Solicitudes de extracción.
2. Botón de etiquetas de problemasSiguiente al campo de búsqueda, haga clic en Etiquetas.

3. Haga clic en Nueva etiqueta para crear una nueva etiqueta, o haga clic en Editar para editar una existente.

4. En el cuadro de texto, escriba el nombre de su nueva etiqueta.

5. Seleccione un color para la etiqueta de la barra de colores. Puede personalizar este color editando el número hexadecimal encima de la barra de colores.



6. Haga clic en Crear etiqueta para guardar la nueva etiqueta.

Espero que ayude. Upvote si lo hace.

Lea ¿Cómo crear etiquetas de GitHub personalizadas? en línea:
https://riptutorial.com/es/github/topic/7159/-como-crear-etiquetas-de-github-personalizadas-

Capítulo 3: Actualizar un repositorio bifurcado

Observaciones

- Ayuda de GitHub: configurando un control remoto para un fork
- Ayuda de GitHub: sincronizando un tenedor
- Ans popular en StackOverFlow

Examples

Configura un control remoto para tu fork y luego sincroniza tu fork (rama principal)

1. Configura un mando a distancia para mi horquilla.

```
$ cd my_local_repo

$ git remote add upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git
    # Specify a new remote upstream repository that will be synced with the fork

$ git remote -v
    # Verify the new upstream repository specified for my fork
```

2. Sincronizar mi horquilla localmente

```
$ cd my_local_repo

$ git fetch upstream
    # Fetch the branches and their respective commits from the upstream repository
    # Commits to master will be stored in a local branch, upstream/master

$ git checkout master

$ git merge upstream/master
    # Merge the changes from upstream/master into your local master branch
    # brings your fork's master branch into sync with the upstream repo
```

3. Sincronizar mi tenedor en github

```
$ git push origin master
```

Lea Actualizar un repositorio bifurcado en línea:
https://riptutorial.com/es/github/topic/3758/actualizar-un-repositorio-bifurcado

Capítulo 4: Clonando un repositorio desde GitHub

Sintaxis

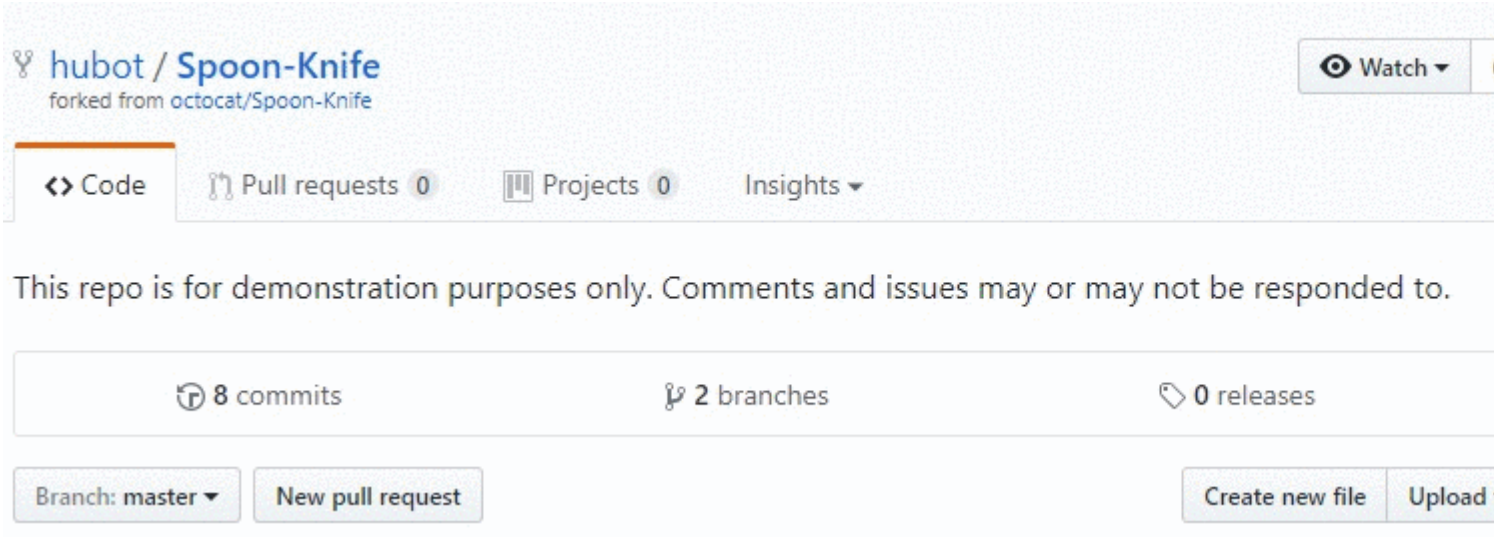
- git clone github.com/username/repository

Examples

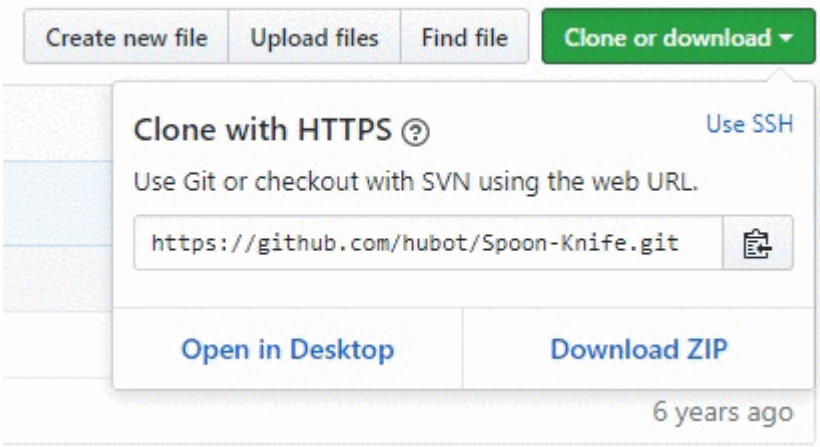
Clonar un repositorio

1. Vaya al repositorio que desea clonar (algo como: https://github.com/ *username* / *repo*)

2. A la derecha, haga clic en el botón verde llamado *clonar o descargar*



3. Aparecerá una pequeña ventana, copie la url (algo como: https://github.com/ *username* / *repo* .git)



4. Abra una ventana de terminal en la máquina en la que desea clonar ese proyecto
5. Navegue desde la línea de comando hasta la ubicación donde desea clonar el proyecto
6. Ingrese el comando: git clone <copied_url_from_step_3>
7. Presiona enter
8. Aparecerá algo parecido a lo siguiente:

   `<repo_name>` en `<repo_name>` ...

   remoto: contando objetos: 10, hecho.

   remoto: Compresión de objetos: 100% (8/8), hecho.

   eliminar: Total 10 (delta 1), reutilizado 10 (delta 1)

   Desembalaje de objetos: 100% (10/10), hecho.

Lea Clonando un repositorio desde GitHub en línea:
https://riptutorial.com/es/github/topic/3761/clonando-un-repositorio-desde-github

Capítulo 5: Copia de seguridad de GitHub

Examples

Clonando todos los repositorios para un nombre de usuario

Ejecute el siguiente comando, reemplazando el nombre de usuario con el nombre de usuario, para clonar todos los repositorios de GitHub para ese usuario en el directorio actual.

```
curl "https://api.github.com/users/username/repos?page=1&per_page=100" | grep -e 'git_url*' |
cut -d \" -f 4 | xargs -L1 git clone
```

Esto solo clonará los primeros 100 repositorios.

Lea Copia de seguridad de GitHub en línea: https://riptutorial.com/es/github/topic/3760/copia-de-seguridad-de-github

Capítulo 6: Cuestiones

Examples

Creando un problema

1. Vaya a la página de GitHub para el proyecto donde desea crear un problema.
2. Haga clic en ***Problemas*** .
3. En la parte superior derecha, haga clic en ***Nuevo número*** .
4. Ingrese el título del asunto.
5. Ingrese el cuerpo del problema (incluidos los registros, fragmentos de código, etc.)
6. *Opcional:* para ver el problema antes de enviarlo, haga clic en vista previa.
7. Haga clic en ***Enviar nuevo número*** .

Lea Cuestiones en línea: https://riptutorial.com/es/github/topic/3757/cuestiones

Capítulo 7: descargar un solo archivo desde el repositorio GitHub

Examples

desde un repositorio público usando la línea de comandos y renombrando archivos

Este ejemplo toma el archivo Node.gitignore del repositorio gitignore de GitHub, lo descarga a su directorio de trabajo actual y lo cambia a .gitignore, todas acciones muy típicas para alguien que está iniciando un nuevo proyecto node.js.

```
$ curl http://github.com/github/gitignore/raw/master/Node.gitignore -o .gitignore
```

Encuentra la url del archivo que quieres descargar.

1. navegue hasta el archivo deseado en un repositorio
2. haga clic en el botón 'raw'
3. copia la url desde la barra de direcciones

Vea el siguiente ejemplo del repositorio gitignore de GitHub:
http://github.com/github/gitignore/raw/master/Node.gitignore

Puede reconocer rápidamente una URL que funcionará para descargar un archivo individual frente a la descarga de la página html. Busque el subdirectorio / raw / justo antes del nombre de la rama.

Lea descargar un solo archivo desde el repositorio GitHub en línea:
https://riptutorial.com/es/github/topic/10898/descargar-un-solo-archivo-desde-el-repositorio-github

Capítulo 8: Eliminar datos confidenciales o archivos grandes

Introducción

Si confirma datos confidenciales, como una contraseña o clave SSH en un repositorio Git, puede eliminarlos del historial. Para eliminar por completo los archivos no deseados del historial de un repositorio, puede usar el comando git filter-branch o BFG Repo-Cleaner.

Observaciones

1. Dígales a sus colaboradores que modifiquen, no fusionen, las ramas que crearon a partir de su antiguo historial de depósito (contaminado). Un compromiso de fusión podría reintroducir parte o toda la historia contaminada que acabas de tomarte la molestia de purgar.

2. Después de que pase un tiempo y confíe en que git filter-branch no tuvo efectos secundarios no deseados, puede forzar la eliminación de referencias de todos los objetos en su repositorio local y la recolección de basura con los siguientes comandos (utilizando Git 1.8.5 o más reciente):

   git for-each-ref --format = 'delete% (refname)' refs / original | git update-ref --stdin

   git reflog expire --expire = now --all

   git gc --prune = ahora

Examples

Utilizando filtro-rama

```
git filter-branch --force --index-filter \
'git rm --cached --ignore-unmatch PATH-TO-YOUR-FILE-WITH-SENSITIVE-DATA' \
--prune-empty --tag-name-filter cat -- --all
```

Agregue su archivo con datos confidenciales a .gitignore para asegurarse de que no vuelva a cometerlo accidentalmente.

```
echo "YOUR-FILE-WITH-SENSITIVE-DATA" >> .gitignore
git add .gitignore
git commit -m "Add YOUR-FILE-WITH-SENSITIVE-DATA to .gitignore"
```

Empuje su repositorio local a GitHub

```
git push origin --force --all
```

Para eliminar el archivo confidencial de sus publicaciones etiquetadas, también deberá forzar el empuje contra sus etiquetas Git:

```
git push origin --force --tags
```

Usando el BFG Repo Cleaner

BFG Repo cleaner es una alternativa a git filter-branch. Se puede usar para eliminar datos confidenciales o archivos grandes que se cometieron incorrectamente, como binarios compilados desde la fuente. Está escrito en scala.

Sitio web del proyecto: BFG Repo Cleaner

Requerimientos

El entorno de ejecución de Java (Java 7 o superior - BFG v1.12.3 fue la última versión compatible con Java 6). La biblioteca de Scala y todas las demás dependencias se pliegan en el contenedor descargable.

Eliminar archivos con datos sensibles

```
bfg --delete-files YOUR-FILE-WITH-SENSITIVE-DATA
```

Lea Eliminar datos confidenciales o archivos grandes en línea:
https://riptutorial.com/es/github/topic/8170/eliminar-datos-confidenciales-o-archivos-grandes

Capítulo 9: GitHub Desktop

Introducción

¿Cómo instalar y trabajar con GitHub Desktop?

GitHub Desktop es, como su nombre lo indica, un entorno de escritorio para Windows y MacOS que incluye las características principales de Git, como la clonación, el empuje, la extracción (sincronización en GitHub Desktop), la fusión ...

El principal objetivo de los clientes de escritorio es ofrecer una forma más fácil de trabajar con git (y GitHub). En segundo plano, utiliza los mismos comandos que la mayoría de los usuarios utilizarían desde la línea de comandos.

Examples

Instalación y configuración

La instalación es bastante simple ya que hay instaladores separados para máquinas MacOS y Windows disponibles aquí . Actualmente hay dos versiones para descargar: una beta y una estable.

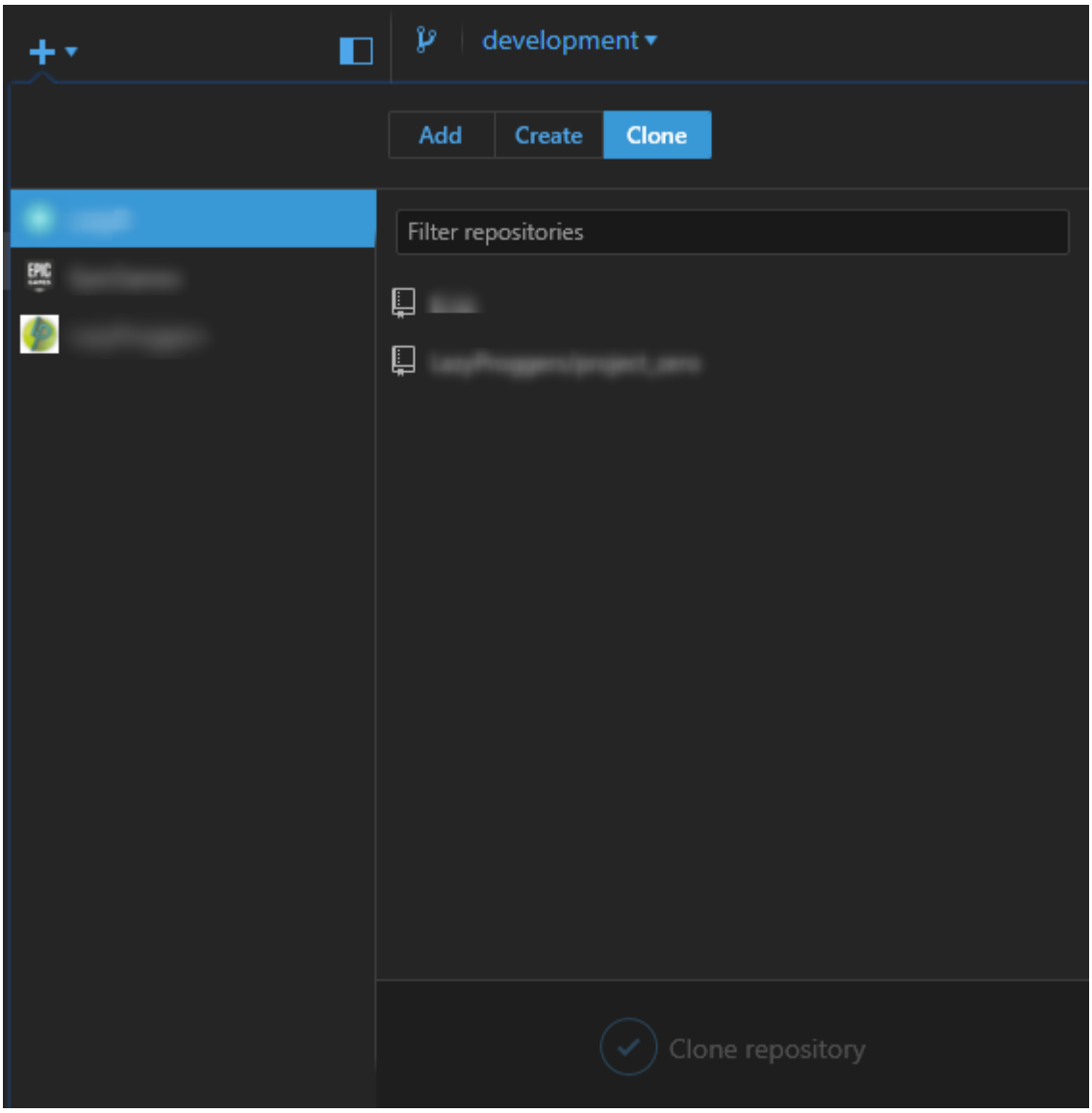
La instalación comenzará una vez que descargue el programa y deberá iniciar sesión con sus credenciales de GitHub. Ese es realmente el único paso porque después de eso puedes comenzar a crear un repositorio o clonar uno.

> Nota: durante la instalación, no solo se instalará GitHub Desktop, sino también Git. Así que no necesitas instalarlo por separado.

Clonando un repositorio

Como sucede con GitHub Desktop, la mayoría del trabajo es bastante simple: seleccionas "Clonar un repositorio" (en la versión estable, el signo más en la parte superior izquierda) y hay algunos repositorios (los tuyos y los repos de cada compañía en la que estás).) recomendado. Alternativamente, puede pegar un enlace a cualquier otro repositorio que quiera clonar.

> Nota: en la versión más reciente (beta) no hay recomendaciones (¿no es jet?).



Derivación

Puede seleccionar una rama en la parte superior izquierda. Cuando seleccionó la rama derecha, necesita presionar el botón de sincronización (arriba a la derecha) que ahora hace lo mismo que `git checkout BRANCHNAME` .

> En la versión anterior puedes ver 2 ramas diferentes a la vez y comparar los empujes. Además, puede ver una línea de tiempo de su proyecto (ver más abajo)

Creando una nueva rama

Puede crear una nueva rama haciendo clic en el símbolo de la rama (antiguo cliente) o en `File --> New Branch` .

> Tenga en cuenta que puede seleccionar de qué rama utiliza la nueva rama como base

haciendo clic en el nombre de la rama.

Empuje y tire (o: el botón de sincronización)

Pull (Sync)

Al igual que en la línea de comandos, debe extraer el estado actual del repositorio de vez en cuando. En GitHub Desktop, este proceso es llamado por el botón de `sync` en la esquina superior derecha.

empujar

Cuando realiza cambios locales y desea forzarlos, realiza una confirmación escribiendo algo en el cuadro de texto de resumen. A continuación, presiona `Commit to YOURCURRENTBRANCH` . Ahora deberás presionar el botón de sincronización y tu pulsación está hecha.

> Nota: Puede usar emoticonos, menciones y referencias a otros compromisos o problemas directamente desde el cuadro de texto.

Por lo tanto, el botón de **sincronización** se puede utilizar para `Push` , `Pull` o `Checkout` .

Lea GitHub Desktop en línea: https://riptutorial.com/es/github/topic/10023/github-desktop

Capítulo 10: Páginas de GitHub

Examples

Uso del generador automático de páginas para un repositorio.

1. Ir al sitio web de GitHub
2. Abre tu repositorio
3. Haga clic en Configuración
4. En las páginas de GitHub, haga clic en "Iniciar generador automático de páginas"
5. Sigue las instrucciones

Usando Git para crear páginas desde cero

1. Crea un nuevo repositorio o clona uno existente.
2. Crea una nueva rama llamada `gh-pages` sin ningún historial

```
$ git checkout --orphan gh-pages

# ensure you are in the correct directory then,
# remove all files from the old working tree
$ git rm -rf
```

3. Agregue un archivo index.html a la raíz del repositorio.

```
$ echo "Hello World" > index.html
$ git add index.html
$ git commit -a -m "First pages commit"
```

4. Empuje a Github.

```
$ git push origin gh-pages
```

Ahora puede cargar su nuevo sitio de Github Pages en
`http(s)://<username>.github.io/<projectname>`

Creando una URL personalizada para tu página de GitHub

Necesitará un nombre de dominio de un registrador .

En la rama de `gh-pages` de su repositorio de proyecto, o la rama principal de su repositorio de `username.github.io` , cree un archivo CNAME con el contenido `www.yourdomain.com` - el dominio *canónico* .

En la página de configuración del dominio de su registrador, dirija su dominio a su sitio web de GitHub. Configure dos registros CNAME (uno para el ápice raíz (@) y otro para www). Ambos deben apuntar a `username.github.io` o `username.github.io/repository` . Si su proveedor de DNS NO

admite los registros ALIAS en el vértice raíz (@), simplemente cree registros A que apunten a 192.30.252.153 y 192.30.252.154.

Recursos

Instrucciones de GitHub para un dominio personalizado.

Preguntas y respuestas sobre el desbordamiento de pila: "Dominio personalizado para páginas de proyectos de GitHub"

Audrey Watters - Uso de GitHub para impulsar un proyecto web: cómo y por qué

Alex Cican - Cómo moví mis sitios web a Dropbox y GitHub

Treehouse - Usar las páginas de GitHub para alojar tu sitio web

Lea Páginas de GitHub en línea: https://riptutorial.com/es/github/topic/3759/paginas-de-github

Capítulo 11: Solicitudes de extracción

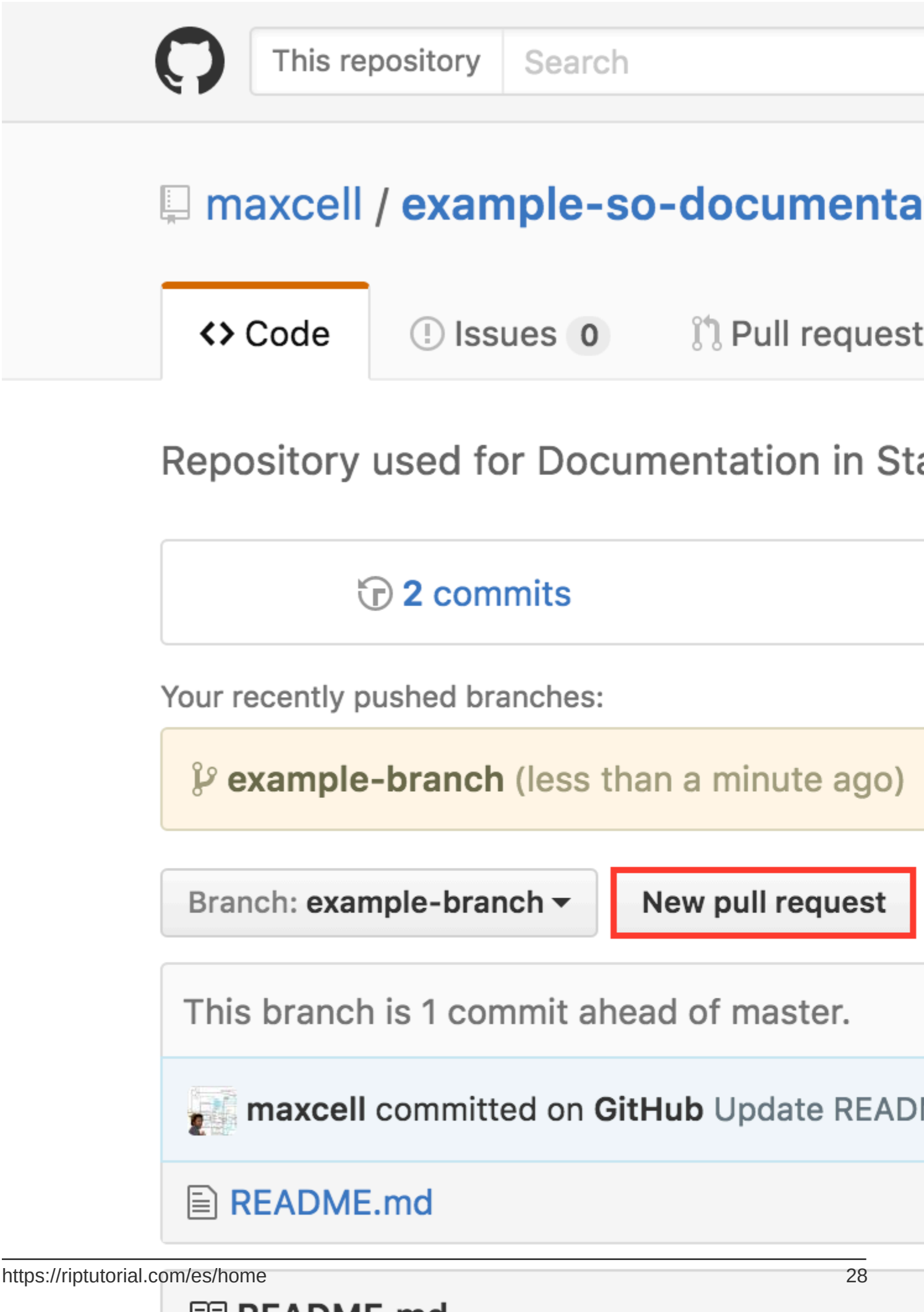
Examples

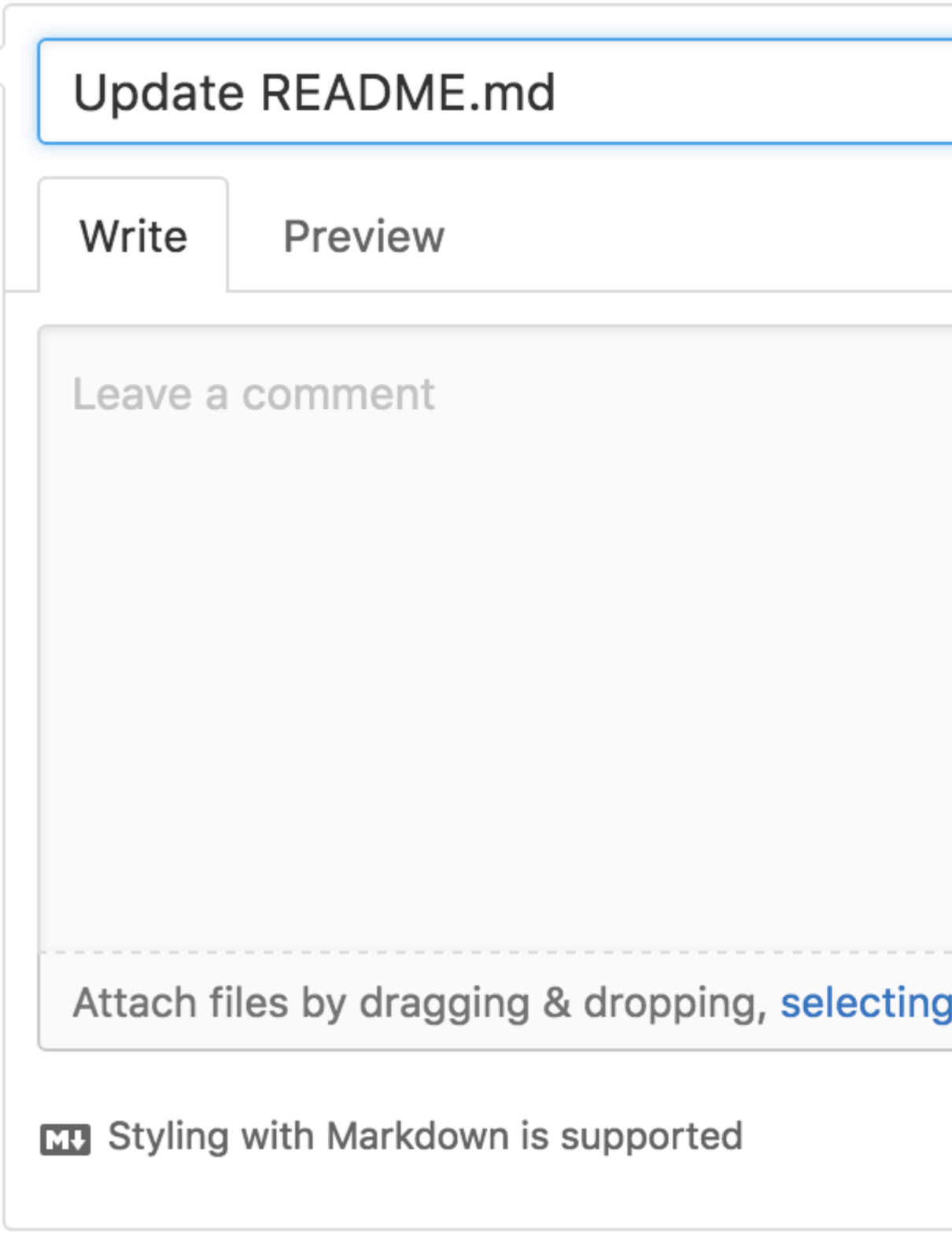
Abrir una solicitud de extracción

Nueva solicitud de extracción

Cuando quiera crear una Solicitud de extracción (digamos que esto se debe a un cambio reciente. ¡Pero también puede hacer esto con un cambio más antiguo!), Puede seguir adelante y dejar que GitHub haga un montón de trabajo pesado por usted y *presione el* botón verde de **solicitud de comparación y** *extracción* (*PARA NO SER CONFUSO CON CLONE O DESCARGAR*) dentro del cuadro de alerta que menciona que acaba de presionar dentro de una rama.

De lo contrario, utilizará el botón **Nueva solicitud de** extracción que se encuentra al lado de su sucursal.


This repository
Search
maxcell / example-so-documenta
Code
Issues 0
Pull request
Repository used for Documentation in Sta
2 commits
Your recently pushed branches:
example-branch (less than a minute ago)
Branch: example-branch
New pull request
This branch is 1 commit ahead of master.
maxcell committed on GitHub Update READ
README.md



Lea Solicitudes de extracción en línea: https://riptutorial.com/es/github/topic/5761/solicitudes-de-extraccion

Capítulo 12: Trabajando con Gitflow

Sintaxis

- git flow <subcommand>
- git flow init
- git flow [característica | lanzamiento | revisión] [inicio | final]

Parámetros

Subcomando	Detalles
en eso	Inicialice un nuevo repositorio git con soporte para el modelo de bifurcación.
característica	Gestiona tus ramas de características.
lanzamiento	Gestiona tus ramas de lanzamiento.
revisión	Gestiona tus sucursales de hotfix.

Observaciones

- concepto de gitflow del autor
- foto de modelo de rama

Examples

Operación en 5 ramas comunes a nivel local.

Uno de los casos de uso más comunes de Gitflow.

1. *Inicializar* repo y definir ramas.

```
$ git flow init
    # if you use default setup, you'll define six types of branches:
    #
    # main branches (lives forever)
    #
    #   1. master:  for production releases
    #   2. develop: for "next release" development
    #
    # supporting branches
    #
    #   3. feature: for a product feature
    #   4. release: for preparation of a new production release
    #   5. hotfix:  for resolving critical bug of production version
    #   6. support
```

```
    #
    # also, two main branches are created: master, develop
```

2. *Iniciar* y *finalizar* una característica

```
$ git flow feature start my_feature
    # create branch 'feature/my_feature' based on the 'develop'

    # made development and commits...

$ git flow feature finish my_feature
    # merge 'feature/my_feature' back to the 'develop'
    # delete 'feature/my_feature'
```

3. *Iniciar* y *finalizar* un lanzamiento

```
$ git flow release start my_release
    # create branch 'release/my_release' based on the 'develop'

    # made bug fixes...

$ git flow release finish my_release
    # merge branch 'release/my_release' to the 'master' and add tag
    # merge branch 'release/my_release' back to the 'develop'
    # delete 'release/my_release'
```

4. *Comenzar* y *terminar* una revisión

```
$ git flow hotfix start my_hotfix
    # create branch 'hotfix/my_hotfix' based on the 'master'

    # made some hotfixes...

$ git flow hotfix finish my_hotfix
    # merge branch 'hotfix/my_hotfix' back to the 'master' and add tag
    # merge branch 'hotfix/my_hotfix' to the 'develop'
    # delete 'hotfix/my_hotfix'
```

Lea Trabajando con Gitflow en línea: https://riptutorial.com/es/github/topic/6231/trabajando-con-gitflow

Capítulo 13: Usando Gist

Introducción

Los gists son una gran manera de compartir tu trabajo. Puede compartir archivos individuales, partes de archivos o aplicaciones completas. Puede acceder a gists en https://gist.github.com .

Cada elemento esencial es un repositorio de Git, lo que significa que se puede bifurcar y clonar. El editor de gist está alimentado por CodeMirror.

Hay dos tipos de gists: gists públicas y gists secretas.

Además, si no has iniciado sesión en GitHub cuando crees tu gist, será un ging anónimo.

Observaciones

Los gists son una gran manera de compartir tu trabajo. Puede compartir archivos individuales, partes de archivos o aplicaciones completas.

Hay dos tipos de gists: gists públicas y gists secretas. Además, si no has iniciado sesión en GitHub cuando crees tu gist, será un ging anónimo.

Gistas públicas

Los gists públicos aparecen en Discover, donde las personas pueden buscar nuevos gists a medida que se crean. También se pueden buscar, por lo que puede usarlos si desea que otras personas encuentren y vean su trabajo.

Gistas secretas

Los gistas secretos no aparecen en Discover y no se pueden buscar. Úsalos para anotar una idea que te surgió en un sueño, crea una lista de tareas o prepara un código o prosa que no esté listo para compartir con el mundo.

Puedes crear tantos gists secretos como quieras.

Gistas anónimas

Si creas un gist sin iniciar sesión en GitHub, será un gist anónimo. Los anónimos pueden ser públicos o secretos. Para eliminar una idea anónima en GitHub.com o GitHub Enterprise, comuníquese con el soporte de GitHub o con el administrador de su sitio, respectivamente. Por favor, proporcione la URL de la esencia que desea eliminar.

Examples

Public Gist

Una idea pública puede ser *casi* cualquier cosa.

Un ejemplo simple de una función de Javascript:

```
function randomInt(min, max) {
return Math.floor((max - min + 1) * Math.random()) + min;
}
```

Secreto gist

Se debe usar una idea secreta para cualquier cosa que no quieras que aparezca públicamente en GitHub. Los gists secretos pueden usarse cuando no desea que las claves privadas sean accesibles al público, o para un código privado en general.

Un ejemplo simple de código JSON que sería mejor para una esencia secreta:

```
{
"id": AKIAIOSFODNN7EXAMPLE,
"secret": wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
}
```

Lea Usando Gist en línea: https://riptutorial.com/es/github/topic/7978/usando-gist

Capítulo 14: Usando los botones de GitHub

Introducción

¿Qué son los botones de GitHub? Los botones de GitHub son botones que puede agregar a su sitio web que redirige a los usuarios a cualquier repositorio que desee.

Observaciones

Créditos:

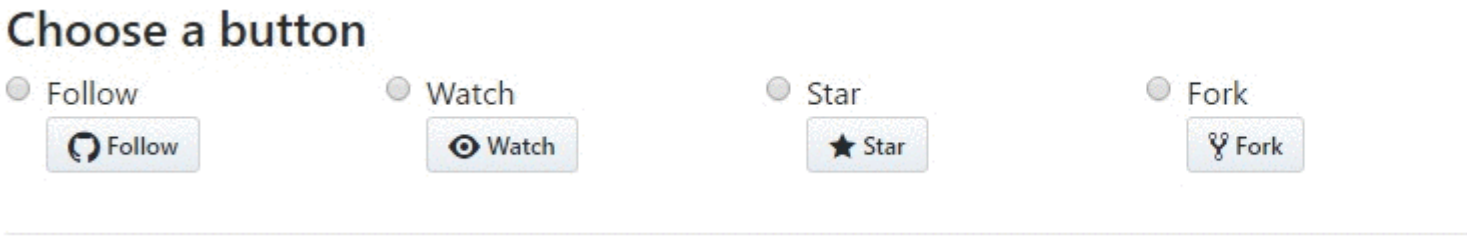
- Imágenes GIF grabadas con Recordit.
- Imágenes estáticas tomadas con la herramienta Snipping
- El editor de código utilizado en los tutoriales completos fue codepen.io

Examples

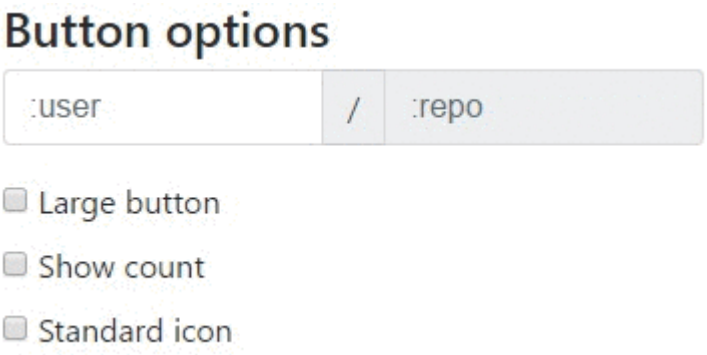
boton de seguir

Un botón de seguimiento es un botón que enlaza con una página de usuario de GitHub y le pide al usuario que siga al usuario. Aquí está cómo crear uno:

1. Ir a github: botones
2. Haga clic en "Seguir"



3. Coloque su nombre de usuario de GitHub en la casilla ": usuario"



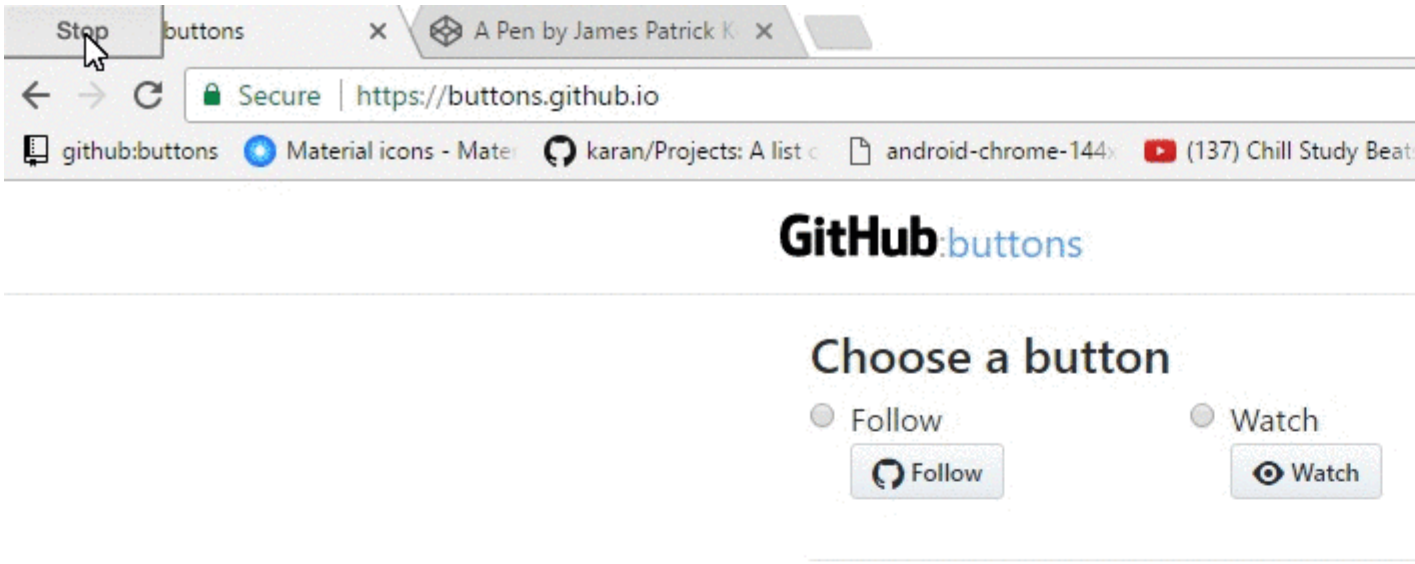
4. Personalice el botón utilizando las casillas "Botón grande", "Mostrar recuento" y "Icono

estándar":

5. Coloque este código en el `<head>` o antes del final del `<body>` de su código:

```
<a class="github-button" href="https://github.com/hubot" aria-label="Follow @hubot on
GitHub">Follow @hubot</a>
```

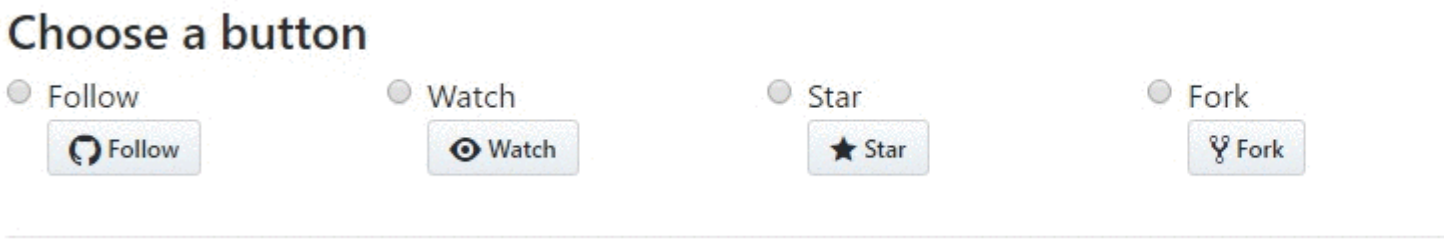
6. Coloque el código de representación de botón personalizado en su código.


Stop
buttons
A Pen by James Patrick K
Secure
https://buttons.github.io
github:buttons
Material icons - Mate
karan/Projects: A list
android-chrome-144
(137) Chill Study Beat
GitHub:buttons
Choose a button
Follow
Follow
Watch
Watch

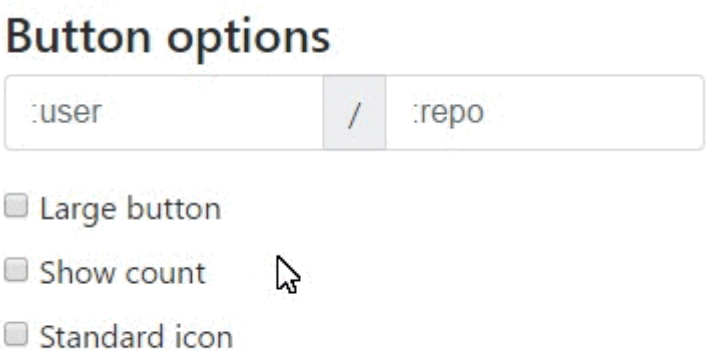
- **Star** un repositorio
- **Bifurcar** un repositorio
- **Descargar** un repositorio
- Listar un **problema** con un repositorio

Aquí es cómo crear algunos:

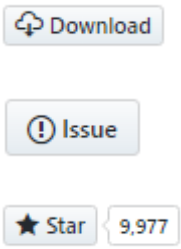
1. Ir a github: botones
2. Haga clic en el tipo de botón que desea crear (Watch, Star, Fork, Download o Issue)



3. Coloque su nombre de usuario de GitHub en la casilla ": usuario", y su repositorio en la casilla ": repo"



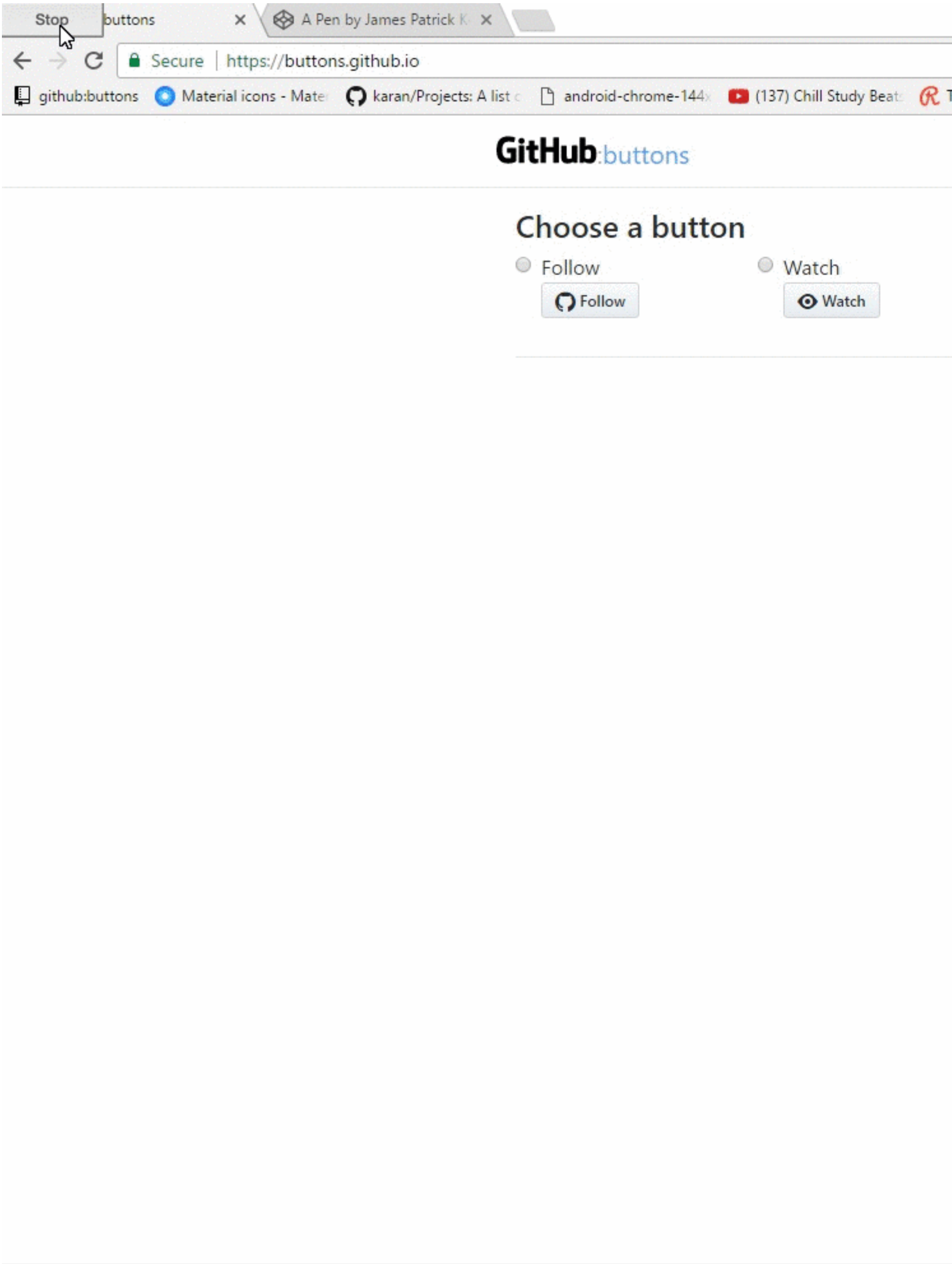
4. Personalice el botón utilizando las casillas "Botón grande", "Mostrar recuento" y "Icono estándar":



5. Coloque este código en el `<head>` o antes del final del `<body>` de su código:

```
<a class="github-button" href="https://github.com/hubot" aria-label="Follow @hubot on GitHub">Follow @hubot</a>
```

6. Coloque el código de representación de botón personalizado en su código.


Stop
buttons
A Pen by James Patrick K
Secure
https://buttons.github.io
github:buttons
Material icons - Mate
karan/Projects: A list
android-chrome-144x
(137) Chill Study Beat
GitHub:buttons
Choose a button
Follow
Watch
Follow
Watch
1J2GxbWa2o (1).gif
1J2GxbWa2o.gif
ss6fW8XoSL (1).gif

https://riptutorial.com/es/github/topic/10585/usando-los-botones-de-github

Capítulo 15: Visualización de la línea de tiempo de GitHub / feeds en su sitio web

Examples

Visualización de la línea de tiempo / feeds de GitHub en su sitio web

Este documento explica cómo mostrar sus feeds / línea de tiempo de GitHub en su sitio web.

Ejemplo: Un ejemplo en vivo está disponible en:

> https://newtonjoshua.com

Línea de tiempo de GitHub:

GitHub proporciona la línea de tiempo pública para cualquier usuario en formato Atom.

Puedes ver tu línea de tiempo en:

> https://github.com/ {{GitHub_username}} .tom

Consulte: https://developer.github.com/v3/activity/feeds

API de Google Feed:

Con la API de fuentes, puede descargar cualquier fuente pública de Atom, RSS o Media RSS utilizando solo JavaScript, por lo que puede combinar las fuentes con su contenido y otras API con solo unas pocas líneas de JavaScript. Esto facilita la integración rápida de feeds en su sitio web.

Consulte: https://developers.google.com/feed/v1/devguide

Cargando la API de JavaScript: para comenzar a utilizar la API de Feed, incluya el siguiente script en el encabezado de su página web.

```
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
```

A continuación, cargue la API de Feed con google.load (módulo, versión, paquete).

```
<script type="text/javascript">
  google.load("feeds", "1");
</script>
```

Especificando la URL del feed: Puede llamar a google.feeds.Feed () de la siguiente manera:

```
var feed = new google.feeds.Feed("https://github.com/{{GitHub_UserName}}.atom");
```

Cargando un feed: .load (callback) descarga el feed especificado en el constructor desde los servidores de Google y llama al callback dado cuando finaliza la descarga.

```
<script type="text/javascript">

    function initialize() {
      feed.load(function(result) {
        if (!result.error) {
          var container = document.getElementById("feed");
          result.feed.entries.forEach(function (feed) {
            var feedTitle= feed.title;
            var feedLink = feed.link;
            var feedDate = formatDate(feed.publishedDate);
            var feedContent = formatContent(feed.content);

           // display the feed in your website
          });
        }
      });
    }
    google.setOnLoadCallback(initialize);

    </script>
```

Llamar al controlador onLoad: setOnLoadCallback (callback) es una función estática que registra la función del controlador especificada para que se llame una vez que se cargue la página que contiene esta llamada, donde callback es una función requerida cuando el documento que contiene está cargado y la API está lista para su uso

```
<script type="text/javascript">
    google.setOnLoadCallback(initialize);
 </script>
```

Configuración del número de entradas de feed: .setNumEntries (num) establece el número de entradas de feed cargadas por este feed en num. Por defecto, la clase Feed carga cuatro entradas.

```
var feed = new google.feeds.Feed("https://github.com/{{GitHub_UserName}}.atom");
feed.setNumEntries(500);
```

Ahora puede formatear y mostrar sus feeds / línea de tiempo de GitHub en su sitio web.

Lea Visualización de la línea de tiempo de GitHub / feeds en su sitio web en línea: https://riptutorial.com/es/github/topic/7479/visualizacion-de-la-linea-de-tiempo-de-github---feeds-en-su-sitio-web

Creditos

S. No	Capítulos	Contributors
1	Empezando con github	BadAllOff, BrechtDeMan, Community, H. Pauwelyn, Hamzawey, Hugo, intboolstring, Kronos, Mateusz Piotrowski, Minhas Kamal, Nicholas Qiao, rpadovani
2	¿Cómo crear etiquetas de GitHub personalizadas?	Ahmad Awais
3	Actualizar un repositorio bifurcado	Derek Liu
4	Clonando un repositorio desde GitHub	demonplus, geek1011, Hamzawey, James Kerrane, Mateusz Piotrowski
5	Copia de seguridad de GitHub	geek1011
6	Cuestiones	geek1011, Hamzawey, SuperBiasedMan
7	descargar un solo archivo desde el repositorio GitHub	ownsourcing dev training
8	Eliminar datos confidenciales o archivos grandes	Gautam Krishna R, Kronos
9	GitHub Desktop	creyD
10	Páginas de GitHub	BrechtDeMan, geek1011, Mono
11	Solicitudes de extracción	Maxcell
12	Trabajando con Gitflow	Derek Liu
13	Usando Gist	Kronos, tehp
14	Usando los botones de GitHub	James Kerrane

15	Visualización de la línea de tiempo de GitHub / feeds en su sitio web	Hugo, Newton Joshua