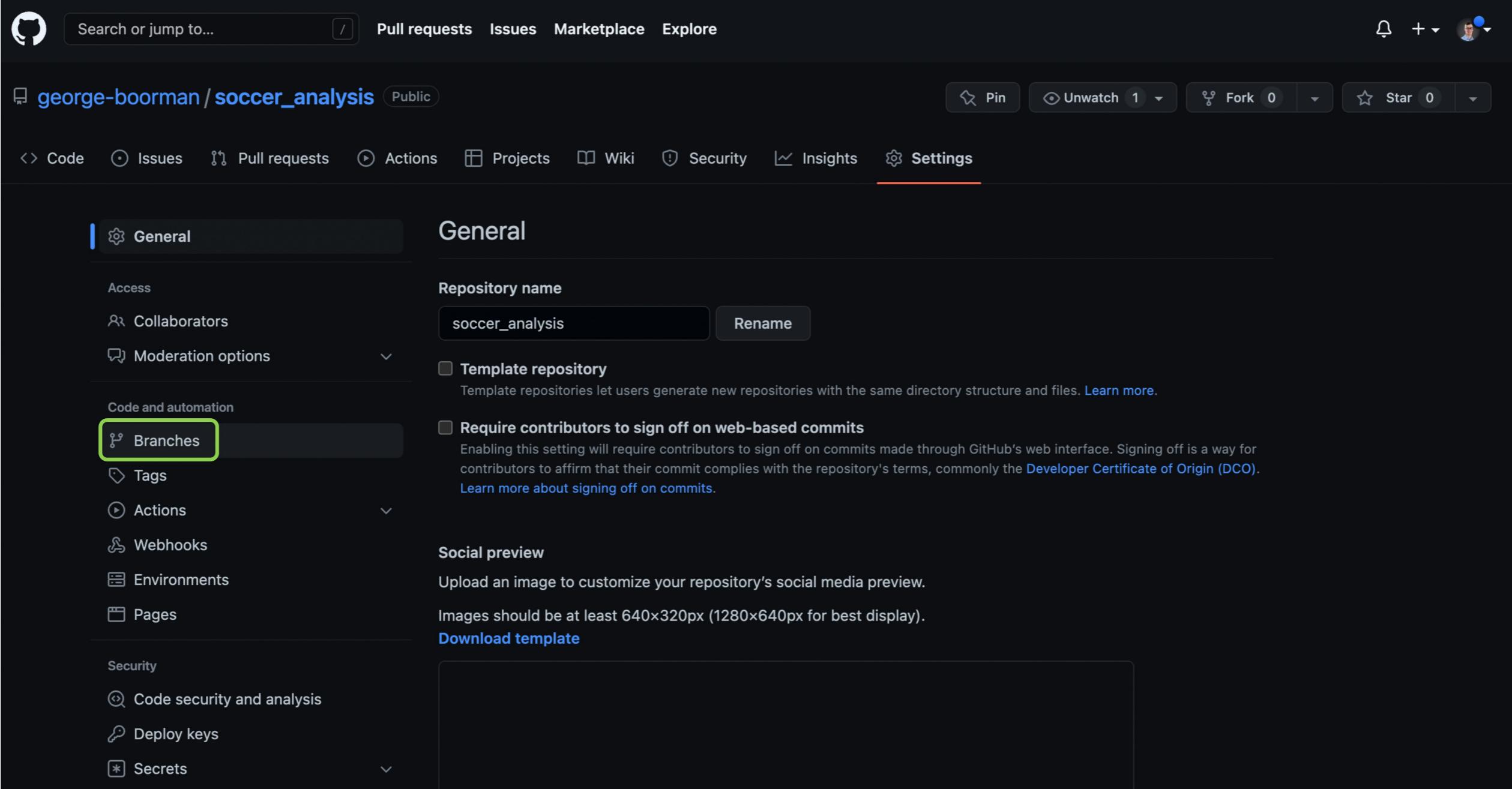
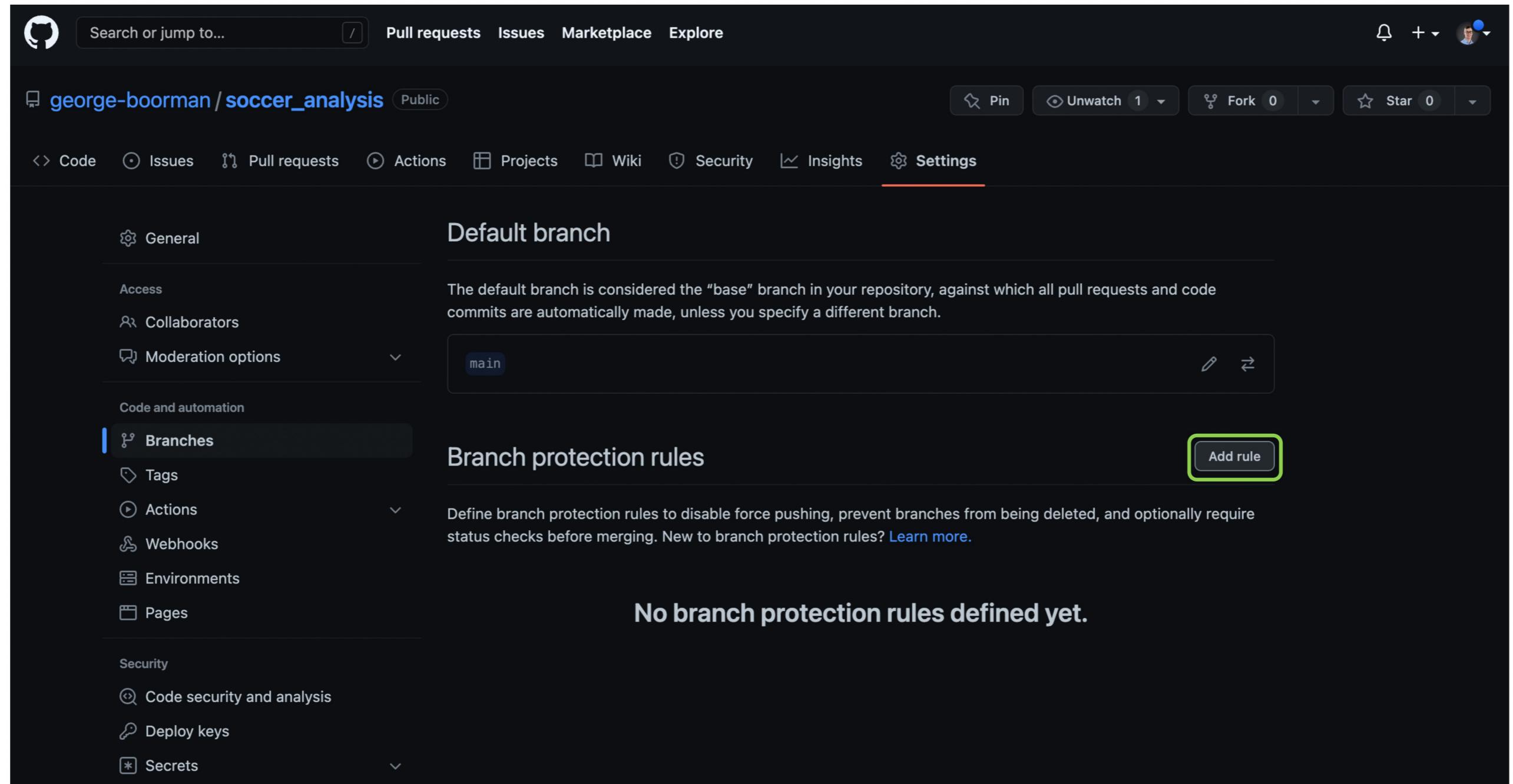


Branch protection rules



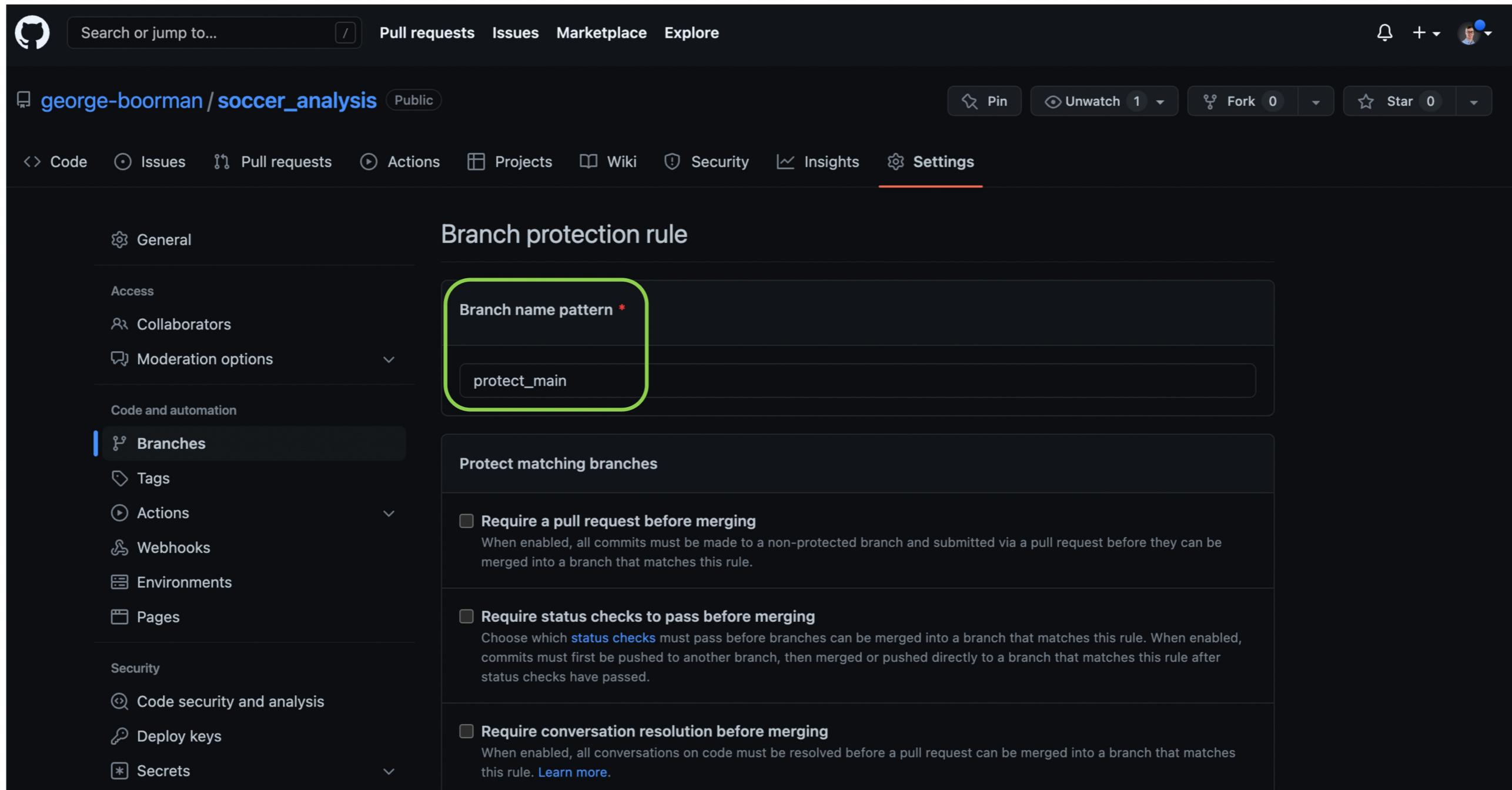
The screenshot shows the GitHub repository settings for the repository `george-boorman/soccer_analysis`. The `General` tab is selected. On the left, a sidebar lists sections: `Access`, `Collaborators`, `Moderation options`, `Code and automation` (with `Branches` highlighted), `Tags`, `Actions`, `Webhooks`, `Environments`, `Pages`, `Security` (with `Code security and analysis`, `Deploy keys`, and `Secrets` listed), and `Insights`. The main content area shows the repository name `soccer_analysis` and a `Rename` button. It also displays two settings under the `Branches` section: `Template repository` (unchecked) and `Require contributors to sign off on web-based commits` (unchecked). Below these is a `Social preview` section with a placeholder for an image and a `Download template` link.

Branch protection rules



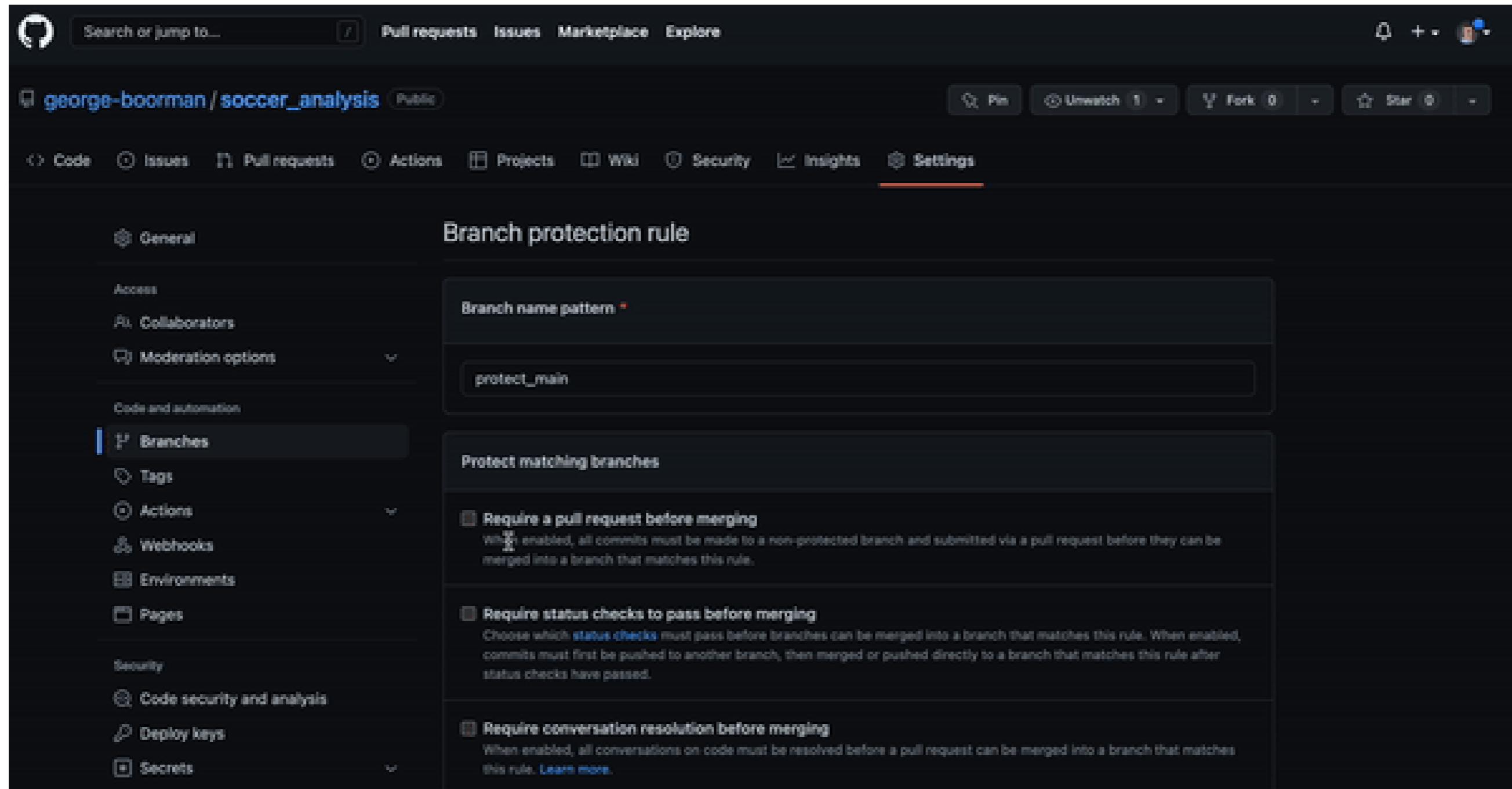
The screenshot shows the GitHub repository settings for the repository `george-boorman / soccer_analysis`. The `Settings` tab is selected. On the left, a sidebar menu is open, showing the `Branches` section highlighted with a blue bar. The main content area is titled `Default branch` and describes the `main` branch as the base branch. Below this, the `Branch protection rules` section is shown, with a green box highlighting the `Add rule` button. The text in this section explains how to define rules to disable force pushing, prevent branch deletion, and require status checks before merging. A message at the bottom states `No branch protection rules defined yet.`

Adding rules



The screenshot shows the GitHub settings page for the repository `george-boorman / soccer_analysis`. The `Settings` tab is selected. On the left, a sidebar lists various settings categories: General, Access, Collaborators, and Moderation options under the General heading; Branches, Tags, Actions, Webhooks, Environments, and Pages under Code and automation; and Security, Code security and analysis, Deploy keys, and Secrets under Security. The `Branches` section is currently active. The main content area is titled **Branch protection rule**. It shows a configuration for a branch name pattern `protect_main`, which is highlighted with a green rounded rectangle. Below this, there are three sections: **Protect matching branches**, **Require a pull request before merging**, and **Require status checks to pass before merging**. The **Require a pull request before merging** section is expanded, showing a description: "When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule." The **Require status checks to pass before merging** section is also expanded, showing a description: "Choose which [status checks](#) must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed." The **Require conversation resolution before merging** section is collapsed.

Require a pull request



The screenshot shows the GitHub settings page for the repository `george-boorman/soccer_analysis`. The `Settings` tab is selected. On the left, the `Branches` section is highlighted in the sidebar. The main content area is titled `Branch protection rule`. It shows a branch name pattern `protect_main` and three protection rules: `Require a pull request before merging`, `Require status checks to pass before merging`, and `Require conversation resolution before merging`. Each rule has a detailed description below it.

Branch name pattern: `protect_main`

Protect matching branches

Require a pull request before merging
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

Require status checks to pass before merging
Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

Require conversation resolution before merging
When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that matches this rule. [Learn more](#).

Saving the rule

Rules applied to everyone including administrators

Allow force pushes

Permit force pushes for all users with push access.

Allow deletions

Allow users with push access to delete matching branches.

Create

Let's practice!

GITHUB CONCEPTS

Repo access

GITHUB CONCEPTS

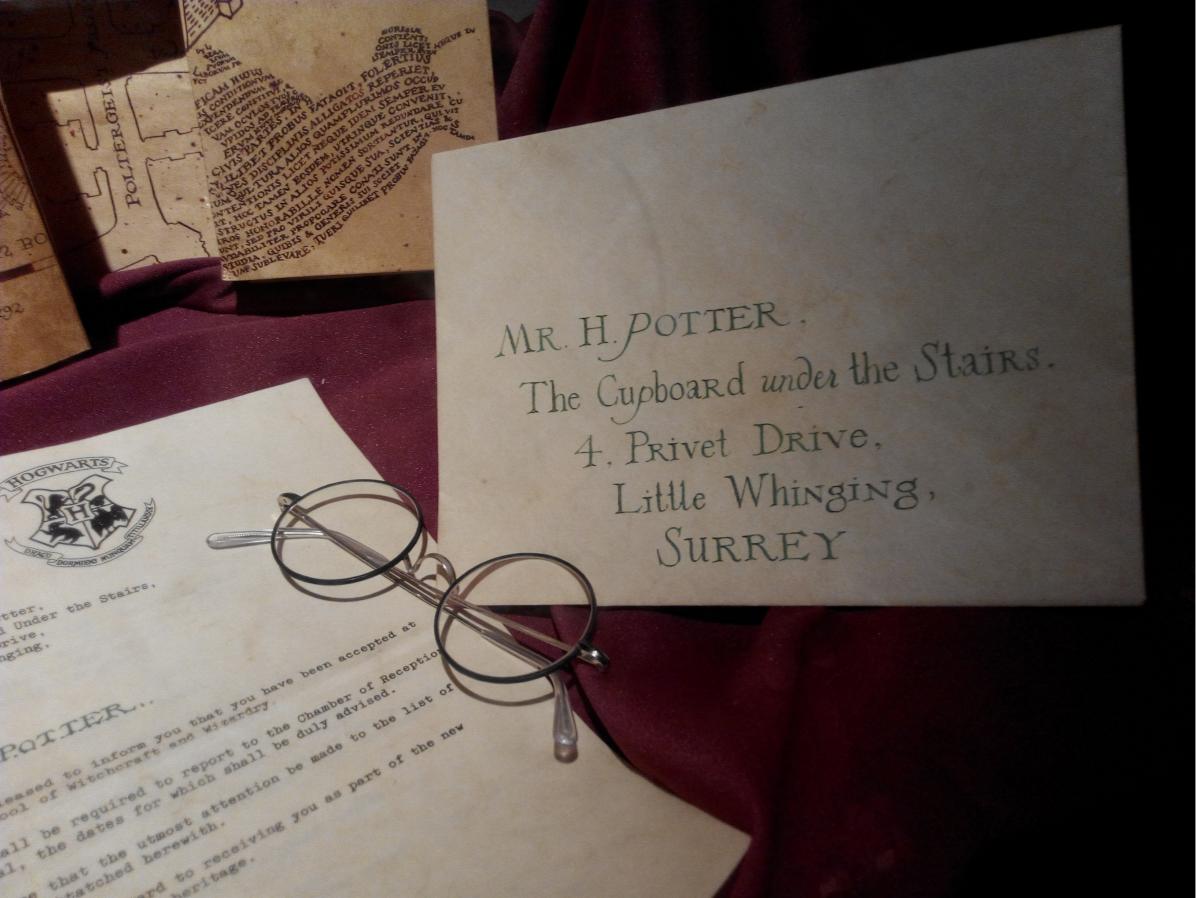


George Boorman

Curriculum Manager, DataCamp

Why restrict access?

- Datasets containing personally identifiable information
- Developing or using commercial products

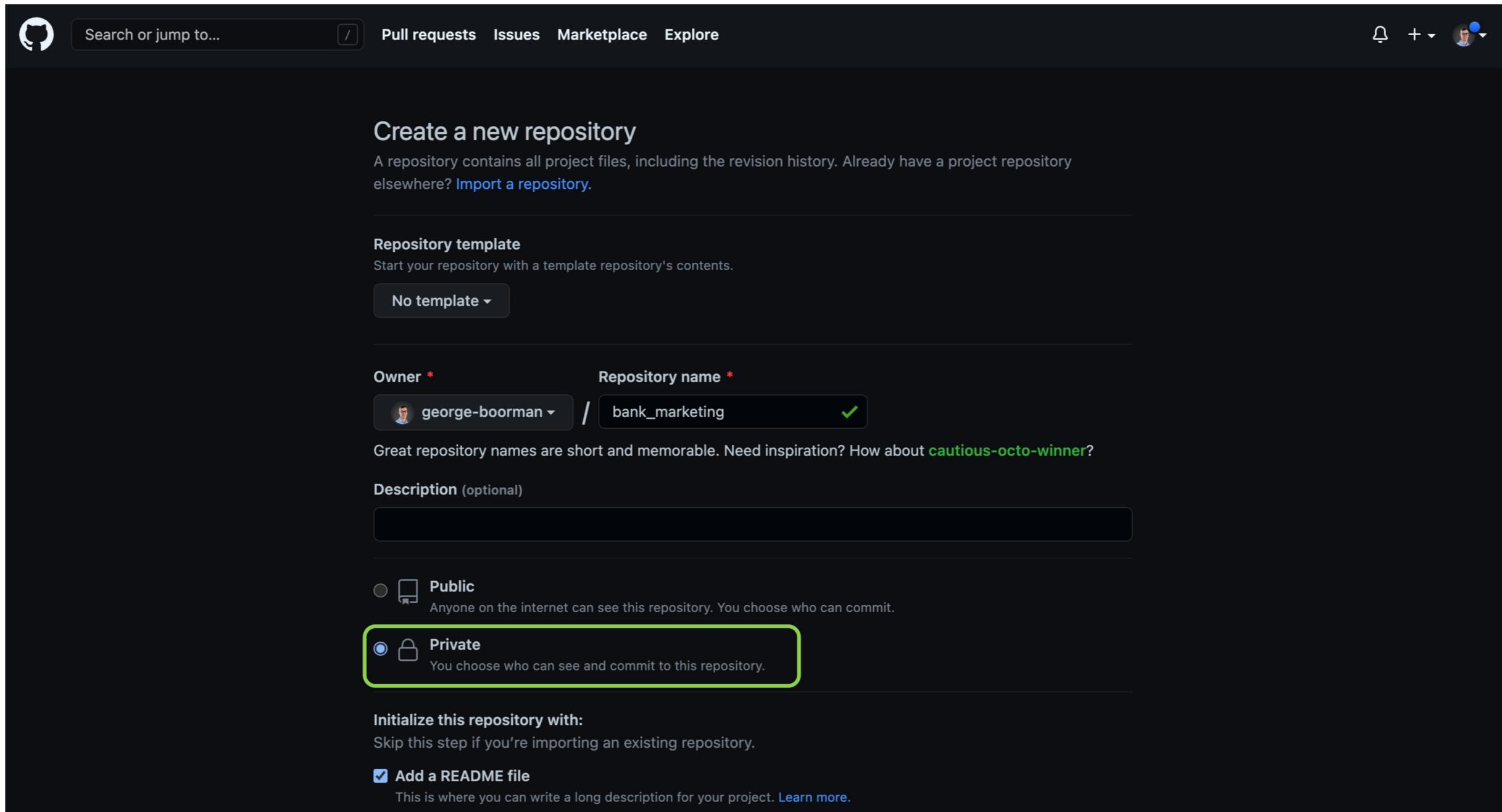


¹ Image credits: https://unsplash.com/@rae_1991

Bank marketing campaign

Column name	Description
name	Full name
address	Home address
age	Age in years
job	Job type
marital	Marital status
education	Level of education
balance	Bank balance (\$)
housing	Has a mortgage
deposit	Made a deposit as a result of the campaign

Creating a private repo



The screenshot shows the GitHub interface for creating a new repository. At the top, there is a navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. On the far right, there are icons for notifications, a plus sign, and a user profile. The main title is 'Create a new repository'. Below it, a sub-instruction says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#)'. The 'Repository template' section has a 'No template' button. The 'Owner' field is set to 'george-boorman'. The 'Repository name' field is 'bank_marketing' with a green checkmark. Below these fields, a note says 'Great repository names are short and memorable. Need inspiration? How about [cautious-octo-winner](#)?'. The 'Description (optional)' field is empty. The 'Visibility' section shows 'Public' (unchecked) and 'Private' (checked, highlighted with a green border). A note under 'Public' says 'Anyone on the internet can see this repository. You choose who can commit.' A note under 'Private' says 'You choose who can see and commit to this repository.' The 'Initialize this repository with:' section includes a note 'Skip this step if you're importing an existing repository.' and a checked checkbox for 'Add a README file'. A note under 'Add a README file' says 'This is where you can write a long description for your project. [Learn more](#)'.

Is it private?

The screenshot shows a GitHub repository page for the user 'george-boorman' with the repository name 'bank_marketing'. The repository is marked as 'Private'. The page includes a navigation bar with links to 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, there are buttons for 'Unwatch', 'Fork', and 'Star'. The main content area shows a 'Code' tab selected, with a 'main' branch dropdown showing '1 branch' and '0 tags'. A commit history is displayed, with the first commit being 'Initial commit' by 'george-boorman' 16 days ago. The repository's 'About' section indicates 'No description, website, or topics provided'. It also shows '0 stars', '1 watching', and '0 forks'. The 'Releases' section shows 'No releases published' and a link to 'Create a new release'. The 'Packages' section shows 'No packages published' and a link to 'Publish your first package'.

george-boorman / bank_marketing Private

Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags

george-boorman Initial commit a675666 16 days ago 1 commit

README.md Initial commit 16 days ago

README.md

bank_marketing

About

No description, website, or topics provided.

Readme 0 stars 1 watching 0 forks

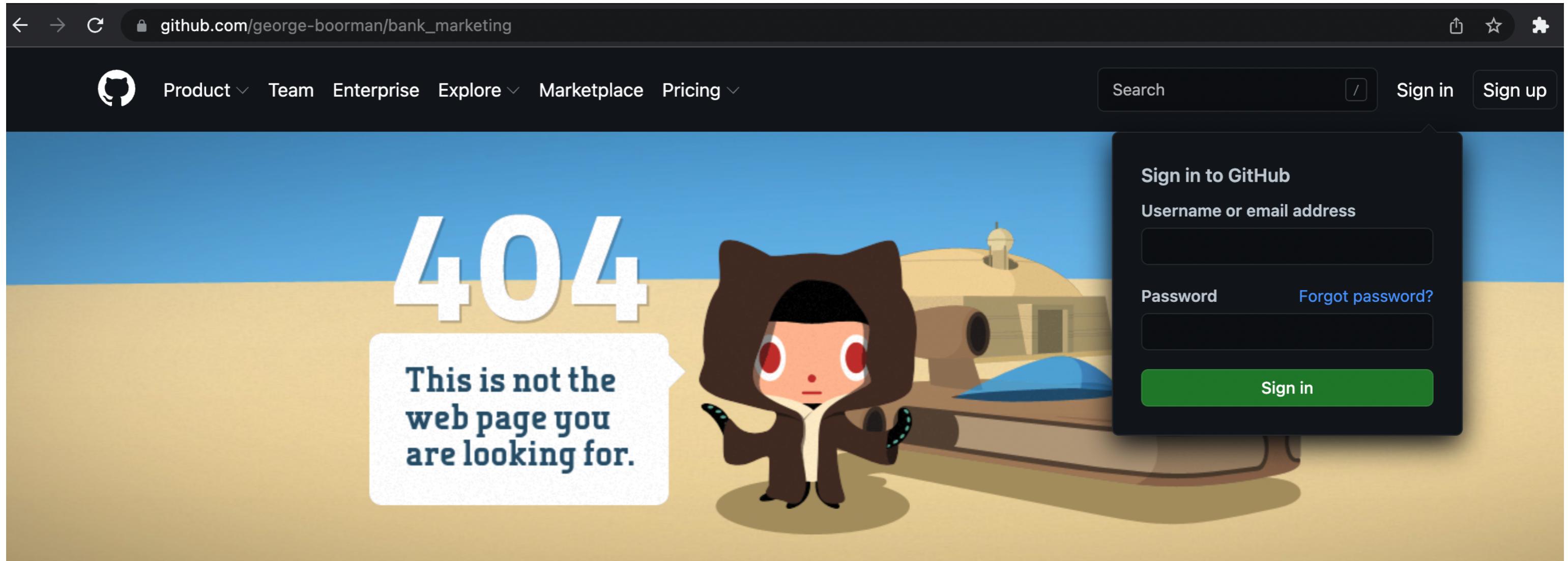
Releases

No releases published Create a new release

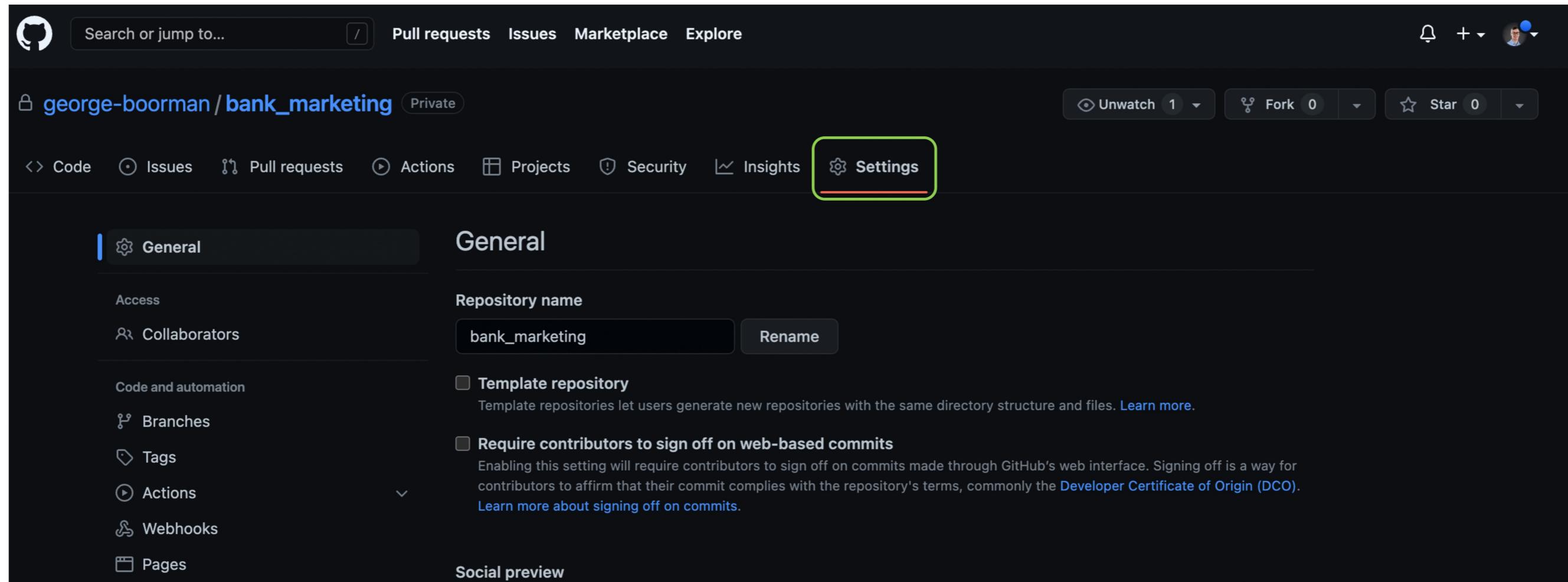
Packages

No packages published Publish your first package

Is it private?

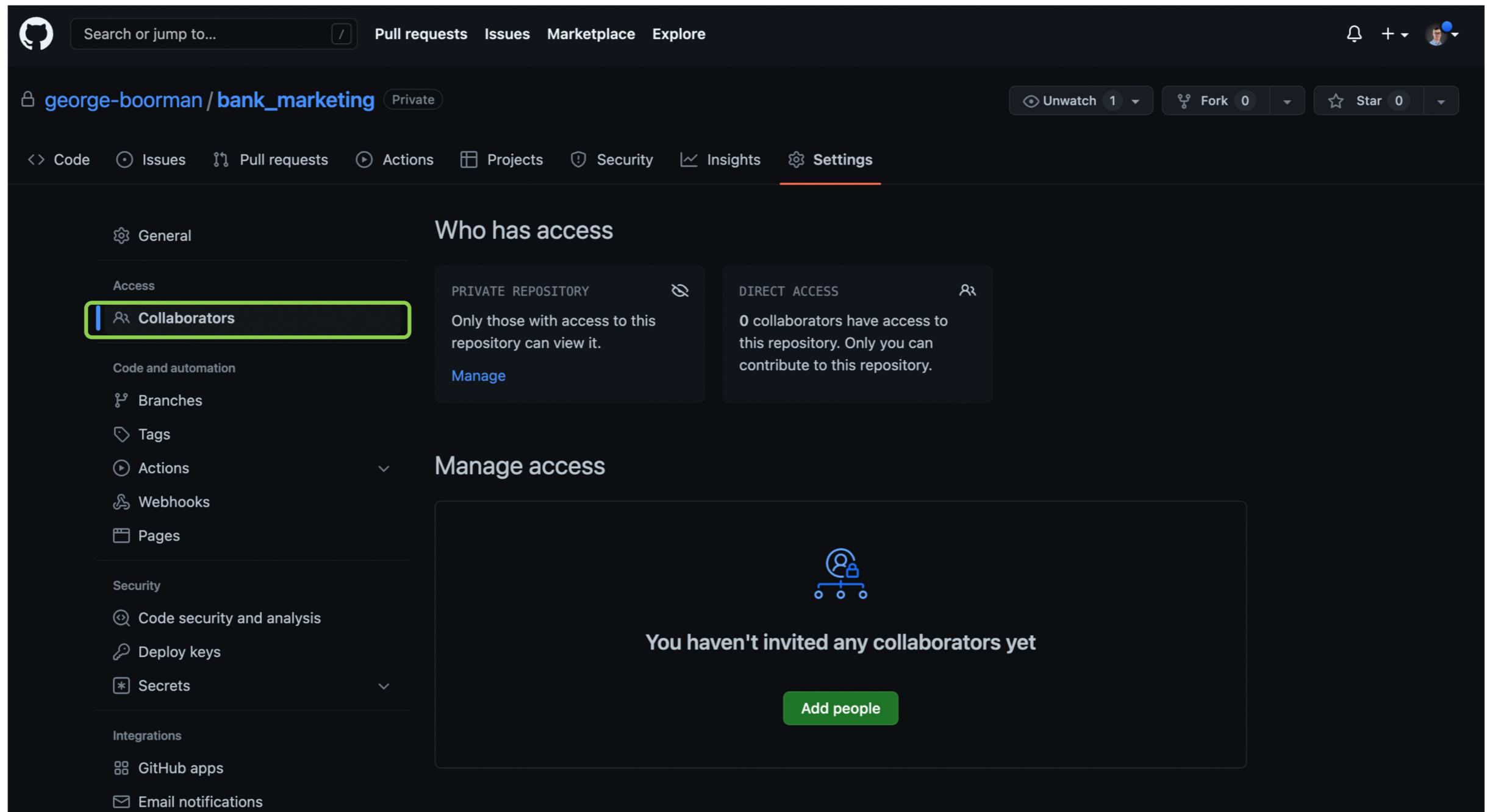


Collaboration in a private repo



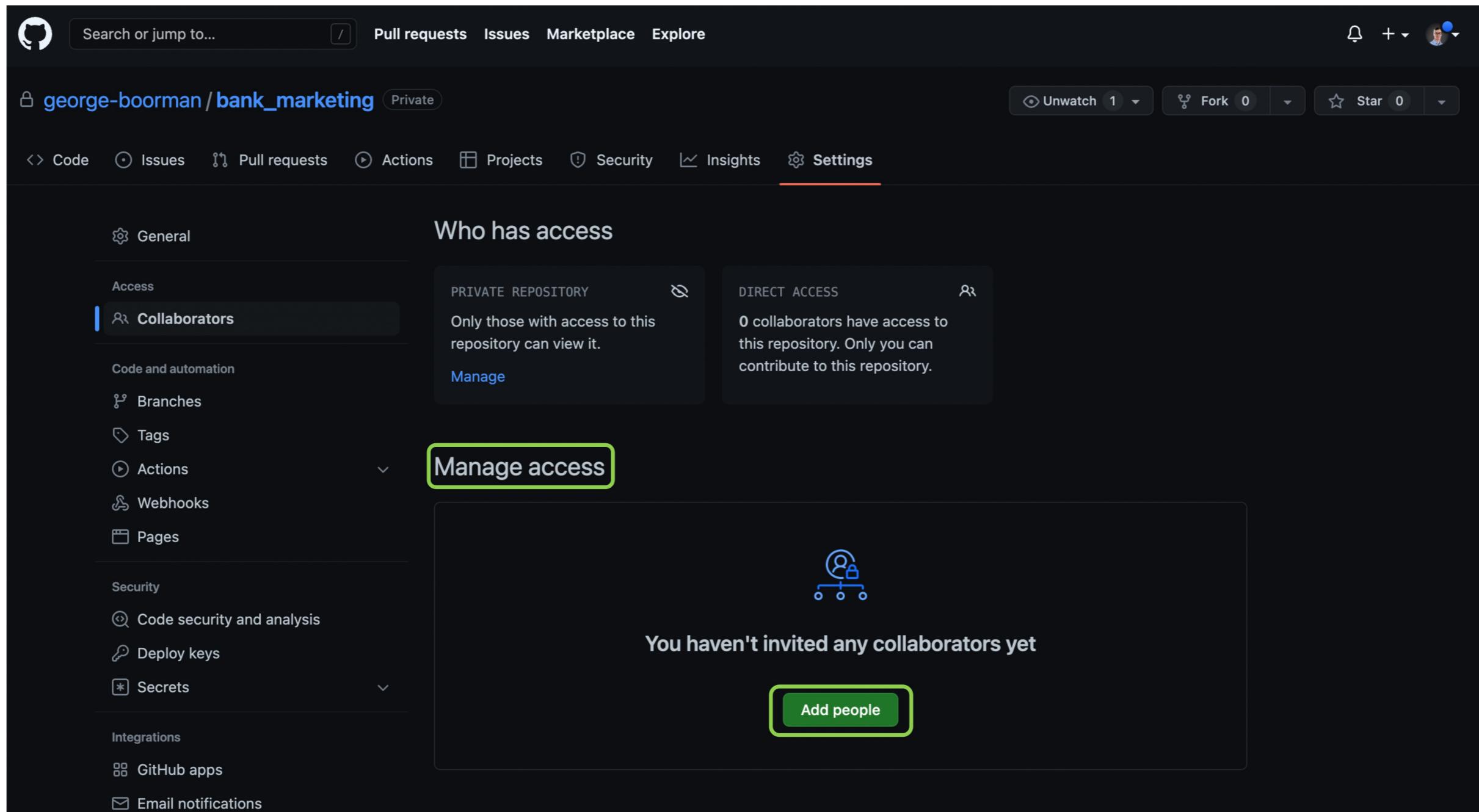
The screenshot shows a GitHub repository page for `george-boorman/bank_marketing`, which is marked as `Private`. The top navigation bar includes links for `Pull requests`, `Issues`, `Marketplace`, and `Explore`. On the far right, there are buttons for `Unwatch` (1), `Fork` (0), and `Star` (0). The main navigation bar below the repository name includes links for `Code`, `Issues`, `Pull requests`, `Actions`, `Projects`, `Security`, `Insights`, and `Settings`, with `Settings` highlighted with a green border. The left sidebar contains sections for `General`, `Access`, `Collaborators`, `Code and automation`, `Branches`, `Tags`, `Actions`, `Webhooks`, and `Pages`. The `General` section is expanded, showing the `Repository name` set to `bank_marketing` with a `Rename` button. It also includes two settings: `Template repository` (unchecked) and `Require contributors to sign off on web-based commits` (unchecked). The `Social preview` section is also visible.

Collaborators



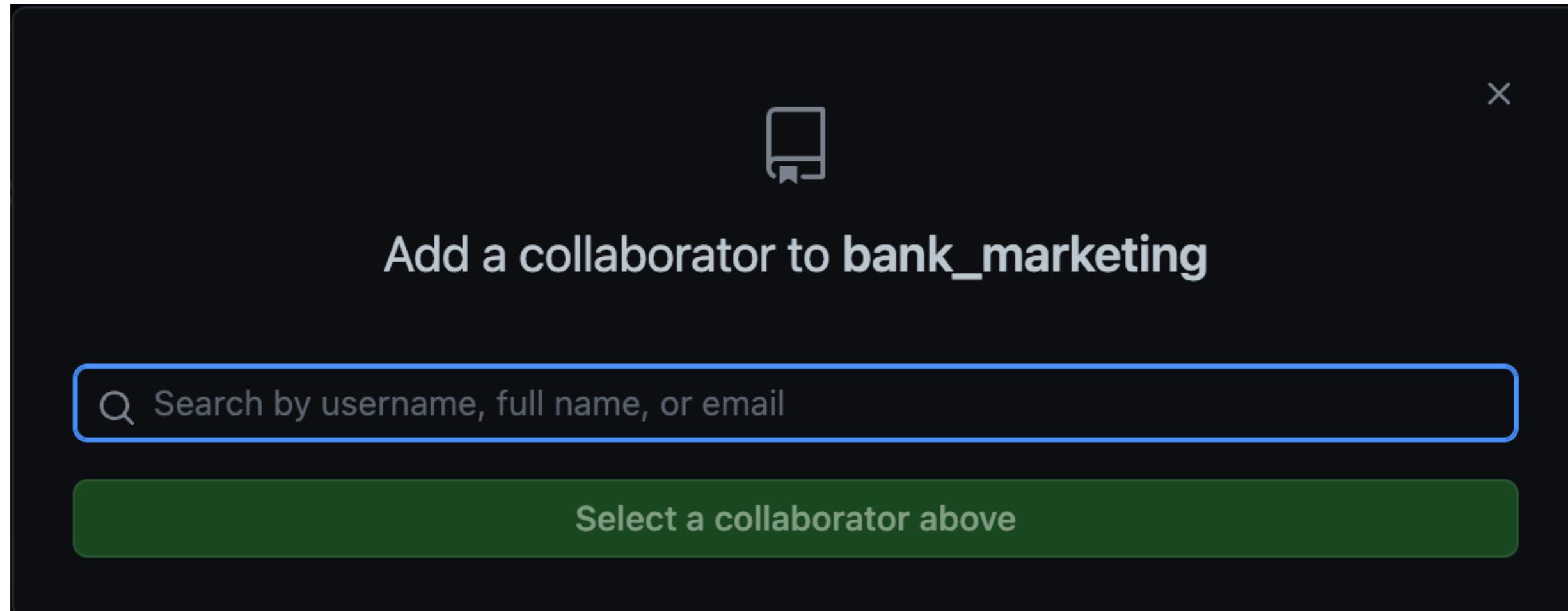
The screenshot shows the GitHub repository settings page for `george-boorman/bank_marketing`. The repository is private, as indicated by the 'Private' badge. The 'Settings' tab is selected. On the left, a sidebar lists various settings categories: General, Access (Collaborators, highlighted with a green box), Code and automation, Branches, Tags, Actions, Webhooks, Pages, Security, Code security and analysis, Deploy keys, Secrets, Integrations, GitHub apps, and Email notifications. The 'Access' section is expanded, showing 'PRIVATE REPOSITORY' and 'DIRECT ACCESS'. Under 'PRIVATE REPOSITORY', it says 'Only those with access to this repository can view it.' and has a 'Manage' button. Under 'DIRECT ACCESS', it says '0 collaborators have access to this repository. Only you can contribute to this repository.' The main content area is titled 'Who has access' and 'Manage access', both of which are currently empty, with a message 'You haven't invited any collaborators yet' and a 'Add people' button.

Collaborators

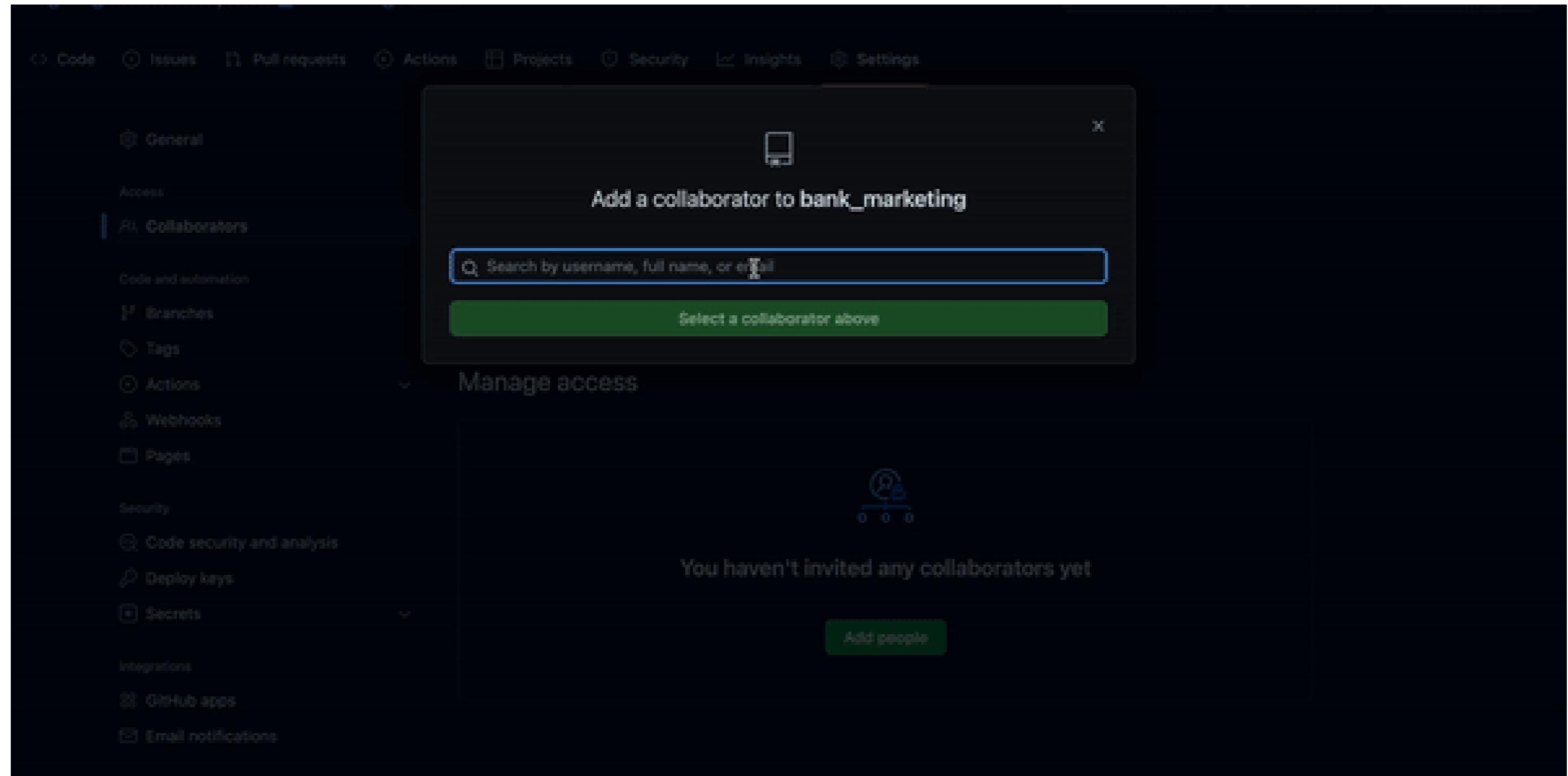


The screenshot shows the GitHub repository settings page for 'george-boorman/bank_marketing'. The 'Settings' tab is selected. On the left, a sidebar lists 'General', 'Access' (with 'Collaborators' selected), 'Code and automation', 'Branches', 'Tags', 'Actions', 'Webhooks', 'Pages', 'Security', 'Code security and analysis', 'Deploy keys', 'Secrets', 'Integrations', 'GitHub apps', and 'Email notifications'. The main content area is titled 'Who has access' and shows 'PRIVATE REPOSITORY' (Only those with access to this repository can view it) and 'DIRECT ACCESS' (0 collaborators have access to this repository. Only you can contribute to this repository). A 'Manage' button is located under the PRIVATE REPOSITORY section. A large 'Manage access' button is centered. Below it, a icon of a person with a lock is shown with the text 'You haven't invited any collaborators yet' and a 'Add people' button.

Adding a collaborator



Adding a collaborator



The screenshot shows the GitHub interface for adding a collaborator to a repository named 'bank_marketing'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The 'Settings' tab is active. On the left, a sidebar lists General, Access, Collaborators (which is selected and highlighted in blue), Code and automation, Branches, Tags, Actions, Webhooks, Pages, Security, Code security and analysis, Deploy keys, and Secrets. Below the sidebar, the 'Manage access' section displays a message: 'You haven't invited any collaborators yet'. A green 'Add people' button is visible. A modal window titled 'Add a collaborator to bank_marketing' is open in the center. It features a search bar with the placeholder 'Search by username, full name, or email' and a green button below it that says 'Select a collaborator above'. The modal has a close button in the top right corner.

Let's practice!

GITHUB CONCEPTS

Personal Access Tokens (PAT)

GITHUB CONCEPTS



George Boorman

Curriculum Manager, DataCamp

Authentication

- Can use Git in the terminal to interact with GitHub

```
git clone https://github.com/george-boorman/bank_marketing
```

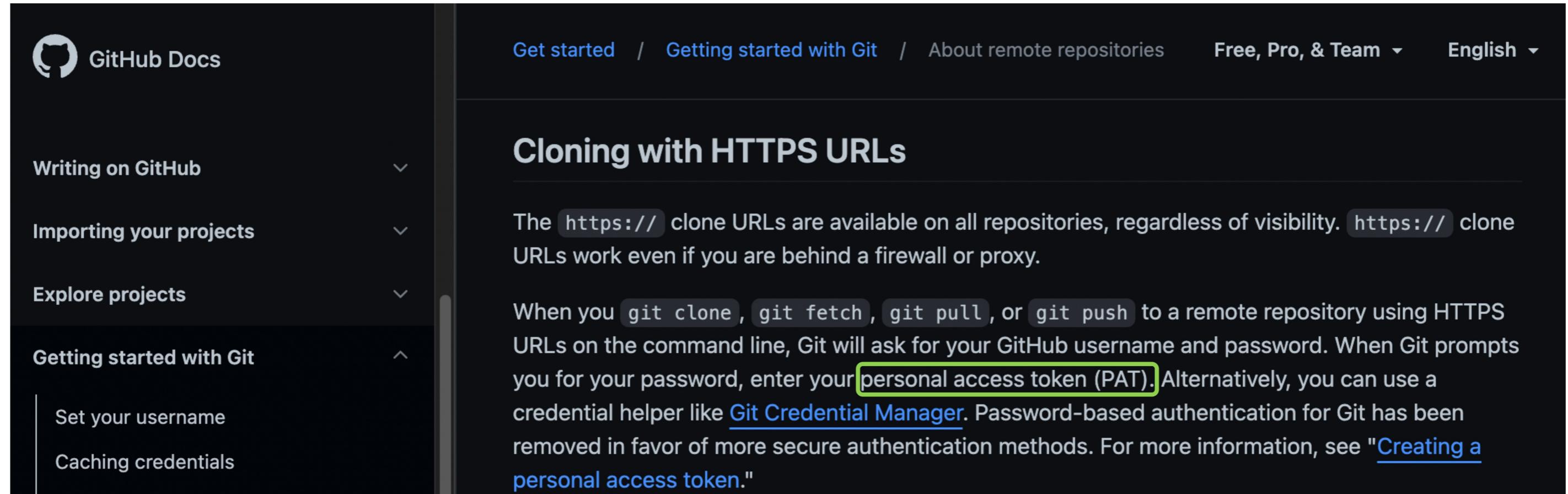
Username for 'https://github.com': george-boorman

Password for 'https://george-boorman@github.com':

Authentication failed

```
remote: Support for password authentication was removed on August 13, 2021.  
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.  
fatal: Authentication failed for 'https://github.com/george-boorman/bank_marketing/'
```

GitHub Docs



The screenshot shows the GitHub Docs interface. On the left, there is a sidebar with a navigation menu. The menu items are: Writing on GitHub, Importing your projects, Explore projects, Getting started with Git (which is currently selected), Set your username, and Caching credentials. The main content area has a title 'Cloning with HTTPS URLs'. Below the title, there is text explaining that HTTPS clone URLs are available on all repositories, regardless of visibility. It also describes the process of using Git commands like 'git clone', 'git fetch', 'git pull', and 'git push' and how to handle password prompts by using a personal access token (PAT) or a credential helper. The text is as follows:

The `https://` clone URLs are available on all repositories, regardless of visibility. `https://` clone URLs work even if you are behind a firewall or proxy.

When you `git clone`, `git fetch`, `git pull`, or `git push` to a remote repository using HTTPS URLs on the command line, Git will ask for your GitHub username and password. When Git prompts you for your password, enter your [personal access token \(PAT\)](#). Alternatively, you can use a credential helper like [Git Credential Manager](#). Password-based authentication for Git has been removed in favor of more secure authentication methods. For more information, see "[Creating a personal access token](#)."

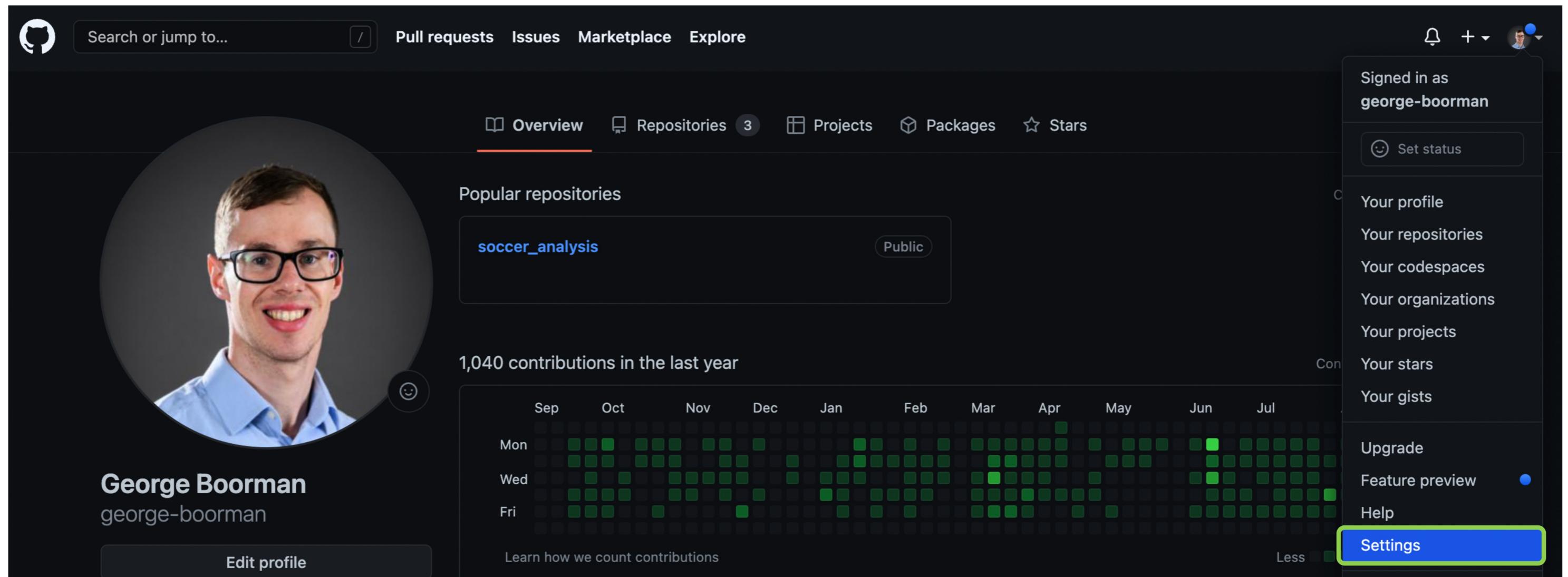
¹ <https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls>

What is a Personal Access Token?

- Personal Access Token (PAT)
 - An alternative to using passwords for authentication in the terminal
 - Required since August 2021 instead of passwords
 - More secure
- Not needed in GitHub
 - Only to access a remote repo using the terminal

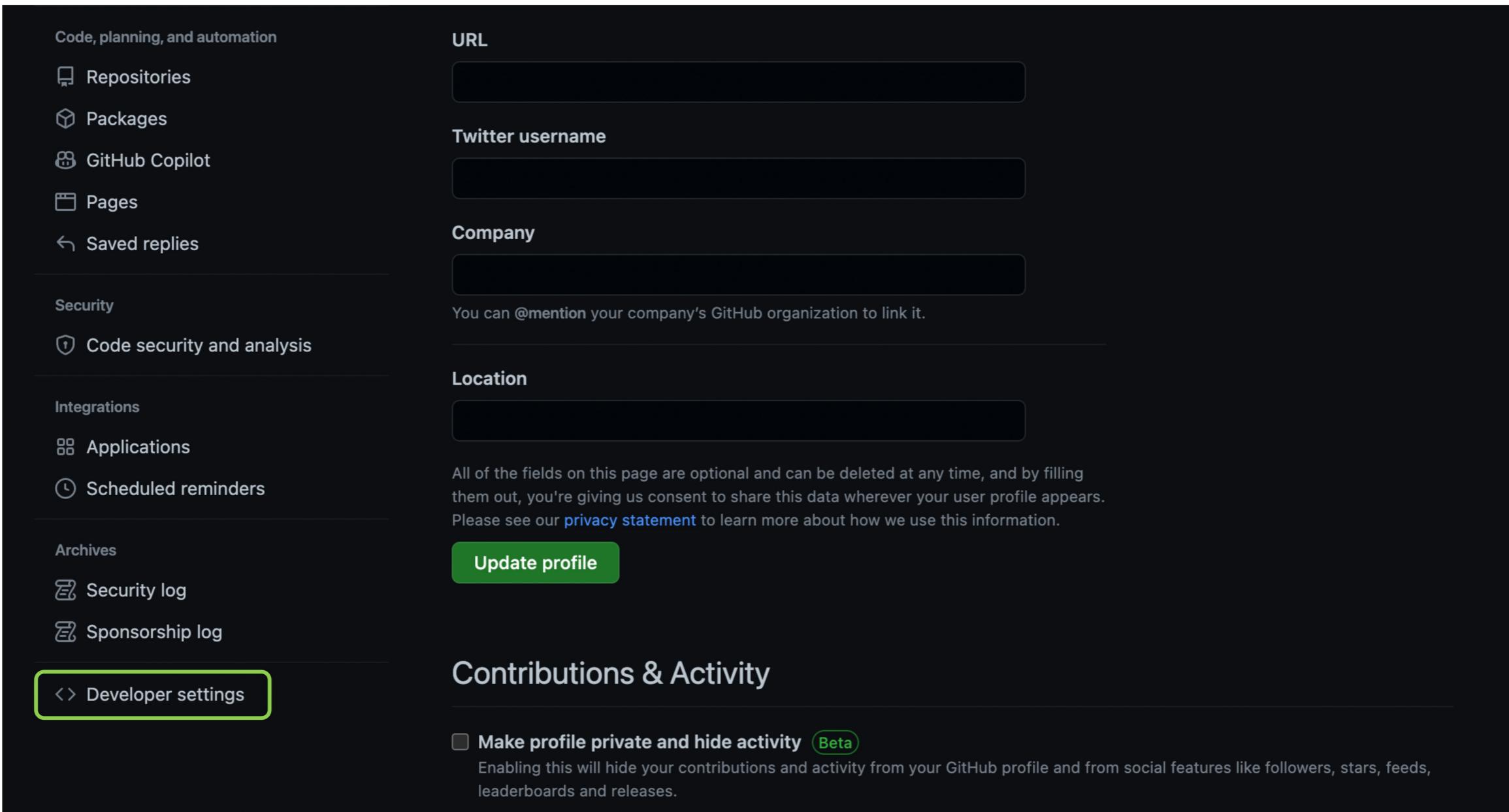
¹ <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>

Creating a PAT



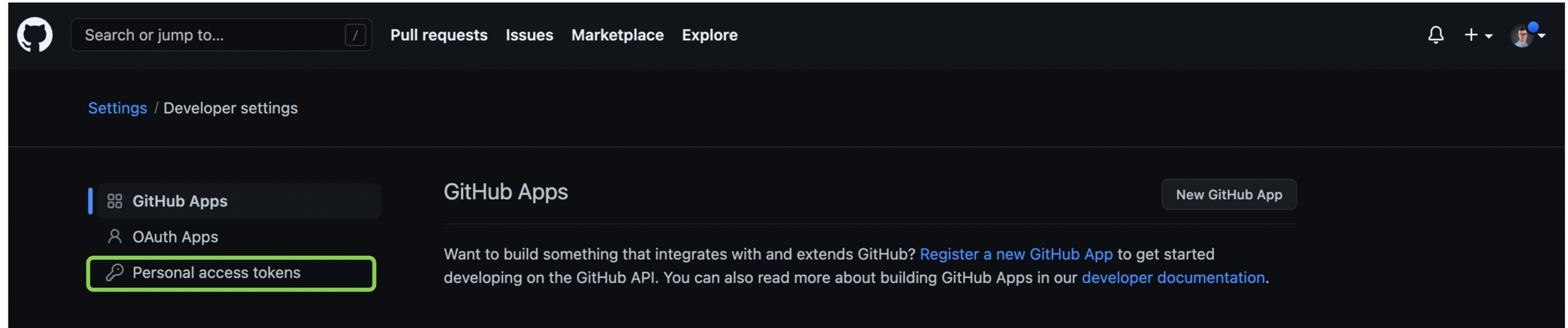
The screenshot shows a GitHub profile for a user named George Boorman. The profile picture is a circular portrait of a smiling man with glasses. The user's name, "George Boorman", and GitHub handle, "george-boorman", are displayed below the picture. A "Edit profile" button is visible. The "Overview" tab is selected, showing a "Popular repositories" section with a card for "soccer_analysis" (Public). Below this, a chart displays "1,040 contributions in the last year" from September to July, with a "Less" button at the bottom right. The top navigation bar includes "Pull requests", "Issues", "Marketplace", and "Explore". The right sidebar, signed in as "george-boorman", lists options like "Your profile", "Your repositories", and "Settings", with "Settings" highlighted with a green border. A "Feature preview" badge with a blue dot is also present.

Creating a PAT



The screenshot shows the GitHub Profile Settings page. On the left, a sidebar lists various profile sections: **Code, planning, and automation** (Repositories, Packages, GitHub Copilot, Pages, Saved replies); **Security** (Code security and analysis); **Integrations** (Applications, Scheduled reminders); and **Archives** (Security log, Sponsorship log). At the bottom of the sidebar is the **<> Developer settings** button, which is highlighted with a green border. The main content area contains optional profile fields: **URL**, **Twitter username**, **Company** (with a note: "You can @mention your company's GitHub organization to link it."), and **Location**. Below these fields is a note: "All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information." At the bottom of this section is a green **Update profile** button. The page also features a **Contributions & Activity** section with a checkbox for "Make profile private and hide activity" (Beta), which includes a note: "Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds, leaderboards and releases."

Creating a PAT



The screenshot shows the GitHub Developer settings page. The navigation bar at the top includes the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. The user's profile picture is in the top right. The main navigation on the left has three items: GitHub Apps, OAuth Apps, and Personal access tokens, with Personal access tokens being the active tab and highlighted with a green border. The central content area is titled "GitHub Apps" and contains a sub-section titled "Personal access tokens". It includes a "New GitHub App" button and a descriptive text about building GitHub Apps. The overall theme is dark.

Search or jump to... /

Pull requests Issues Marketplace Explore

Settings / Developer settings

GitHub Apps

New GitHub App

OAuth Apps

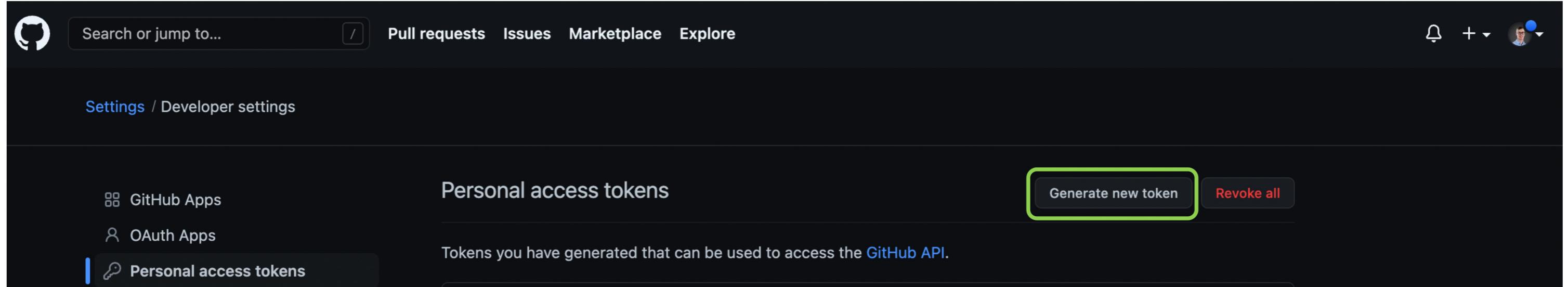
Personal access tokens

GitHub Apps

New GitHub App

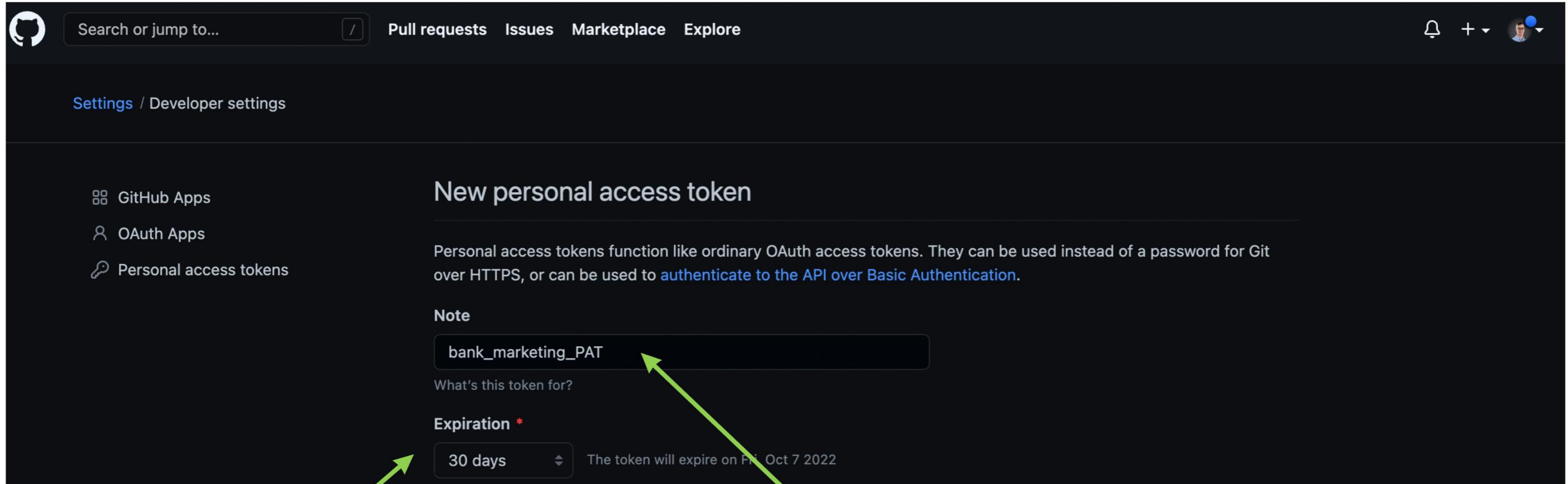
Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

Creating a PAT



The screenshot shows the GitHub Developer settings page. The navigation bar at the top includes the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. The user's profile icon is in the top right. The main navigation on the left lists GitHub Apps, OAuth Apps, and Personal access tokens, with Personal access tokens being the active tab. The central content area is titled "Personal access tokens" and contains a sub-instruction: "Tokens you have generated that can be used to access the GitHub API." Below this are two buttons: "Generate new token" (highlighted with a green box) and "Revoke all".

Creating a PAT



Search or jump to... / Pull requests Issues Marketplace Explore

Settings / Developer settings

GitHub Apps OAuth Apps Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API](#) over Basic Authentication.

Note

bank_marketing_PAT

What's this token for?

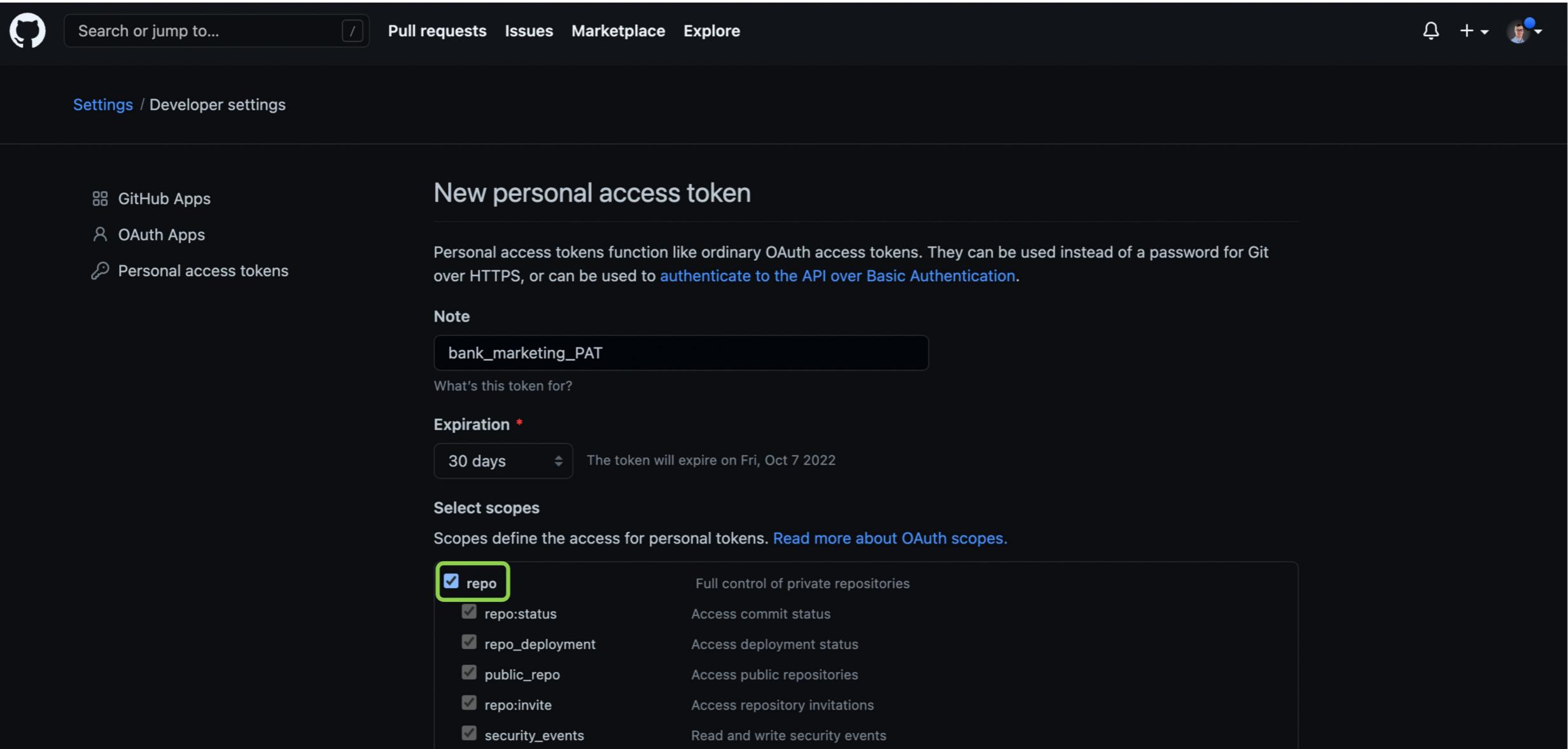
Expiration *

30 days The token will expire on Fri, Oct 7 2022

Expiration

Note

Creating a PAT



The screenshot shows the GitHub Developer settings page with a dark theme. The left sidebar includes links for GitHub Apps, OAuth Apps, and Personal access tokens. The main content area is titled "New personal access token" and explains that these tokens function like OAuth access tokens, can be used for Git over HTTPS, or for API authentication. A note field contains the text "bank_marketing_PAT". The "Expiration" dropdown is set to "30 days", with a note that the token will expire on Fri, Oct 7 2022. The "Select scopes" section lists several options, with "repo" checked and highlighted with a green border. Other checked scopes include "repo:status", "repo_deployment", "public_repo", "repo:invite", and "security_events".

Search or jump to... / Pull requests Issues Marketplace Explore

Settings / Developer settings

GitHub Apps OAuth Apps Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

bank_marketing_PAT

What's this token for?

Expiration *

30 days The token will expire on Fri, Oct 7 2022

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

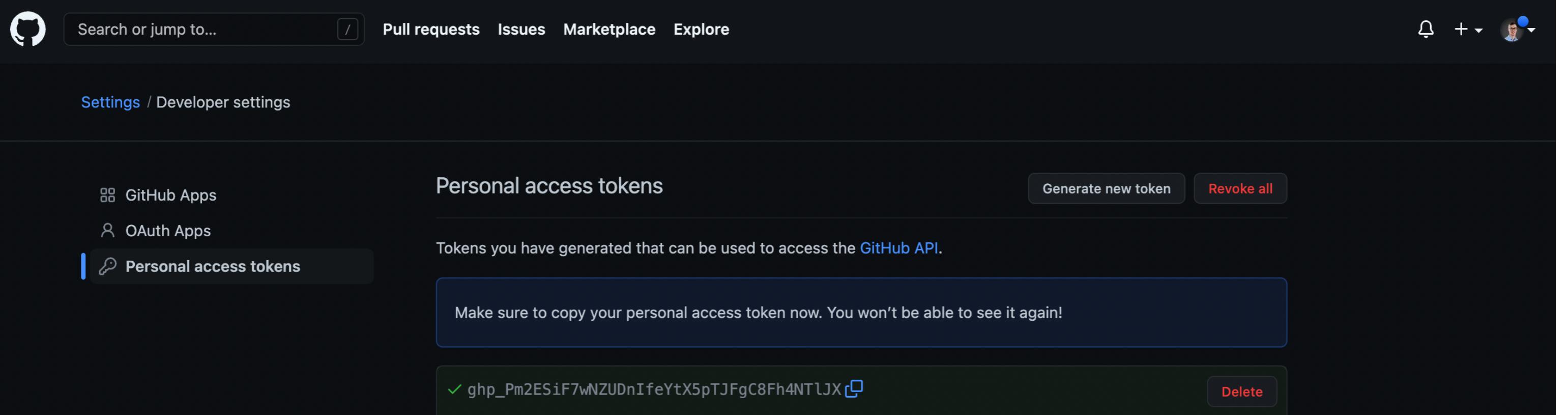
<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

Creating a PAT

<input type="checkbox"/> write:discussion	Read and write team discussions
<input type="checkbox"/> read:discussion	Read team discussions
<input type="checkbox"/> admin:enterprise	Full control of enterprises
<input type="checkbox"/> manage_runners:enterprise	Manage enterprise runners and runner groups
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> project	Full control of projects
<input type="checkbox"/> read:project	Read access of projects
<input type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input type="checkbox"/> write:gpg_key	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys
<input type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

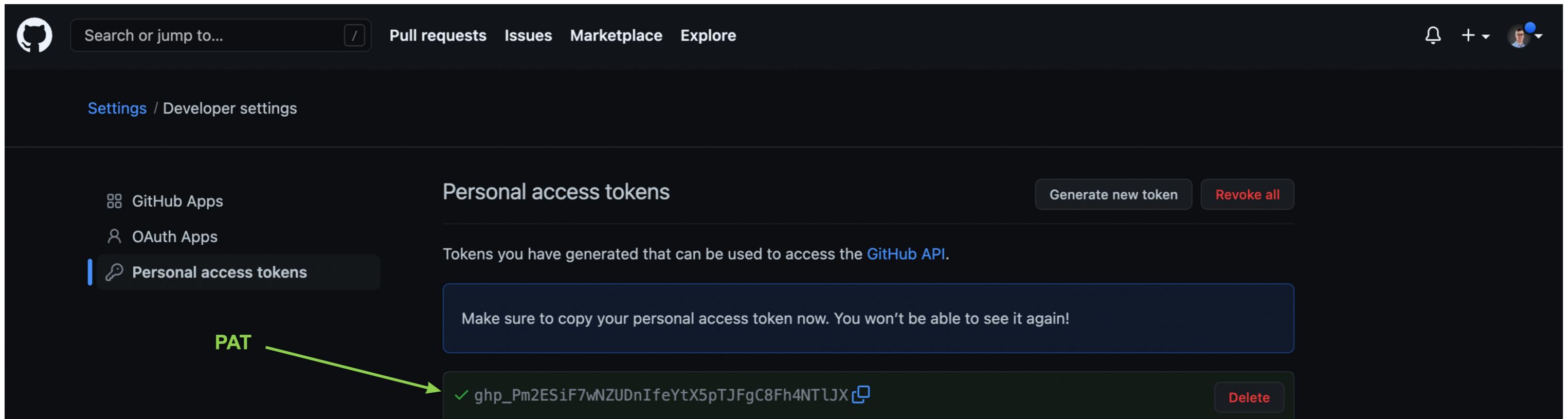
[Generate token](#) [Cancel](#)

Using the PAT



The screenshot shows the GitHub Developer settings page with the 'Personal access tokens' section selected. The page includes a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. The 'Personal access tokens' section has a 'Generate new token' button and a 'Revoke all' button. A note says 'Tokens you have generated that can be used to access the GitHub API.' A message box提醒 'Make sure to copy your personal access token now. You won't be able to see it again!' A token entry is shown with a green checkmark, the text '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', and a copy icon. A 'Delete' button is also present.

Using the PAT



A screenshot of the GitHub 'Personal access tokens' page. The page is located at [Settings / Developer settings](#). The 'Personal access tokens' tab is selected, indicated by a blue background and white text. Other tabs include 'GitHub Apps' and 'OAuth Apps'. At the top right are buttons for 'Generate new token' and 'Revoke all'. A prominent message in a blue box says: 'Make sure to copy your personal access token now. You won't be able to see it again!'. Below this, a token is listed: `✓ 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` with a copy icon. A green arrow points from the word 'PAT' in the text below to this token.

- Don't share a PAT!

Using the PAT

```
git clone https://github.com/george-boorman/bank_marketing
Cloning into 'bank_marketing'...
Username for 'https://github.com': george-boorman
Password for 'https://george-boorman@github.com':
```

```
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

```
ls
```

```
bank_marketing
```

Let's practice!

GITHUB CONCEPTS