

Introduzione

GitHub è la principale piattaforma per l'hosting di codice: su di essa sono ospitati letteralmente milioni di progetti open source. Abbinata a Git, offre tutto ciò che serve per un flusso di lavoro produttivo ed è lo strumento preferito fra gli sviluppatori.

Partiremo con le nozioni di base sulla creazione di un repository, poi vedremo come si gestisce l'issue tracker, dove è possibile discutere dei progetti. Continuando il nostro viaggio, esploreremo come usare il wiki e scrivere una documentazione ricca che accompagni il progetto. Le tappe successive saranno la gestione di organizzazioni e team e quindi le richieste di pull, per le quali GitHub è ben noto.

Poi ci concentreremo sulla creazione di semplici pagine web ospitate in GitHub e, infine, esploreremo come configurare le impostazioni di un utente e di un repository.

A chi è rivolto questo libro

Questo libro è pensato per sviluppatori esperti o alle prime armi con una conoscenza di base di Git. Se vi siete mai ripromessi di studiare come progetti di grandi dimensioni quali Twitter, Google o lo stesso GitHub collaborino nella creazione del codice, allora questo libro fa per voi.

Struttura del libro

Il Capitolo 1, *Repository e uso dell'issue tracker*, spiega alcune delle principali funzionalità offerte da GitHub e quale uso se ne possa fare. L'issue tracker è il cuore delle comunicazioni fra gli sviluppatori e/o gli utenti di un progetto. Potete considerarlo un blocco di appunti dedicato per ciascun repository in cui potete tenere traccia di errori, report, richieste di funzionalità e qualsiasi altra cosa che possa essere messa per iscritto. GitHub ha implementato molte altre funzionalità che si basano sull'issue tracker, come etichette e *milestones*, che danno la possibilità di visualizzare e classificare meglio tutti gli issue.

Il Capitolo 2, *Usare il wiki e gestire le versioni del codice*, vi aiuterà a capire come creare, modificare e mantenere un wiki che possa essere la sede della vostra documentazione a complemento del progetto. Vedremo anche come creare una nuova release a partire da

un ramo o da un tag esistenti, accompagnata da note di release. In questo modo, l'utente finale potrà comprendere quali siano i cambiamenti rispetto alle versioni precedenti.

Il Capitolo 3, *Gestire organizzazioni e team*, mostrerà come creare e gestire le organizzazioni di cui siete proprietari. Vedrete anche come creare team, aggiungervi utenti e assegnare livelli di accesso diversi in funzione delle vostre esigenze.

Il Capitolo 4, *Collaborazione con il flusso di lavoro di GitHub*, si concentrerà su come lavorare con i rami (*branch*) e le richieste di pull, le caratteristiche più potenti di GitHub. Il Capitolo 5, *GitHub Pages e Web Analytics*, parlerà di come costruire pagine web relative al vostro progetto, ospitate esclusivamente in GitHub. È possibile creare pagine web statiche con HTML, CSS e JavaScript.

Il Capitolo 6, *Le impostazioni degli utenti e dei repository*, esplora le impostazioni essenziali e più comuni per un utente e un repository. Come utenti, sono moltissime le informazioni che potete definire nella pagina delle impostazioni dell'utente: per esempio, associare più di un indirizzo email al vostro account, aggiungere più chiavi SSH o impostare l'autenticazione a due fattori. Analogamente si possono configurare alcune funzionalità di un repository attraverso la sua pagina delle impostazioni: per esempio, si possono abilitare o disabilitare le pagine wiki, si può concedere l'accesso in scrittura al pubblico, oppure disabilitare completamente l'issue tracker.

File degli esempi e figure a colori

Per seguire questo libro avrete bisogno di Git (qualsiasi versione andrà bene) e di un account GitHub.

Potete scaricare i file di codice degli esempi e le immagini a colori di screenshot e figure dalla pagina web del volume all'indirizzo <http://bit.ly/apo-github>.

Lo stesso codice degli esempi è inoltre disponibile su GitHub all'indirizzo <http://github.com/PacktPublishing/GitHub-Essentials-Second-Edition>.

Convenzioni

Nel libro sono utilizzate alcune convenzioni.

Il carattere monospaziato evidenzia nel testo codice, nomi di tabelle di database, nomi di cartelle e di file, estensioni dei nomi di file, percorsi, URL esemplificativi, input dell'utente e handle di Twitter. Per esempio: "Montate su un altro disco del vostro sistema il file di immagine disco WebStorm-10*.dmg che avete scaricato".

Un blocco di codice è formattato nel modo seguente:

```
echo "\n## Description\n\nGitHub for dummies" >> README.md
git add README.md
git commit -m "Add second level header to README file"
git push origin add_description
```

Quando vogliamo attirare l'attenzione su una particolare parte di un blocco di codice, le righe o gli elementi pertinenti sono evidenziati in grassetto:

```
echo "\n## Description\n\nGitHub for dummies" >> README.md
git add README.md
```

```
git commit -m "Add second level header to README file"  
git push origin add_description
```

Input e output da riga di comando sono scritti nel modo seguente:

```
mkdir -p ~/github-essentials  
cd $_
```

Nel testo, il corsivo è utilizzato per evidenziare l'introduzione di un nuovo termine, frasi ed espressioni importanti ed elementi dell'interfaccia (voci di menu, etichette nelle finestre di dialogo e così via). Per esempio: "Selezionate *System info* dal pannello *Administration*".