EBook Gratuito

APPRENDIMENTO

github

Free unaffiliated eBook created from
Stack Overflow contributors.

#github

Sommario

Di

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

Capitolo 1: Iniziare con github

Osservazioni

Questa sezione fornisce una panoramica su cosa sia GitHub e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di github e collegarsi agli argomenti correlati. Poiché la Documentazione per github è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione o configurazione

GitHub è una vasta collezione di repository Git. In altre parole, puoi pensare a GitHub come a una collezione di molti progetti!

Creare un account

- Visita la pagina principale di GitHub qui
- Scegli un nome utente, inserisci il tuo indirizzo email e scegli una password sicura e sei pronto per partire!

Strumenti utili

Per i principianti di Git / GitHub, capire come funziona il controllo della versione potrebbe essere inizialmente fonte di confusione. Esiste una versione GUI di GitHub che è possibile scaricare e utilizzare. GitHub Desktop è proprio questo strumento.

Creare il tuo primo deposito

Puoi pensare a un repository come a un progetto. È possibile creare un repository online o offline. Segui i passaggi seguenti:

in linea

1. Primo login e vai al tuo profilo.
2. Passare alla scheda "Archivi" nella parte superiore della pagina
3. Premi il pulsante verde "Nuovo" e sei pronto per il rombo!

disconnesso

1. Scarica e installa git (scegli il sistema operativo in esecuzione)
2. Dopo il download e l'installazione, è possibile utilizzare lo strumento della riga di comando oppure è possibile scaricare un client della GUI.
3. Dopo l'installazione, crea un account su github
4. In alto a destra, fai clic su + e scegli di creare un nuovo repository o di importarne uno esistente.
5. Se ne selezioni uno nuovo, inserisci il nome del repository e scegli se renderlo pubblico o privato.
6. Fare clic: Crea repository

NB I repository privati non sono disponibili per gli utenti gratuiti.

File README

Se il tuo progetto non ha README.md, GitHub può analizzare README.rdoc per visualizzare i dettagli. Se ha entrambi, utilizzerà README.md, ignorando silenziosamente rdoc.

Un file README può includere-

titolo del progetto

Descrivi brevemente il tuo progetto. È inoltre possibile fornire il link del sito Web del progetto, i badge, la comunità e le informazioni di contatto (ad es. Email, sito social).

Scaricare

Link eseguibile (file eseguibile o minisito o file di installazione). Ci possono essere collegamenti anche alle versioni precedenti.

Installazione

Come può essere usato il tuo lavoro. Può includere i prerequisiti, le impostazioni, le librerie di terze parti, l'uso, le precauzioni, ecc.

Dimostrazione

Può includere esempio di codice, file GIF, collegamento video o anche schermate.

autori

Nomi degli autori, informazioni di contatto, ecc.

Ringraziamenti

Elenco di persone o comunità aiutato e ispirato durante il progetto

contribuire

Istruzioni per contribuire (es. Aggiungere funzionalità, segnalare bug, inviare patch) al progetto. Può includere anche un link alla documentazione.

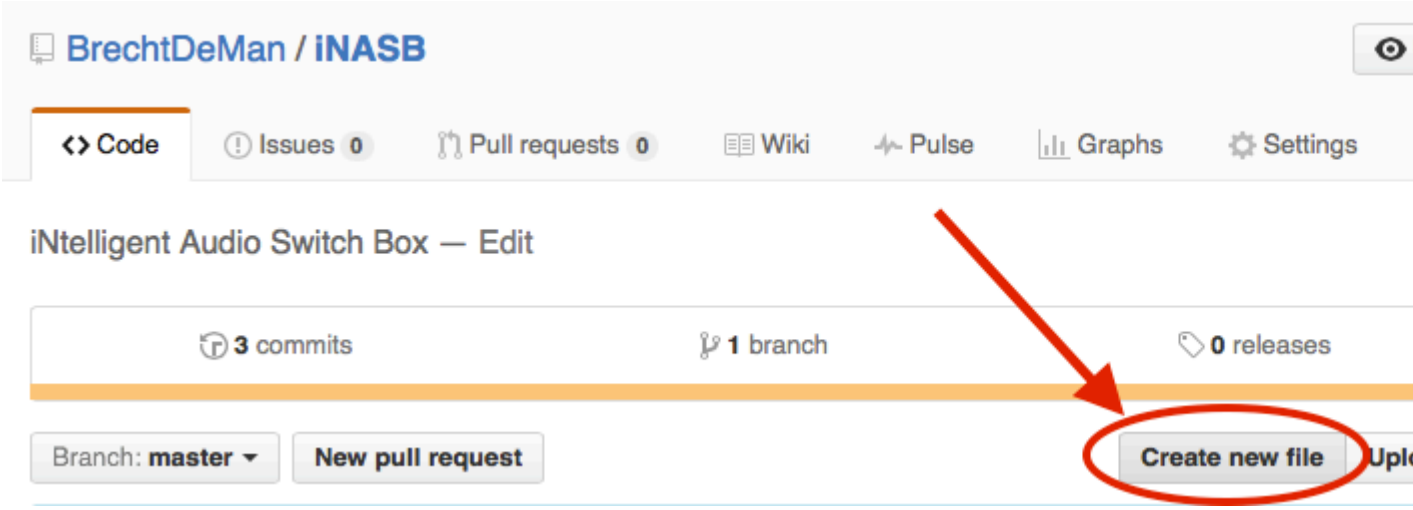
Licenza

Fai una breve introduzione sulla tua licenza. Puoi anche dare un link al sito della licenza.

File LICENZA

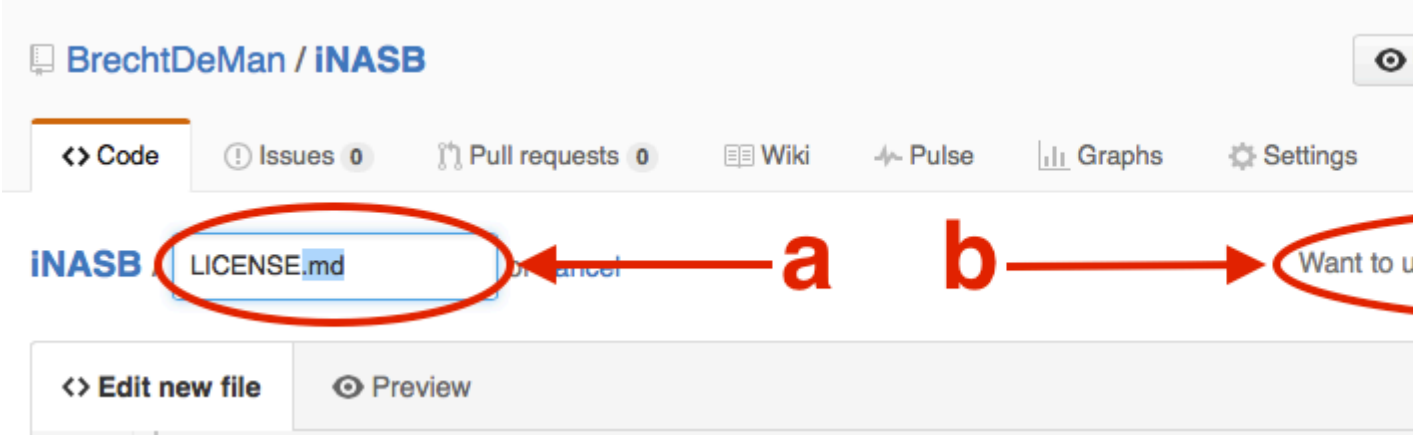
GitHub ti aiuta a aggiungere rapidamente una licenza al tuo repository, come alternativa per aggiungere il tuo file text / markdown.

1. Nel tuo repository, fai clic su "Crea nuovo file"

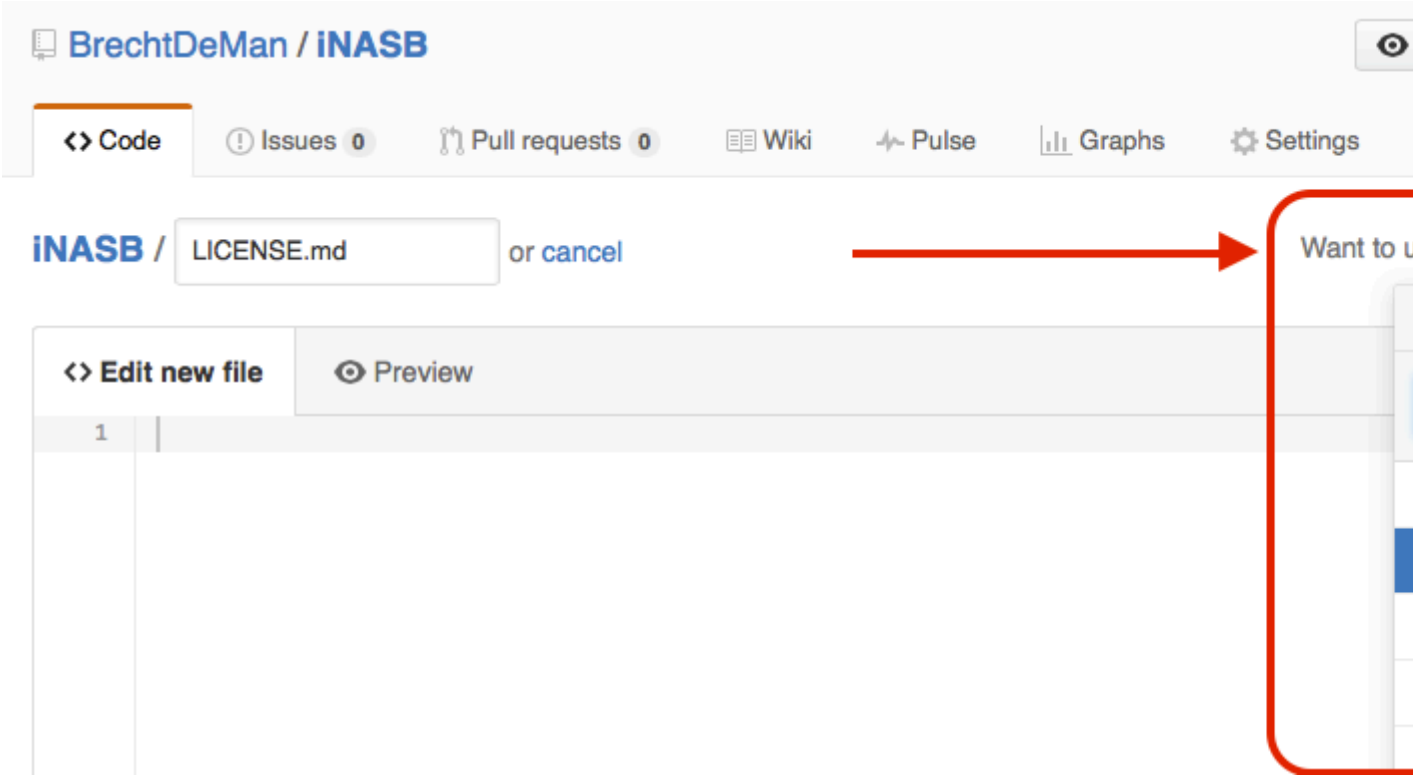


2. Nella pagina successiva:

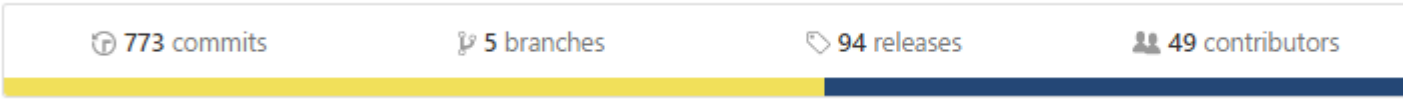
    1. Digitare `LICENSE.md` o `LICENSE.txt` come nome file del nuovo file.
    2. *Vuoi usare un nuovo modello?* apparirà la finestra di dialogo.



3. Scegli la tua licenza preferita.



4. La licenza che potresti vedere nei dettagli del repository:



Da Q & A - Come aggiungere la licenza a un progetto Github esistente

GitHub Flavored Markdown

GitHub espande la sintassi di Markdown per fornire nuove utili funzionalità.

Intestazione

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

Header1

Header2

Header3

Header4

Header5

Header6

H1

H2

enfasi

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

Italic1 Italic2
Bold1 Bold2
Bold_Italic
~~Strikethrough~~

Linea orizzontale

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

Elenco

```
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

tavolo

```
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

Codice

```
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-

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

Citazione

```
> 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.

collegamento

```
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

Immagine

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

Elenchi di attività

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

- [x] completed item
- [ ] incomplete item

emoji

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

🐙👍📖👻💡😈

Per tutte le emoji di GitHub visitate- Emoji Cheat Sheet .

Riferimenti SHA

Qualsiasi riferimento a un hash SHA1 di un commit verrà convertito in un link al commit stesso su GitHub:

```
e7909ea4fbb162db3f7f543d43c30684a3fb745f
```

Write Preview

`e7909ea`

Richiama referenze di richieste e problemi

Qualsiasi riferimento a una richiesta di pull o un problema verrà automaticamente collegato a tale

richiesta o problema.

Questo può essere fatto mettendo un `#` di fronte al numero di problema / Pull Request.

Leggi Iniziare con github online: https://riptutorial.com/it/github/topic/1214/iniziare-con-github

Capitolo 2: Aggiorna un repository a forcella

Osservazioni

- GitHub Help: Configurare un telecomando per una forcella
- GitHub Help: Sincronizzare un fork
- ans popolare in StackOverFlow

Examples

Configura un telecomando per la tua forcella quindi sincronizza la tua forcella (ramo principale)

1. Config un telecomando per la mia forcella

```
$ 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. Sincronizza la mia forchetta 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. Sincronizza la mia forchetta su Github

```
$ git push origin master
```

Leggi Aggiorna un repository a forcella online: https://riptutorial.com/it/github/topic/3758/aggiorna-un-repository-a-forcella

Capitolo 3: Backup di GitHub

Examples

Clonazione di tutti i repository per un nome utente

Eseguire il seguente comando, sostituendo il nome utente con il nome utente, per clonare tutti i repository GitHub per quell'utente nella directory corrente.

```
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
```

Questo clonerà solo i primi 100 repository.

Leggi Backup di GitHub online: https://riptutorial.com/it/github/topic/3760/backup-di-github

Capitolo 4: Clonazione di un repository da GitHub

Sintassi

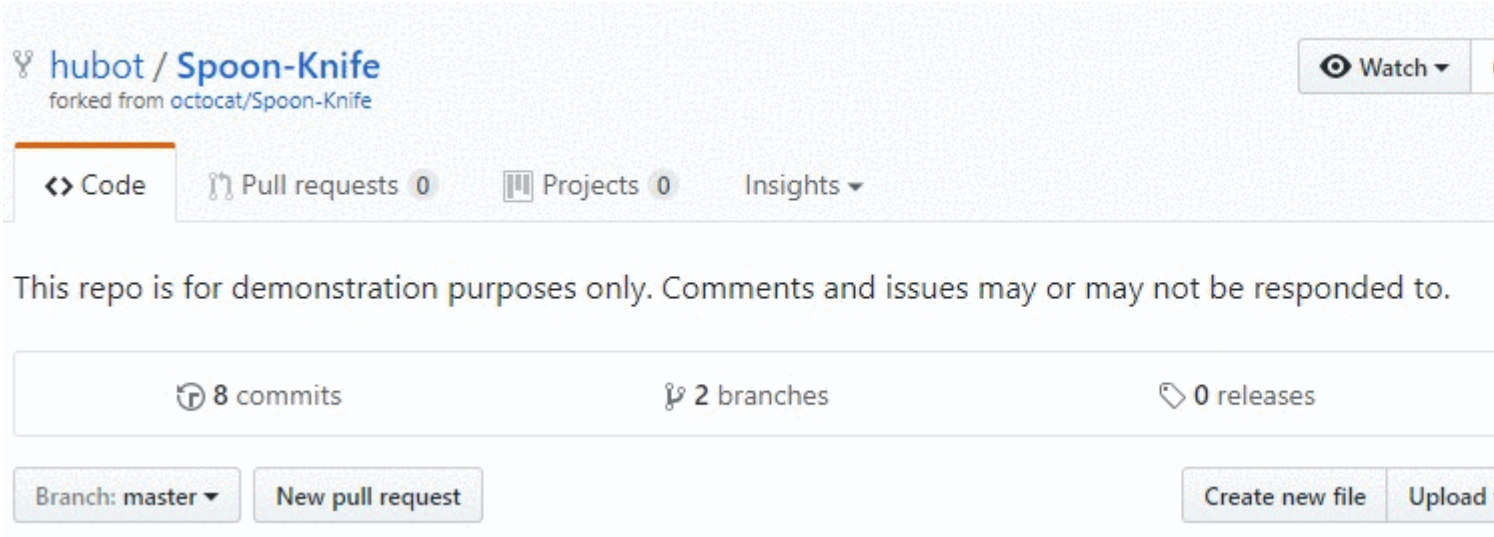
- git clone github.com/username/repository

Examples

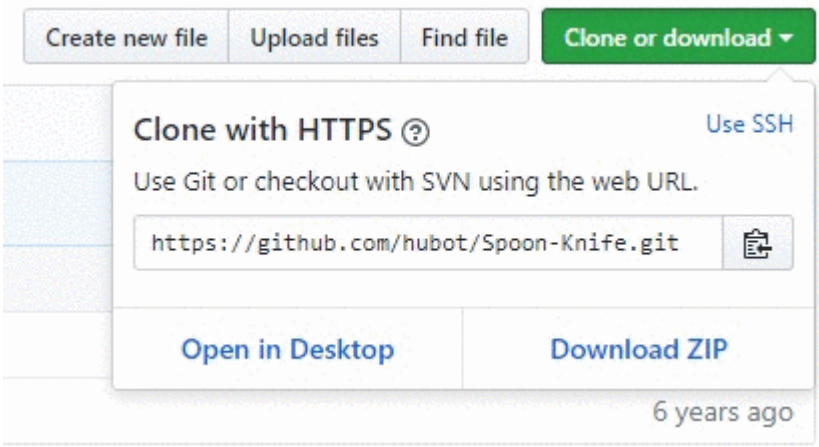
Clona un repository

1. Vai al repository che vuoi clonare (qualcosa come: https://github.com/ *username* / *repo*)

2. Sulla destra, fai clic sul pulsante verde denominato *clone o download*



3. Apparirà una piccola finestra, copia l'url (qualcosa come: https://github.com/ *username* / *repo* .git)



4. Aprire una finestra di terminale sulla macchina su cui si desidera clonare quel progetto
5. Passare dalla riga di comando alla posizione in cui si desidera clonare il progetto
6. Immettere il comando: git clone <copy_url_from_step_3>
7. premere Invio
8. Apparirà qualcosa come il seguente:

   Clonazione in `<repo_name>` ...

   remote: conteggio oggetti: 10, fatto.

   remote: compressione di oggetti: 100% (8/8), fatto.

   remove: Total 10 (delta 1), riutilizzato 10 (delta 1)

   Disimballaggio degli oggetti: 100% (10/10), finito.

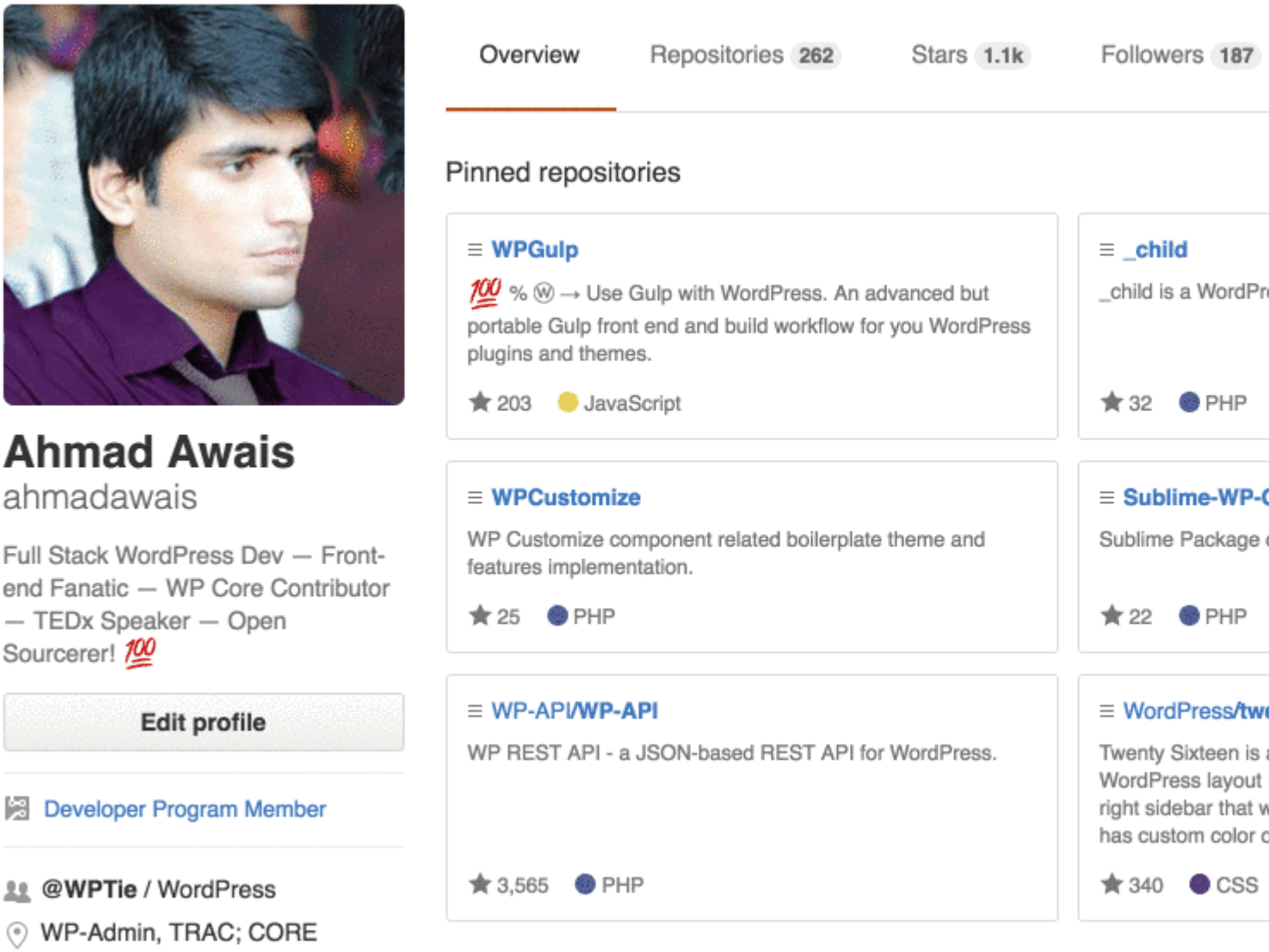
Leggi Clonazione di un repository da GitHub online:
https://riptutorial.com/it/github/topic/3761/clonazione-di-un-repository-da-github

Capitolo 5: Come creare etichette GitHub personalizzate?

Examples

Crea etichette GitHub personalizzate!

Ecco una rapida GIF per rendere il processo il più semplice possibile.



Le etichette possono essere applicate a problemi e richiama richieste per indicare priorità, categoria o qualsiasi altra informazione che ritieni utile.

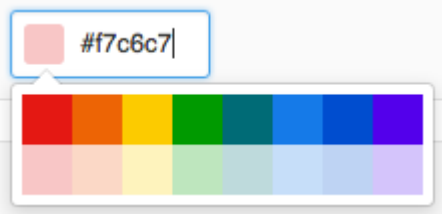
Su GitHub, vai alla pagina principale del repository.

1. Sotto il nome del tuo repository, fai clic su Issues o Pull request.
2. Pulsante Etichette ProblemaOvanti nel campo di ricerca, fare clic su Etichette.

3. Fai clic su Nuova etichetta per creare una nuova etichetta o fai clic su Modifica per modificarne una esistente.

4. Nella casella di testo, digitare il nuovo nome dell'etichetta.

5. Seleziona un colore per l'etichetta dalla barra dei colori. È possibile personalizzare questo colore modificando il numero esadecimale sopra la barra dei colori.



6. Fai clic su Crea etichetta per salvare la nuova etichetta.

Spero possa essere d'aiuto. Evitalo se lo fa.

Leggi Come creare etichette GitHub personalizzate? online:
https://riptutorial.com/it/github/topic/7159/come-creare-etichette-github-personalizzate-

Capitolo 6: GitHub Desktop

introduzione

Come installare e lavorare con GitHub Desktop?

GitHub Desktop è - come suggerisce il nome - un ambiente desktop per Windows e MacOS che include le caratteristiche principali di Git come la clonazione, la spinta, l'estrazione (sincronizzazione in GitHub Desktop), l'unione ...

Lo scopo principale dei client desktop è fornire un modo più semplice di lavorare con git (e GitHub). In background utilizza gli stessi comandi che la maggior parte degli utenti userebbe dalla riga di comando.

Examples

Installazione e configurazione

L'installazione è abbastanza semplice in quanto vi sono programmi di installazione separati per macchine MacOS e Windows disponibili qui . Attualmente sono disponibili per il download due versioni: una beta e una stabile.

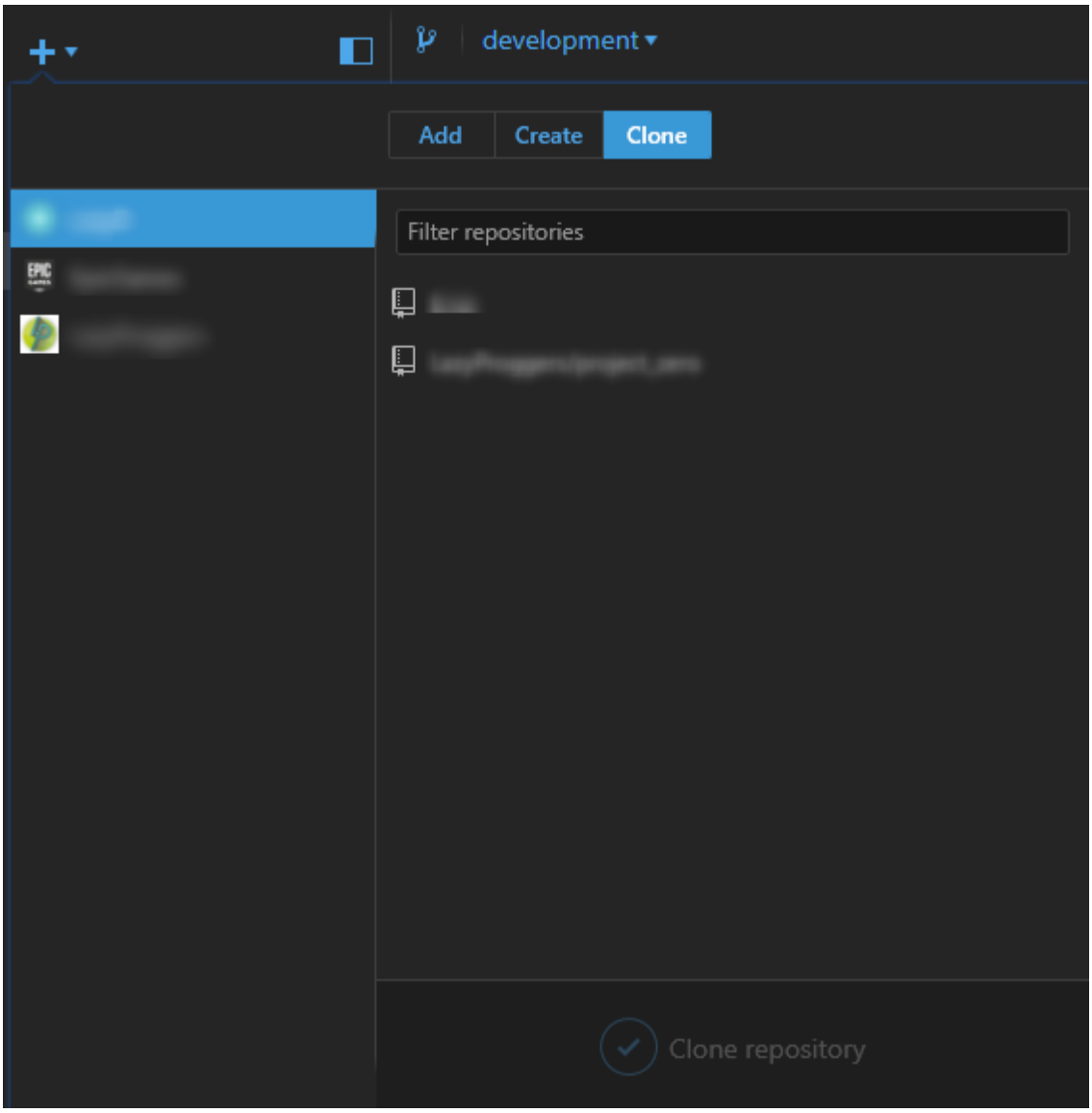
L'installazione inizierà dopo aver scaricato il programma e sarà necessario accedere con le credenziali di GitHub. Questo è davvero l'unico passo perché dopo puoi iniziare a creare un repository o clonare uno.

> Nota: durante l'installazione non verrà installato solo GitHub Desktop, ma anche Git. Quindi non è necessario installarlo separatamente.

Clonazione di un repository

Come con GitHub Desktop, la maggior parte del lavoro è piuttosto semplice: selezioni "Clona un repository" (nella versione stabile il plus in alto a sinistra) e ci sono alcuni repository (i tuoi e i repository di ogni azienda in cui ti trovi)) consigliato. In alternativa puoi incollare un link a qualsiasi altro repository che potresti voler clonare.

> Nota: nella versione più recente (beta) non ci sono raccomandazioni (non jet?).



branching

È possibile selezionare un ramo in alto a sinistra. Quando hai selezionato il ramo giusto devi premere il pulsante di sincronizzazione (in alto a destra) che ora è lo stesso di `git checkout BRANCHNAME` .

> Nella versione precedente è possibile visualizzare 2 diversi rami contemporaneamente e confrontare i push. Inoltre puoi vedere una timeline del tuo progetto (vedi sotto)

Creare un nuovo ramo

È possibile creare un nuovo ramo facendo clic sul simbolo del ramo (vecchio client) o su `File --> New Branch` .

> Si noti che è possibile selezionare il ramo che il nuovo ramo utilizza come base

facendo clic sul nome del ramo.

Spingi e tira (o: il pulsante di sincronizzazione)

Pull (sincronizzazione)

Come nella riga di comando, è necessario richiamare lo stato corrente del repository una volta ogni tanto. In GitHub Desktop questo processo è chiamato dal pulsante di `sync` nell'angolo in alto a destra.

Spingere

Quando hai apportato modifiche locali e desideri inviarle, esegui un commit scrivendo qualcosa nella casella di testo di riepilogo. Quindi premi `Commit to YOURCURRENTBRANCH` . Ora devi premere il pulsante di sincronizzazione e la tua spinta è fatta.

> Nota: è possibile utilizzare emoticon, citazioni e riferimenti ad altri commit o problemi direttamente dalla casella di testo.

Quindi il pulsante **Sync** può essere usato per `Push` , `Pull` o `Checkout` .

Leggi GitHub Desktop online: https://riptutorial.com/it/github/topic/10023/github-desktop

Capitolo 7: Lavorare con Gitflow

Sintassi

- git flow <sottocomando>
- git flow init
- git flow [funzione | versione | hotfix] [inizio | fine]

Parametri

sottocomando	Dettagli
dentro	Inizializza un nuovo repository git con supporto per il modello di branching.
caratteristica	Gestisci i tuoi rami di funzionalità.
pubblicazione	Gestisci i tuoi rami di rilascio.
hotfix	Gestisci i tuoi rami hotfix.

Osservazioni

- concetto di gitflow dall'autore
- immagine del modello di ramo

Examples

Operazione su 5 filiali comuni a livello locale

Uno dei casi d'uso più comuni di Gitflow

1. *Inizializza il* repository e definisce i rami

```
$ 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. *Avvia* e *termina* una caratteristica

```
$ 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. *Inizia* e *termina* una versione

```
$ 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. *Avvia* e *termina* una correzione

```
$ 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'
```

Leggi Lavorare con Gitflow online: https://riptutorial.com/it/github/topic/6231/lavorare-con-gitflow

Capitolo 8: Pagine GitHub

Examples

Utilizzo del generatore di pagine automatico per un repository

1. Vai al sito Web GitHub
2. Apri il tuo repository
3. Clicca su Impostazioni
4. Sotto GitHub Pages, fai clic su "Avvia generatore di pagine automatico"
5. Seguire le istruzioni

Usare Git per creare pagine da zero

1. Creare un nuovo repository o clonarne uno esistente.
2. Crea un nuovo ramo chiamato `gh-pages` senza alcuna storia

```
$ 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. Aggiungi un file index.html alla radice del repository.

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

4. Spingere su Github.

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

Ora puoi caricare il tuo nuovo sito Github Pages su `http(s)://<username>.github.io/<projectname>`

Creazione di un URL personalizzato per la tua pagina GitHub

Avrai bisogno di un nome di dominio da un registrar .

Nel ramo `gh-pages` del tuo repository di progetto o nella sezione principale del tuo repository `username.github.io` , crea un file CNAME con i contenuti `www.yourdomain.com` - il dominio *canonico* .

Alla pagina di configurazione del dominio del tuo registrar, indirizza il tuo dominio al tuo sito web GitHub. Imposta due record CNAME (uno per l'apice di root (@) e uno per www). Entrambi dovrebbero puntare a `username.github.io` o `username.github.io/repository` . Se il tuo provider DNS NON supporta i record ALIAS sull'apice di root (@), crea semplicemente record A che puntano a 192.30.252.153 e 192.30.252.154.

risorse

Istruzioni GitHub per un dominio personalizzato

Domande e risposte sull'overflow dello stack: "Dominio personalizzato per le pagine del progetto GitHub"

Audrey Watters - Utilizzo di GitHub per alimentare un progetto Web: come e perché

Alex Cican - Come ho spostato i miei siti web su Dropbox e GitHub

Treehouse: utilizzo di pagine GitHub per ospitare il tuo sito web

Leggi Pagine GitHub online: https://riptutorial.com/it/github/topic/3759/pagine-github

Capitolo 9: Problemi

Examples

Creare un problema

1. Vai alla pagina GitHub per il progetto in cui desideri creare un problema.
2. Clicca su ***Problemi*** .
3. In alto a destra, fai clic su ***Nuovo numero*** .
4. Inserisci il titolo del problema.
5. Inserisci il corpo del problema (inclusi i registri, i frammenti di codice, ecc.)
6. *Facoltativo:* per visualizzare il problema prima di inviarlo, fai clic sull'anteprima.
7. Fai clic su ***Invia nuovo numero*** .

Leggi Problemi online: https://riptutorial.com/it/github/topic/3757/problemi

Capitolo 10: Pull Requests

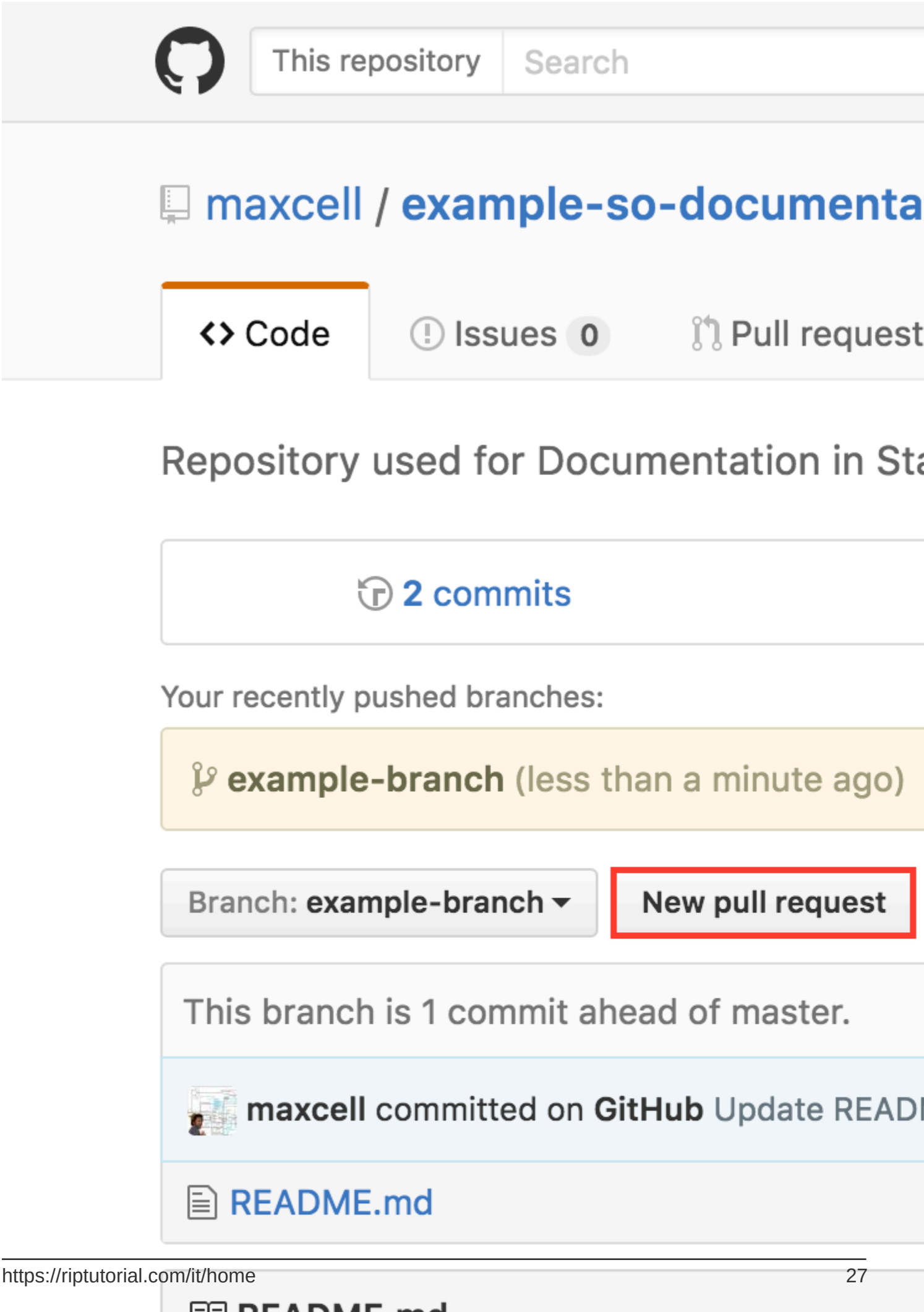
Examples

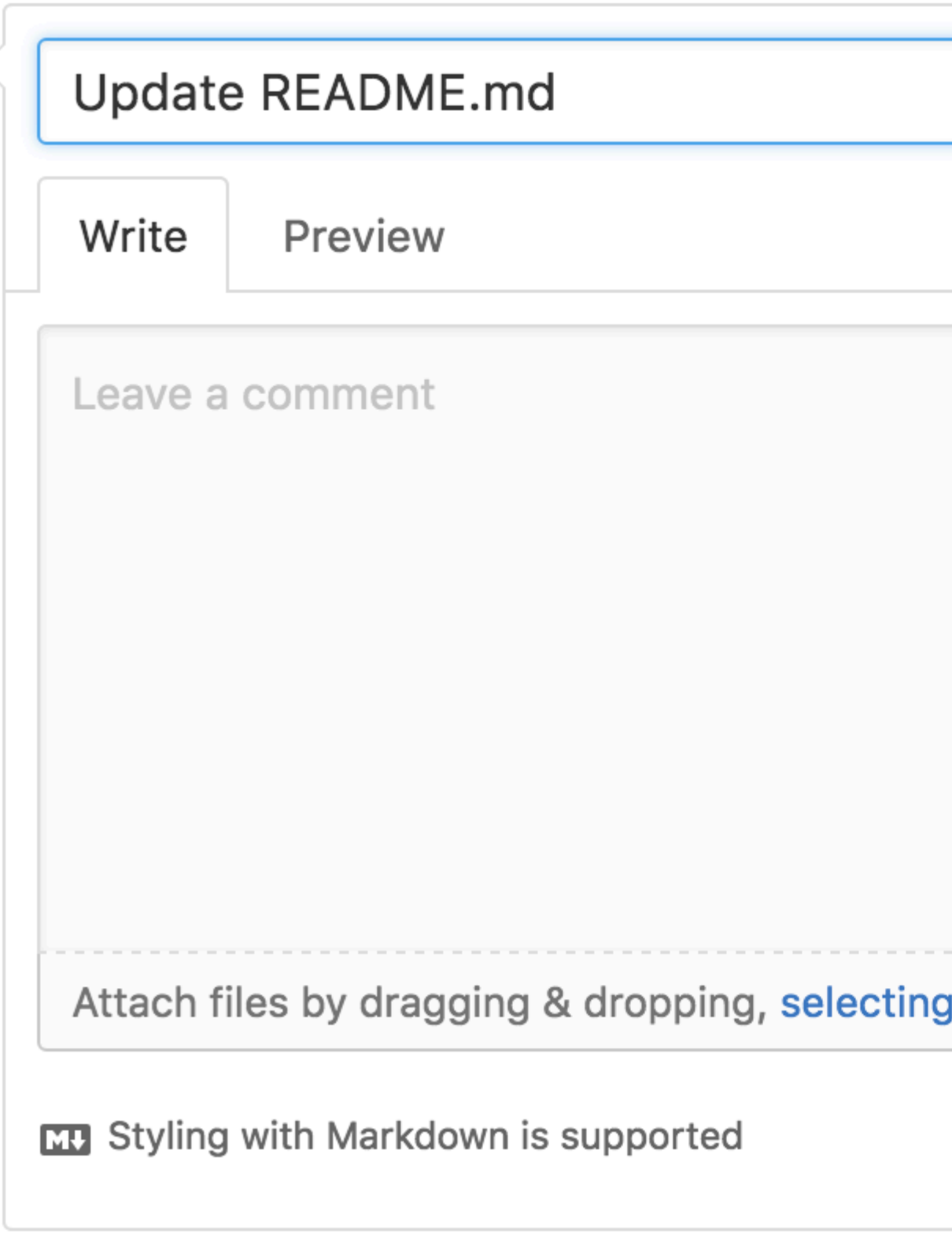
Apertura di una richiesta di pull

Nuova richiesta di pull

Ogni volta che vuoi creare una richiesta di pull (diciamo che è una variazione recente, ma puoi farlo anche con una modifica più vecchia!), Puoi andare avanti e lasciare che GitHub faccia un sacco di lavori pesanti per te e premi il pulsante verde **Confronta e richiedi** (*NON PER ESSERE CONFUSO CON CLONE O DOWNLOAD*) all'interno della casella di avviso che menziona che hai appena inserito all'interno di un ramo.

Altrimenti, utilizzerai il pulsante **Nuova richiesta pull che si** trova accanto al tuo ramo.


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



Leggi Pull Requests online: https://riptutorial.com/it/github/topic/5761/pull-requests

Capitolo 11: Rimozione di dati sensibili o file di grandi dimensioni

introduzione

Se si commettono dati sensibili, come una password o una chiave SSH in un repository Git, è possibile rimuoverli dalla cronologia. Per rimuovere completamente i file indesiderati dalla cronologia di un repository, è possibile utilizzare il comando git filter-branch o BFG Repo-Cleaner.

Osservazioni

1. Spiega ai tuoi collaboratori di rebase, non unire, eventuali rami creati dalla tua vecchia cronologia del repository (contaminato). Un commit di unione potrebbe reintrodurre parte o tutta la storia contaminata che hai appena passato alla purga.

2. Dopo che è trascorso un po 'di tempo e sei sicuro che git filter-branch non ha effetti collaterali indesiderati, puoi forzare la dereferenziazione di tutti gli oggetti nel tuo repository locale e raccogliere i dati con i seguenti comandi (usando Git 1.8.5 o successivi):

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

   git reflog expire --expire = now --all

   git gc --prune = now

Examples

Utilizzando il filtro-ramo

```
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
```

Aggiungi il tuo file con dati sensibili a .gitignore per assicurarti di non commetterlo accidentalmente di nuovo.

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

Invia il tuo repo locale a GitHub

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

Per rimuovere il file sensibile dalle tue versioni codificate, devi anche forzare a spingere i tuoi tag Git:

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

Uso del detergente Repo BFG

BFG Repo cleaner è un'alternativa al git filter-branch. Può essere usato per rimuovere dati sensibili o file di grandi dimensioni che sono stati commessi in modo errato come i binari compilati dalla fonte. È scritto in Scala.

Sito web del progetto: BFG Repo Cleaner

Requisiti

Java Runtime Environment (Java 7 o versione successiva - BFG v1.12.3 era l'ultima versione per supportare Java 6). La libreria Scala e tutte le altre dipendenze sono piegate nel contenitore scaricabile.

Rimuovi file con dati sensibili

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

Leggi Rimozione di dati sensibili o file di grandi dimensioni online:
https://riptutorial.com/it/github/topic/8170/rimozione-di-dati-sensibili-o-file-di-grandi-dimensioni

Capitolo 12: scarica un singolo file dal repository GitHub

Examples

da un repository pubblico utilizzando la riga di comando e rinominando il file

Questo esempio prende il file Node.gitignore dal repository gitignore di GitHub, lo scarica nella directory di lavoro corrente e lo rinomina in .gitignore - tutte le azioni molto tipiche per qualcuno che avvia un nuovo progetto node.js.

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

trova l'url del file che vuoi scaricare

1. navigare fino al file desiderato in un repository
2. fai clic sul pulsante "raw"
3. copia l'url dalla barra degli indirizzi

Guarda il seguente esempio dal repository gitignore di GitHub:
http://github.com/github/gitignore/raw/master/Node.gitignore

Puoi riconoscere rapidamente un URL che funzionerà per scaricare un singolo file o scaricare la pagina html. Cerca la sottodirectory / raw / subito prima del nome del ramo.

Leggi scarica un singolo file dal repository GitHub online:
https://riptutorial.com/it/github/topic/10898/scarica-un-singolo-file-dal-repository-github

Capitolo 13: Usando Gist

introduzione

Gli elenchi sono un ottimo modo per condividere il tuo lavoro. È possibile condividere singoli file, parti di file o applicazioni complete. Puoi accedere agli elenchi su https://gist.github.com .

Ogni gist è un repository Git, il che significa che può essere biforcuto e clonato. L'editor di gist è alimentato da CodeMirror.

Ci sono due tipi di gists: public gists e secret gists.

Inoltre, se non hai effettuato l'accesso a GitHub quando crei il tuo gist, sarà un anonimo gist.

Osservazioni

Gli elenchi sono un ottimo modo per condividere il tuo lavoro. È possibile condividere singoli file, parti di file o applicazioni complete.

Ci sono due tipi di gists: public gists e secret gists. Inoltre, se non hai effettuato l'accesso a GitHub quando crei il tuo gist, sarà un anonimo gist.

Gists pubblici

Gli elenchi pubblici vengono visualizzati in Discover, in cui le persone possono sfogliare i nuovi elenchi man mano che vengono creati. Sono anche ricercabili, quindi puoi usarli se desideri che altre persone trovino e vedano il tuo lavoro.

Secret Gists

Gli elenchi segreti non vengono visualizzati in Discover e non sono ricercabili. Usali per annotare un'idea che ti è venuta in mente, creare una lista di cose da fare o preparare un codice o una prosa che non è pronta per essere condivisa con il mondo.

Puoi creare tutti gli elenchi segreti che desideri.

Elenchi anonimi

Se crei un gist senza accedere a GitHub, sarà un anonimo gist. Gli anonimi possono essere pubblici o segreti. Per eliminare un Gist anonimo su GitHub.com o GitHub Enterprise, contatta l'assistenza GitHub o l'amministratore del tuo sito, rispettivamente. Si prega di fornire l'URL degli elementi che si desidera eliminare.

Examples

Gist pubblico

Un elenco pubblico può essere *praticamente* qualsiasi cosa.

Un semplice esempio di una funzione Javascript:

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

Secret Gist

Un segreto dovrebbe essere usato per tutto ciò che non vuoi apparire pubblicamente su GitHub. Gli elenchi segreti possono essere utilizzati quando non si desidera che le chiavi private siano accessibili al pubblico o per il codice privato in generale.

Un semplice esempio di codice JSON che sarebbe più adatto per un segreto:

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

Leggi Usando Gist online: https://riptutorial.com/it/github/topic/7978/usando-gist

Capitolo 14: Utilizzo dei pulsanti GitHub

introduzione

Cosa sono i pulsanti GitHub? I pulsanti GitHub sono pulsanti che puoi aggiungere al tuo sito web per reindirizzare gli utenti verso qualsiasi repository che ti piace!

Osservazioni

Titoli di coda:

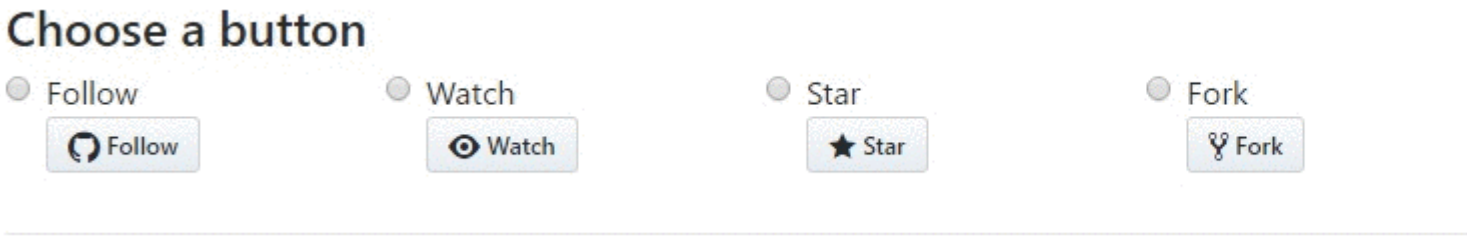
- Immagini GIF registrate con Recordit
- Immagini statiche scattate con lo strumento di cattura
- L'editor di codice utilizzato nei tutorial completi era codepen.io

Examples

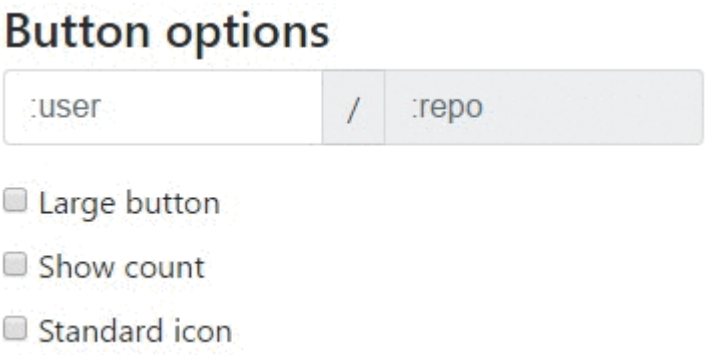
Segui il pulsante

Un pulsante segui è un pulsante che si collega a una pagina utente GitHub e richiede all'utente di seguire l'utente. Ecco come crearne uno:

1. Vai su github: pulsanti
2. Clicca "Segui"



3. Inserisci il tuo nome utente GitHub nella casella ": utente"



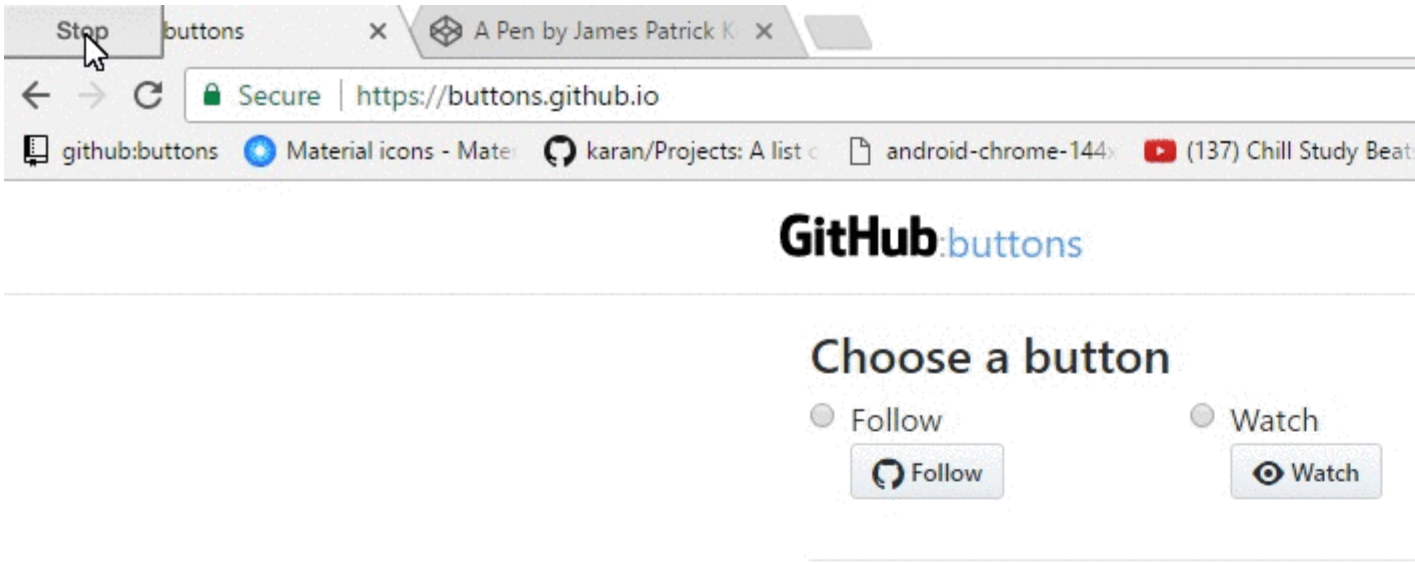
4. Personalizza il pulsante utilizzando le caselle "Pulsante grande", "Mostra numero" e "Icona

standard":

5. Inserisci questo codice in `<head>` o prima della fine del `<body>` del tuo codice:

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

6. Inserisci il codice di rendering del pulsante personalizzato nel codice.

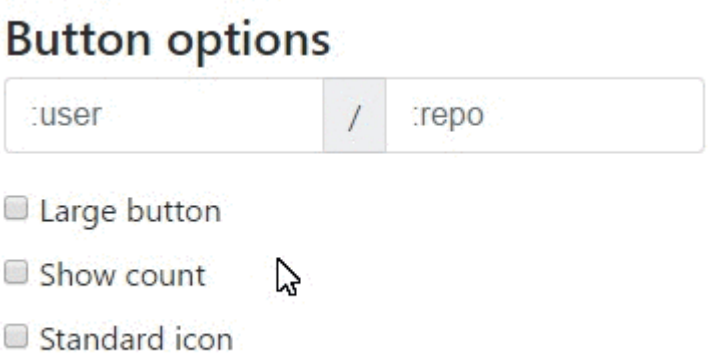

Stop
buttons
A Pen by James Patrick K
Secure
https://buttons.github.io
github:buttons
Material icons - Mate
karan/Projects: A list
android-chrome-144
(137) Chill Study Beat
GitHub:buttons
Choose a button
Follow
Follow
Watch
Watch

- **Star** un repository
- **Fork** un repository
- **Scarica** un repository
- Elenca un **problema** con un repository

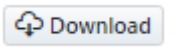
Ecco come crearne alcuni:

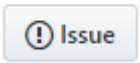
1. Vai su github: pulsanti
2. Fare clic sul tipo di pulsante che si desidera creare (orologio, stella, forcella, download o numero)

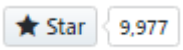
3. Inserisci il tuo nome utente GitHub nella casella ": utente" e il tuo repository nella casella ": repo"



4. Personalizza il pulsante utilizzando le caselle "Pulsante grande", "Mostra numero" e "Icona standard":



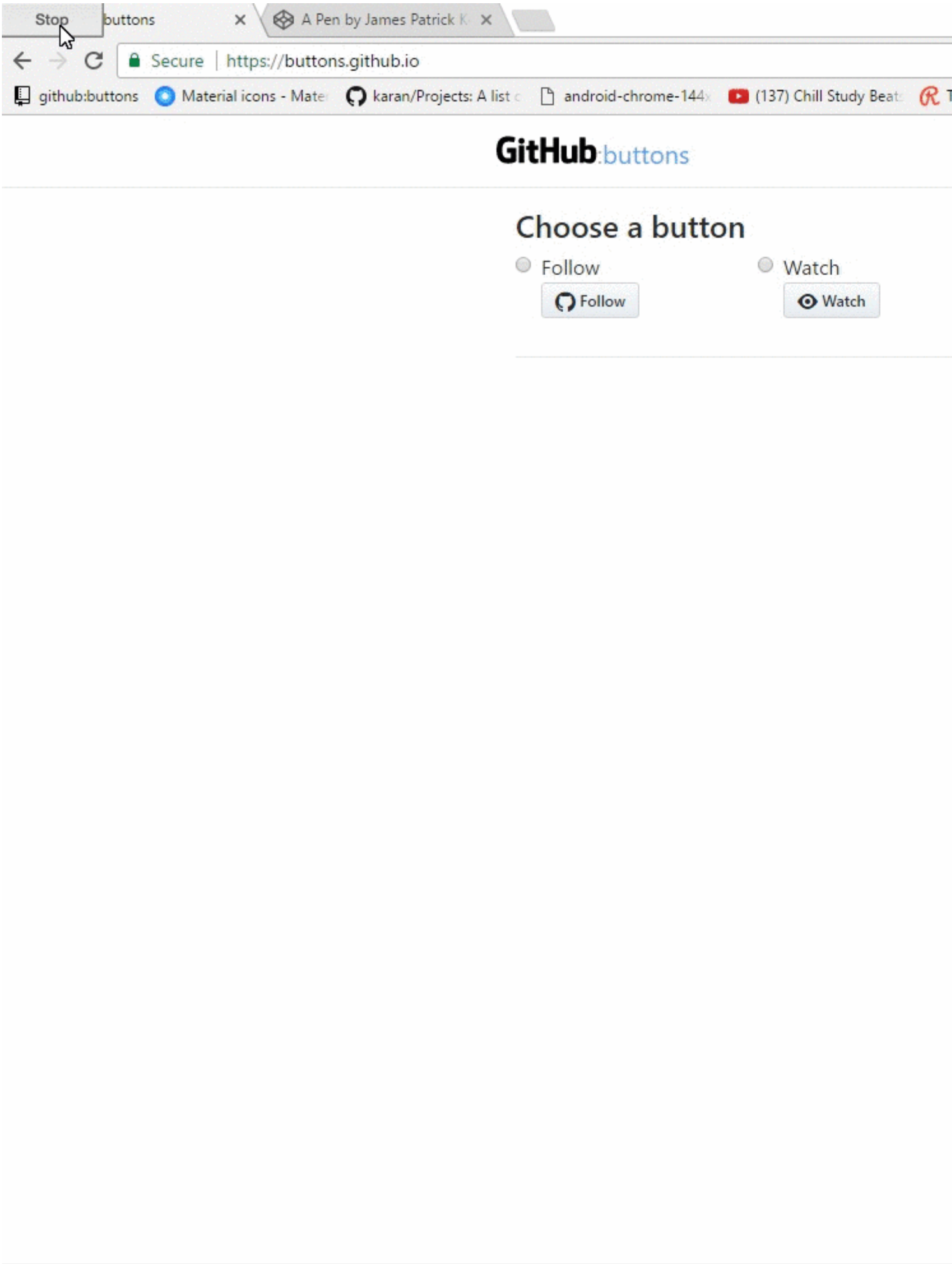




5. Inserisci questo codice in `<head>` o prima della fine del `<body>` del tuo codice:

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

6. Inserisci il codice di rendering del pulsante personalizzato nel codice.


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/it/github/topic/10585/utilizzo-dei-pulsanti-github

Capitolo 15: Visualizzazione della timeline / feed di GitHub nel tuo sito web

Examples

Visualizzazione della timeline / feed di GitHub sul tuo sito web

Questo documento spiega come visualizzare i tuoi feed / timeline GitHub sul tuo sito web.

Esempio: un esempio dal vivo è disponibile all'indirizzo:

> https://newtonjoshua.com

Timeline di GitHub:

GitHub fornisce la linea temporale pubblica per qualsiasi utente in formato Atom.

Puoi visualizzare la tua cronologia su:

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

fare riferimento a: https://developer.github.com/v3/activity/feeds

API di Google Feed:

Con l'API Feed, puoi scaricare qualsiasi feed RSS Atom, RSS o multimediale pubblico utilizzando solo JavaScript, in modo da poter condividere i feed con i tuoi contenuti e altre API con solo poche righe di JavaScript. Ciò semplifica l'integrazione rapida dei feed sul tuo sito web.

fare riferimento a: https://developers.google.com/feed/v1/devguide

Caricamento dell'API JavaScript: per iniziare a utilizzare l'API del feed, includere il seguente script nell'intestazione della pagina Web.

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

Successivamente, carica l'API del feed con google.load (modulo, versione, pacchetto).

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

Specifica dell'URL del feed: puoi chiamare google.feeds.Feed () come segue:

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

Caricamento di un feed: .load (callback) scarica il feed specificato nel costruttore dai server di

Google e chiama il callback assegnato al termine del download.

```
<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>
```

Chiamando il gestore onLoad: setOnLoadCallback (callback) è una funzione statica che registra la funzione del gestore specificata da chiamare una volta caricata la pagina contenente questa chiamata, in cui callback è una funzione richiesta chiamata quando il documento contenente viene caricato e l'API è pronta per l'uso

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

Impostazione del numero di voci del feed: .setNumEntries (num) imposta il numero di voci del feed caricate da questo feed su num. Per impostazione predefinita, la classe Feed carica quattro voci.

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

Ora puoi formattare e visualizzare i tuoi feed / timeline GitHub sul tuo sito web.

Leggi Visualizzazione della timeline / feed di GitHub nel tuo sito web online: https://riptutorial.com/it/github/topic/7479/visualizzazione-della-timeline---feed-di-github-nel-tuo-sito-web

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con github	BadAllOff, BrechtDeMan, Community, H. Pauwelyn, Hamzawey, Hugo, intboolstring, Kronos, Mateusz Piotrowski, Minhas Kamal, Nicholas Qiao, rpadovani
2	Aggiorna un repository a forcella	Derek Liu
3	Backup di GitHub	geek1011
4	Clonazione di un repository da GitHub	demonplus, geek1011, Hamzawey, James Kerrane, Mateusz Piotrowski
5	Come creare etichette GitHub personalizzate?	Ahmad Awais
6	GitHub Desktop	creyD
7	Lavorare con Gitflow	Derek Liu
8	Pagine GitHub	BrechtDeMan, geek1011, Mono
9	Problemi	geek1011, Hamzawey, SuperBiasedMan
10	Pull Requests	Maxcell
11	Rimozione di dati sensibili o file di grandi dimensioni	Gautam Krishna R, Kronos
12	scarica un singolo file dal repository GitHub	ownsourcing dev training
13	Usando Gist	Kronos, tehp
14	Utilizzo dei pulsanti GitHub	James Kerrane
15	Visualizzazione della timeline / feed di GitHub nel tuo sito web	Hugo, Newton Joshua