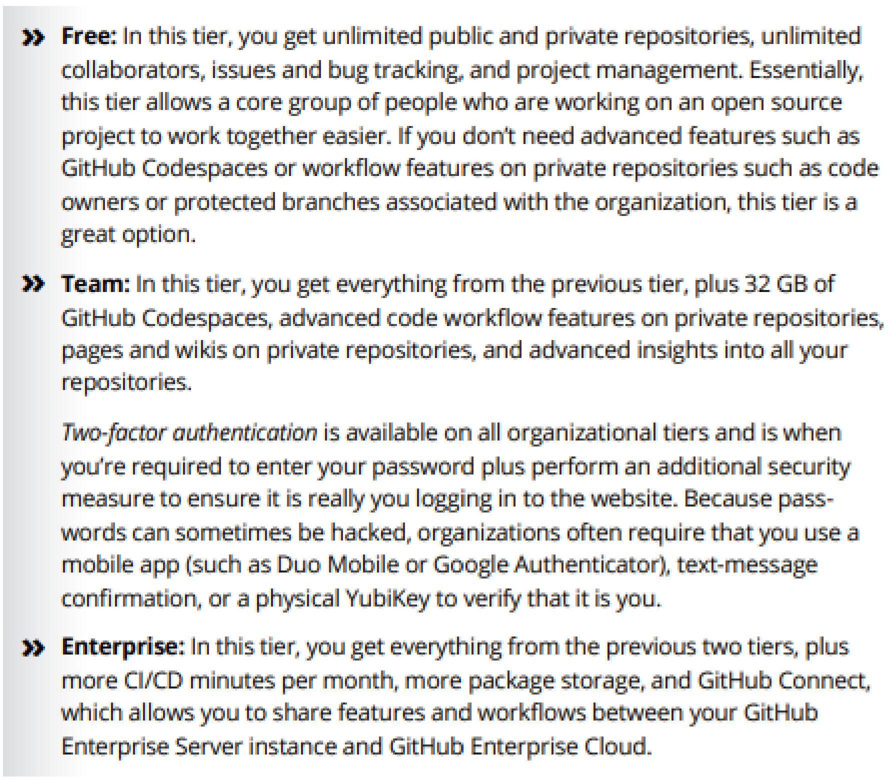
Unit-4

Why Code in Private?

Inner-sourcing is really just a play on words with open source. It implies that you use the same (or similar) strategies to collaborative code writing as open source, but you do it on private repositories.

Using GitHub Organizations:

1. **Creating a GitHub organization:**
   GitHub organizations allow you to give a group of users access to a set of repositories all at once. With features such as teams, you can also have subgroups of users who have different access rights to different repositories. This setup also makes communicating across GitHub easier because you can tag an organization (or team) instead of an individual person. If you're running a large project that has more than one repository, organizations may be a good option for you.
   You can start an organization in one of three tiers:

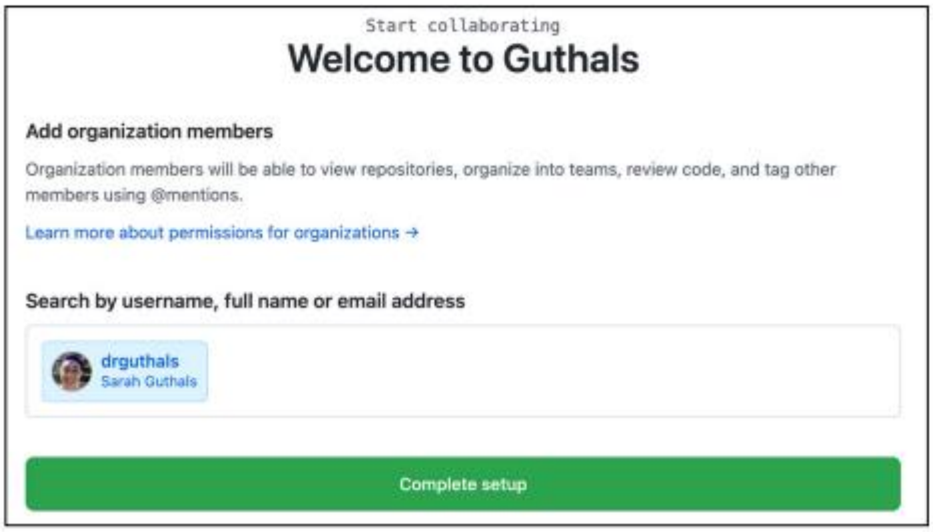


- **Free:** In this tier, you get unlimited public and private repositories, unlimited collaborators, issues and bug tracking, and project management. Essentially, this tier allows a core group of people who are working on an open source project to work together easier. If you don't need advanced features such as GitHub Codespaces or workflow features on private repositories such as code owners or protected branches associated with the organization, this tier is a great option.
- **Team:** In this tier, you get everything from the previous tier, plus 32 GB of GitHub Codespaces, advanced code workflow features on private repositories, pages and wikis on private repositories, and advanced insights into all your repositories.

  Two-factor authentication is available on all organizational tiers and is when you're required to enter your password plus perform an additional security measure to ensure it is really you logging in to the website. Because pass-words can sometimes be hacked, organizations often require that you use a mobile app (such as Duo Mobile or Google Authenticator), text-message confirmation, or a physical YubiKey to verify that it is you.
- **Enterprise:** In this tier, you get everything from the previous two tiers, plus more CI/CD minutes per month, more package storage, and GitHub Connect, which allows you to share features and workflows between your GitHub Enterprise Server instance and GitHub Enterprise Cloud.

2. **Inviting members to your GitHub organization:**
   During the organization setup process, you're asked if you want to invite others to join your organization. Using their GitHub.com alias, you can search for them in the provided box and send them an invite. If you forget to invite someone or want to invite them later, you can still do so by going to the People tab on the organization's home page and clicking Invite Member.

You can choose the permission level for the person you invite to your organization, shown in Figure 11-1. You can change the permission level later, if you need to.



Start collaborating

Welcome to Guthals

Add organization members

Organization members will be able to view repositories, organize into teams, review code, and tag other members using @mentions.

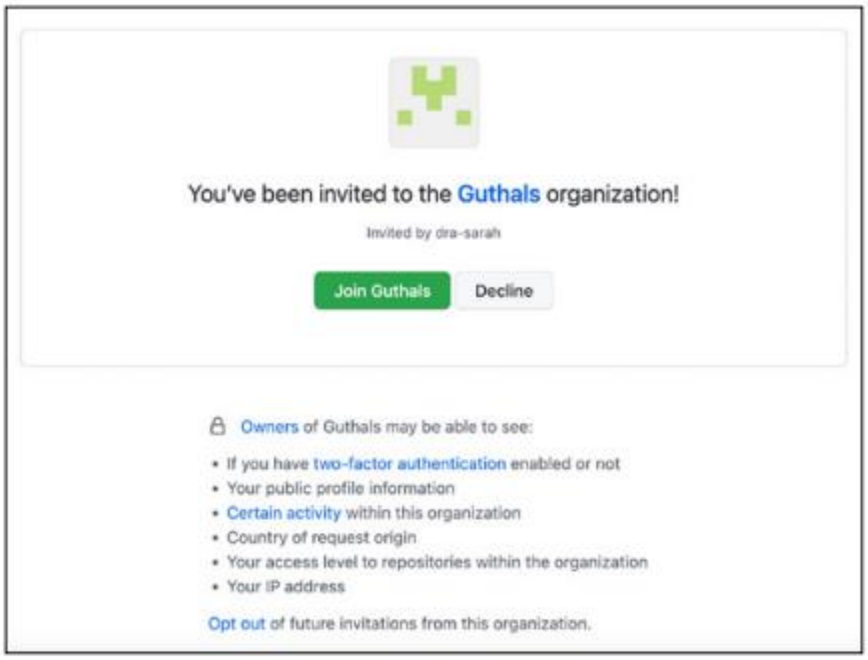
Learn more about permissions for organizations →

Search by username, full name or email address

drguthals
Sarah Guthals

FIGURE 11-1: Inviting someone to join your organization.

After you invite someone to your organization, they receive an email notification with a link to accept the invitation, or they can go to the organization's GitHub home page to accept the invitation. For example, in Figure 11-1, I invited Sarah to be a part of the Guthals organization because it's my primary account. When I visited https://github.com/Guthals while logged in to my primary GitHub account, I saw a banner and View Invitation button at the top of the page. Clicking the View Invitation button, I was taken to a new page, shown in Figure 11-2, where I was able to first see what kind of access the owners of the Guthals organization would have to my primary GitHub account information if I were to join



You've been invited to the Guthals organization!

Invited by dra-sarah

Owners of Guthals may be able to see:

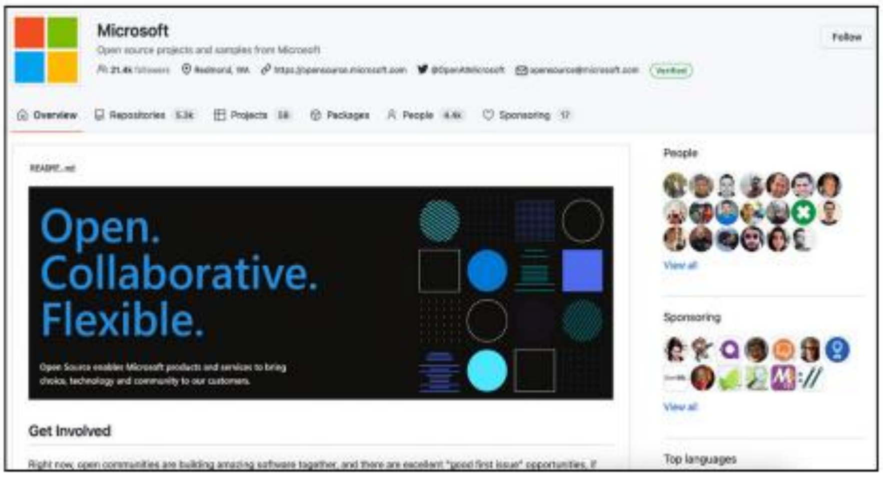
- If you have two-factor authentication enabled or not
- Your public profile information
- Certain activity within this organization
- Country of request origin
- Your access level to repositories within the organization
- Your IP address

Opt out of future invitations from this organization.

FIGURE 11-2: Invitation request from the invitee's account.

3. **Viewing repositories for your organization:**

Repositories is the default tab on your organization home page. This page shows you all the repositories associated with this organization. For example, if you go to the Microsoft organization home page on GitHub (https://github.com/ microsoft), you find more than 5,000 repositories and more than 4,000 people involved in open source projects for Microsoft (see Figure 11-3).



Right now, open communities are building amazing software together, and there are excellent "good first issue" opportunities. If

FIGURE 11-3: The overview page for the Microsoft organization on GitHub.com.

The Overview page has a README.md file at the top that describes how to get involved with open source projects maintained by the Microsoft organization. Pinned repositories appear just under the organization's README.md. Pinned repositories are the ones that the Microsoft organization owners think are the most relevant to folks interested in what Microsoft is doing in the open source space. For example, the VS Code repository has more than 140,000 stars and more than 24,300 forks. As one of the most popular editors and most popular open source projects, Microsoft wants to make sure this repository is front and center for visitors to their open source organization home page

4. **Managing members of your organization:**
You will always have at least one member of your organization — you! But this section is more interesting if you have more than one member, so if you haven't invited other members yet, go to the earlier section "Inviting members to your GitHub organization" and invite someone else. To see all your organization's members, from your organization home page, click the People tab. You should see all the members of your organization on this tab, as shown in Figure 11-4.

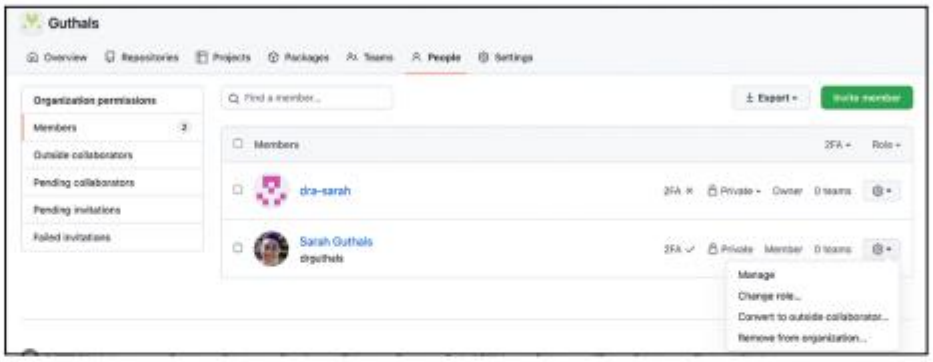


FIGURE 11-4: A list of members of an organization.

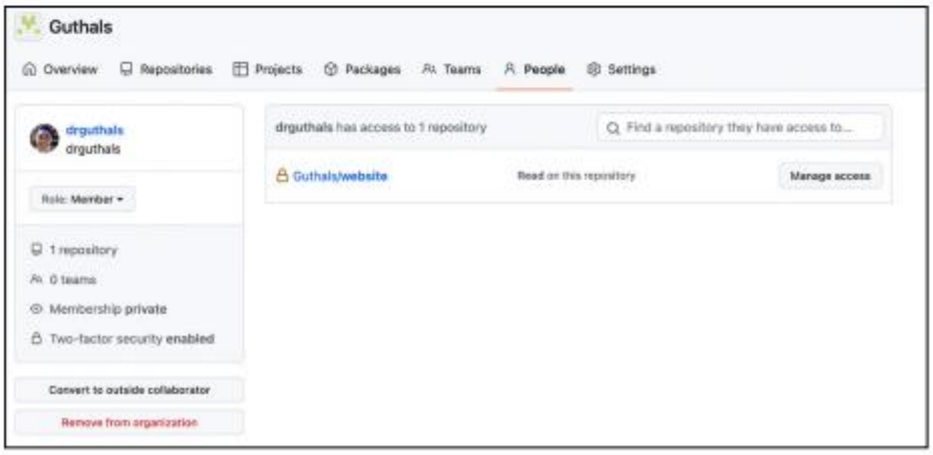
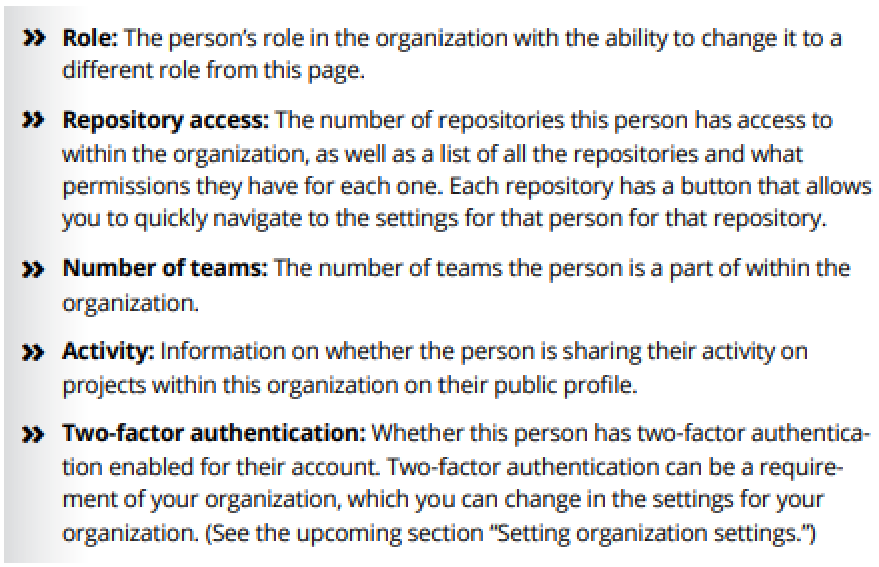
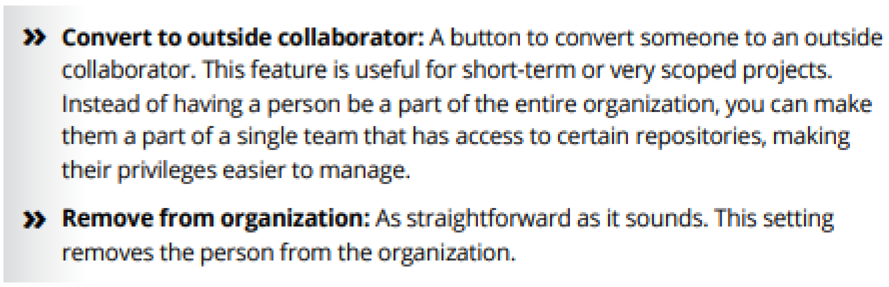


FIGURE 11-5: An overview of a member of an organization.

Figure 11-5 gives you the following information about the person:



- **Role:** The person's role in the organization with the ability to change it to a different role from this page.
- **Repository access:** The number of repositories this person has access to within the organization, as well as a list of all the repositories and what permissions they have for each one. Each repository has a button that allows you to quickly navigate to the settings for that person for that repository.
- **Number of teams:** The number of teams the person is a part of within the organization.
- **Activity:** Information on whether the person is sharing their activity on projects within this organization on their public profile.
- **Two-factor authentication:** Whether this person has two-factor authentication enabled for their account. Two-factor authentication can be a requirement of your organization, which you can change in the settings for your organization. (See the upcoming section "Setting organization settings.")



- **Convert to outside collaborator:** A button to convert someone to an outside collaborator. This feature is useful for short-term or very scoped projects. Instead of having a person be a part of the entire organization, you can make them a part of a single team that has access to certain repositories, making their privileges easier to manage.
- **Remove from organization:** As straightforward as it sounds. This setting removes the person from the organization.

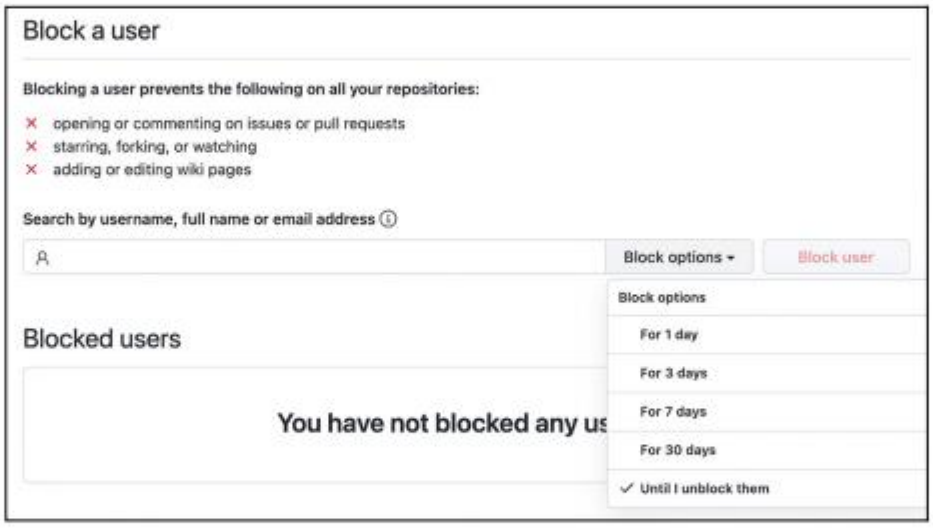
5. **Creating teams within your organization:**
As your organization begins to grow, it may make sense for you to create teams within your organization. The benefit of teams is that you can quickly give access to a repository to an entire team, without having to remember every single person that is on that team. To create a team, click the Teams tab on your organization home page and click New Team. A new page appears where you can choose a team name, add a description, choose a parent team (if you've created other teams already), and set the team's visibility within the organization.

Creating teams has a lot more benefits than the ability to mention them all using one alias.

6. **Setting organization settings:**
Organizations have a few more settings than typical individual GitHub accounts. On the right-most Settings tab on the organization home page a Settings page that is similar to the one I describe in Chapter 1, but with some key differences:
    - **General** is where you can change the organization's name, avatar, and
primary contact email address or even delete the organization. You can also choose to join the GitHub Developers Program (you can read about at https://developer.github.com/program). If your organization is representing a corporation, you can even sign the corporate terms of service, which helps protect your IP even on public repositories.
    - **Access** is where you can specify what level of access your members and repositories have. Here you find the billing and plans, repository roles,member privileges, team discussion settings, functionality to import and export data, and options for moderation.

Under the billing and plans setting, you can add billing managers to your organization; this can be really useful for folks within your company who need to have access to billing information, but may not be savvy or interested in the code aspect.

- **Code**, **planning, and automation** is where you can specify settings across your repositories, manage your Actions, webhooks, discussions, packages, pages, and projects.
- **Security** is the area where you can require that all members of your organization have two-factor authentication, set up SSH certificates, create IP allow lists, manage code security, and verify a domain that you own so that you can verify your organization's identity on GitHub.
- **Third-party access** is all third-party applications that you have given access to your repositories.
- **Integrations** has the integrations with your organization; you can connect a Slack organization to your GitHub organizations to get started.
- **Archives** has the log of all activity done to the organization (not the individual repositories part of the organization), and a list of deleted repositories.
- **Developer Settings** has two settings — OAuth Apps and GitHub Apps — and a place to specify management of the organization (under GitHub Apps)



Block a user

Blocking a user prevents the following on all your repositories:

× opening or commenting on issues or pull requests
× starring, forking, or watching
× adding or editing wiki pages

Search by username, full name or email address

Block options ▾ Block user

Block options
For 1 day
For 3 days
For 7 days
For 30 days
✓ Until I unblock them

Blocked users

You have not blocked any us

FIGURE 11-6: Options for blocking an individual.

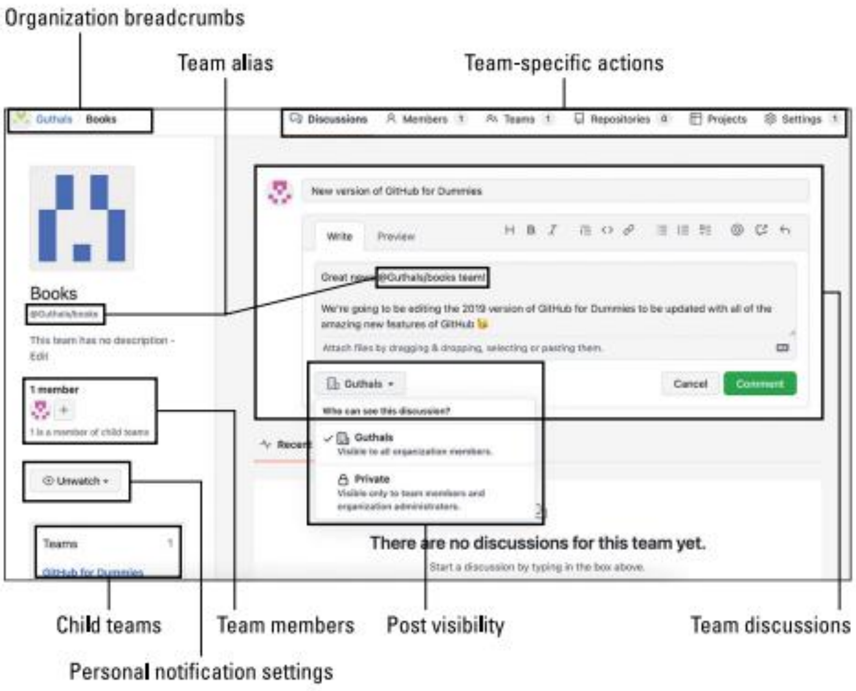
Making the Most of Your Teams

Having an organization with members grouped into various teams can be useful simply for understanding who is doing what, but GitHub provides you with tools that make your workflow even more effective when you group organization members into teams.

1. **Creating parent/child teams**
   You can add a team as a parent team to a new team that you're creating (see the section "Creating teams within your organization," earlier in this chapter). Essentially, teams can have hierarchies. The permissions of the parent teams are passed down to all child teams, but not vice versa. Hierarchies can become extremely useful when you want to ensure that everyone within an organization has access to exactly what they need and no more, no less.

As an example, you may have a team named Employees at the top of your hierarchy. As part of that team, you may have three other teams: Human Resources, Marketing, and Engineering. Under the Human Resources team, you may have some private project boards or private repositories that only the Human Resources team members should be able to see (such as employee personal information). Under the Marketing team, you may have customer data that shouldn't be shared publicly within the organization (such as billing account information). By assigning all the employees to respective child teams within the organization, you can ensure that everyone has access to the employee handbook, while only the folks in Human Resources have access to Social Security numbers.

2. **Discussing teams**
   GitHub offers the feature of team discussions when you have an organization with teams. Figure 11-7 shows this team discussion home page, which you can find at https://github.com/orgs/ORGNAME/teams/TEAMNAME where ORGNAME and TEAMNAME are replaced with the actual names of the organization and team, respectively.



New version of GitHub for Dummies

Great news @Guthals/books team!

We're going to be editing the 2019 version of GitHub for Dummies to be updated with all of the amazing new features of GitHub

Attach files by dragging & dropping, selecting or pasting them.

There are no discussions for this team yet.

Start a discussion by typing in the box above.

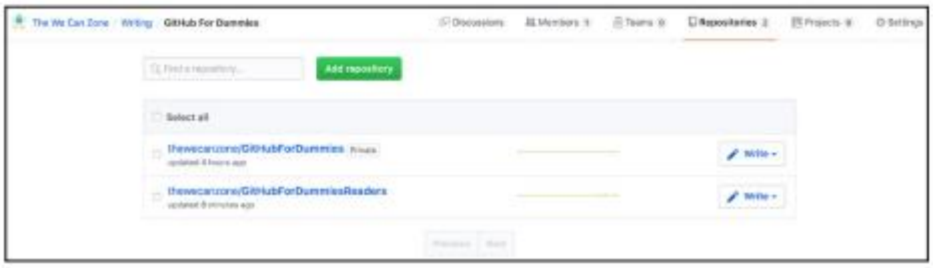
FIGURE 11-7: Discussions on a Team page.

From the team discussion home page, you find
» **Organization breadcrumbs**: On the top left, the hierarchy of this team, all the way up to the organization in a clickable breadcrumb-like fashion for easy navigation.
» **Team-specific actions:** On the top right, similar menu items that are found on repositories and the organization itself, but all are specific to this team.
» **Team discussion:** In the center right, a place to add new posts to the team discussion page. Below this area, you can find previous posts, and you can even pin specific posts to stay at the top (useful for important announcements or onboarding information).
» **Team alias:** The team alias appears below the team avatar and name on the left-hand column. You can use this alias to notify everyone in the team about something that may be of interest to them, as seen in the team post.
» **Team members:** A list of all team members, a count of how many also belong to child teams, and a button for adding additional members to the team.

» **Personal notification settings:** On the left-hand column, a place to quickly
change your notification settings for this team.
» **Child teams:** A list of all child teams to this team that are linked for quick
navigation, located on the bottom of the left column.

3. **Assigning code owners:**
A really neat feature that GitHub has that is enhanced with the use of teams is the CODEOWNERS file. Code owners is a way to specify certain people who may have the most experience and knowledge on a piece of code. They're the people who, when you're looking for someone to review your code, you most likely want to take a look. They probably have the most history with the code or are the most expert in that domain. By creating teams within your organization, you can assign multiple people to be code owners of certain code, without having to assign each person or keep the list up to date.

To get code owners working on one of your repositories, first make sure that you have a team setup that includes members of the organization who should be held responsible for ensuring all code that gets added to that repository is correct. In my example, I have a team called GitHub For Dummies. On that team page, make sure that you've linked the repository that you want this team to be code owners of and make sure that this team has at least Write access to that repository. Figure 11-8 shows that my team has two repositories that I have Write access to: the public one that you can access as a part of this book, and a private one that I'm using to track the progress of writing this book. I'm using code owners for the public repository



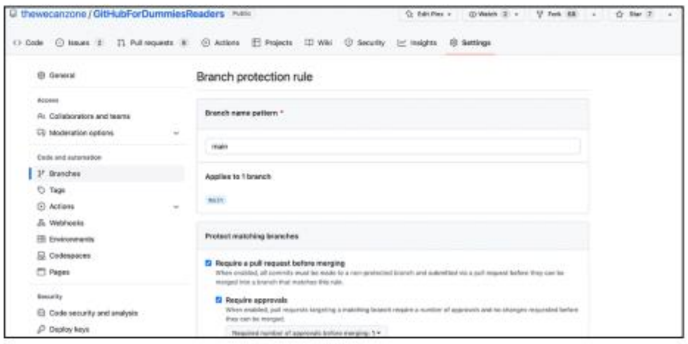
thewecanzone/GitHubForDummies Private
updated 4 hours ago
thewecanzone/GitHubForDummiesReaders
updated 8 minutes ago

FIGURE 11-8: Repository access for a specific team.

In the repository where you want the code owners to be automatically added as a reviewer for all pull requests, add a file on the main branch .github/CODEOWNERS with the following code:

```
# These owners will be the default owners for everything in
# the repo. Unless a later match takes precedence,
# @thewecanzone/github-for-dummies will be requested for
# review when someone opens a pull request.
@thewecanzone/github-for-dummies
```

Make sure under the Settings for that repository, you've added a branch protection to ensure that every time someone tries to make changes to the main branch using a pull request, they have to get their code reviewed and approved by at least one person and that each pull request requires a code owner to review and approve it. You can set up these requirements in the Branches area of the Settings for the repository, as shown in Figure 11-9.


thewecanzone / GitHubForDummiesReaders

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

General · Access · Collaborators and teams · Moderation options · Code and automation · Branches · Tags · Actions · Webhooks · Environments · Codespaces · Pages · Security · Code security and analysis · Deploy keys

Branch protection rule

Branch name pattern *

main

Applies to 1 branch

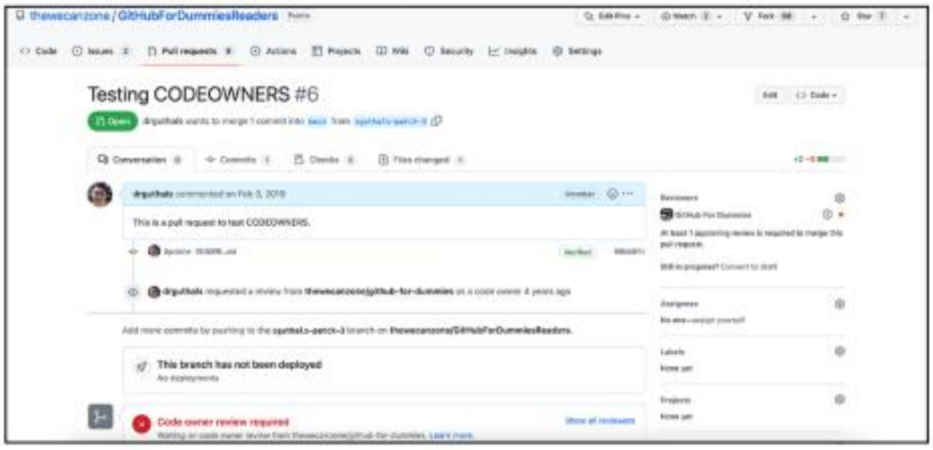
Protect matching branches

☑ Require a pull request before merging

☑ Require approvals

FIGURE 11-9: Requiring at least one approving review from a code owner.

Now, when you try to add code by opening a pull request, the team @thewecanzone/github-for-dummies is automatically added to every single pull request, as shown in Figure 11-10.


thewecanzone / GitHubForDummiesReaders

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

Testing CODEOWNERS #6

Open

This is a pull request to test CODEOWNERS.

This branch has not been deployed

Code owner review required

FIGURE 11-10: Code owners are automatically added to a pull request.

Best Practices for Inner-Sourcing

1. **Repository insights:**
When you’re a part of a company, evaluating an engineer on their contributions to the team can get tricky. Either as an individual trying to make sure you’re making good progress or as a manager trying to write end-of-year reviews and performance ratings, you want to avoid using information to qualify someone’s overall contributions and performance.
That being said, you can get some information to help guide conversations to
more effectively evaluate yourself or another individual. On a repository, you can
click the Insights tab at the top right. You see a list of insights about your repository and the people that contribute to it:
» **Pulse** is a snapshot of your repository for a specific time period (defaulted to
one week). For example, in the month of October 2022, the Microsoft/VSCode
project had the following pulse summary: Excluding merges,36 authorshave pushed
208 commits to main and 319 commits to all branches. On main,481 fileshave
changed and there have been10,619additions and 7,883deletions.
» **Contributors** shows an overall contribution graph for the repository, as well
as individual contribution graphs for each contributor with a summary of the

number commits, lines added, and lines removed.Tracking contributions to a project simply by lines of code modified or number of commits isn't an effective way to evaluate someone if nothing else is taken into account. For example, Shawn could commit every 15 minutes for fear of losing work, and Sam could always focus his energies on refactoring, making the number of lines of code they change substantial. Comparing these two to Sandra, who had only a few commits and not as many lines of modified code but found and fixed a security vulnerability that could have taken down the application for good, isn't a fair comparison.

» **Community/Traffic:** On public repositories, the community profile on a repository helps maintainers understand where they can improve their repository to better support their community. You can find out more at https://help.github.com/articles/about-community-profiles-for public-repositories. On private repositories, the Traffic section gives you insight into who is coming to this repository, where they're coming from, and what files they're interacting with.

» **Commits:** This simple commit graph can give you insights into problems your repository may have or trends with the lives of your contributors. For example, if contributions drop off every year around May, maybe a computer science student contributes to your repository as a part of the course they are taking. And if you typically have 300 commits a week but suddenly you start getting only 50 commits, maybe folks are running into a problem where they can't get their code working well enough to commit.

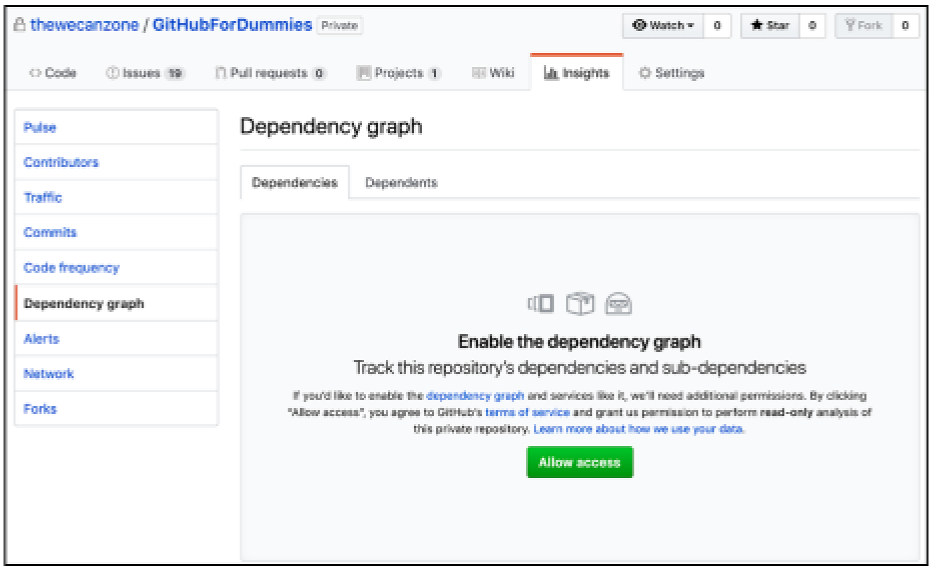
» **Code frequency:** Similar to commits, code frequency shows the frequency at which lines of code are added and lines of code are deleted. Identifying patterns here can help you understand the health of your code as well as the profile of your contributors.

» **Dependency graph:** The dependency graph lists all dependencies (and dependents) of this repository and the version it depends on.

» **Alerts:** On a private repository, you have an Alerts section. If you've enabled read-only access to GitHub either via allowing it through the dependency graph setting shown in Figure 11-11 or in the Settings tab for the repository in the Options category under Data Services, as shown in Figure 11-12, you see security alerts here.

» **Network:** The network graph shows all the people who have branched and forked the repository and any branches of branches or forks of forks. Essentially, it shows all the possible states of your repository in the world today. Figure 11-13 shows the network graph for Visual Studio Code.

» **Forks:** The Forks tab lists links to all forks of your repository.



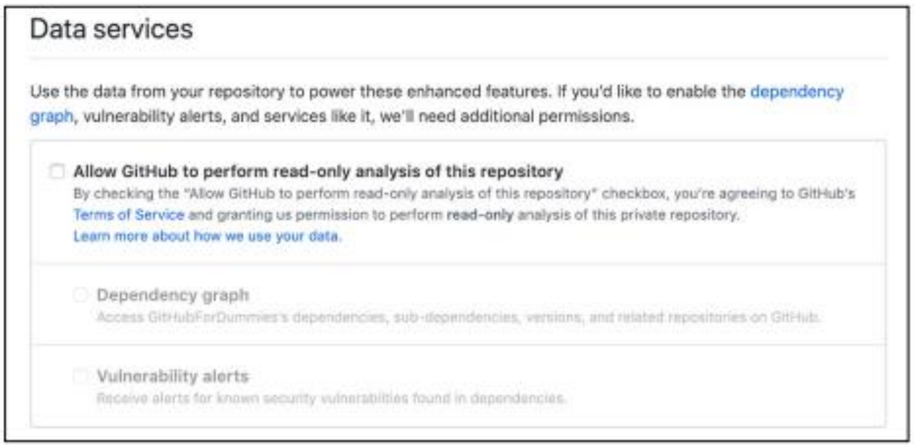
Dependency graph

Enable the dependency graph

Track this repository's dependencies and sub-dependencies

If you'd like to enable the dependency graph and services like it, we'll need additional permissions. By clicking "Allow access", you agree to GitHub's terms of service and grant us permission to perform **read-only** analysis of this private repository. Learn more about how we use your data.

FIGURE 11-11: GitHub must have read access to your repository to give security alerts.



Data services

Use the data from your repository to power these enhanced features. If you'd like to enable the dependency graph, vulnerability alerts, and services like it, we'll need additional permissions.

Allow GitHub to perform read-only analysis of this repository

By checking the "Allow GitHub to perform read-only analysis of this repository" checkbox, you're agreeing to GitHub's Terms of Service and granting us permission to perform **read-only** analysis of this private repository. Learn more about how we use your data.

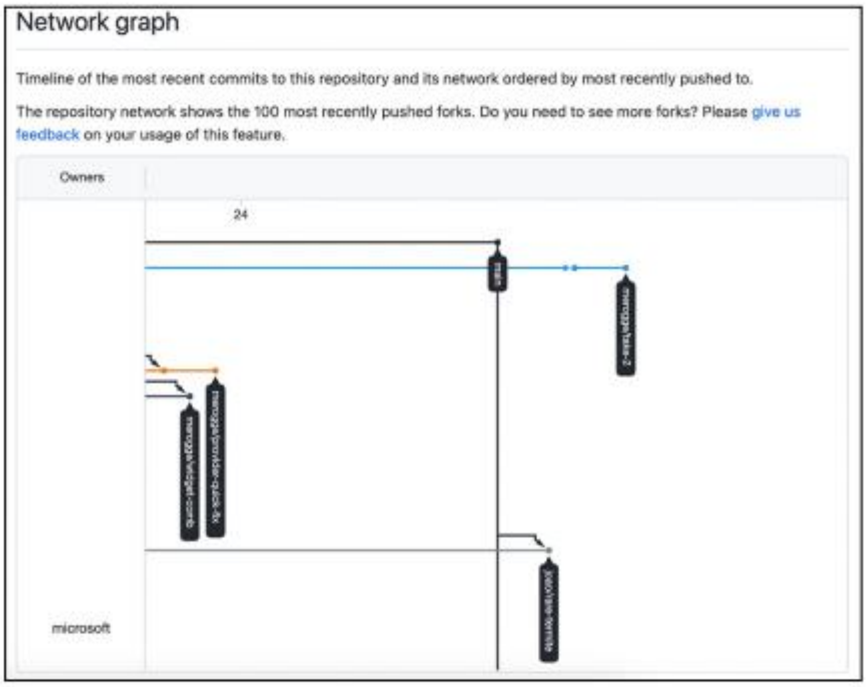
Dependency graph

Access GitHubForDummies's dependencies, sub-dependencies, versions, and related repositories on GitHub.

Vulnerability alerts

Receive alerts for known security vulnerabilities found in dependencies.

FIGURE 11-12: Allow GitHub read access to your repository for various data services.



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

The repository network shows the 100 most recently pushed forks. Do you need to see more forks? Please give us feedback on your usage of this feature.

FIGURE 11-13: The network graph for the Visual Studio Code open source project.

2. **Milestones for larger projects:**

Project boards are an effective way to track the progress you're making through the issues and pull requests, but when you're working as part of a larger organization, you often have larger milestones that you're trying to reach. GitHub provides support for milestones that can be linked to issues and pull requests.

To create a milestone:

1. Go to the Issues tab of your repository and click the Milestones button.

    If you don't have a milestone yet, you can click the big, green New milestone button on the top right or the big, green Create a Milestone button in the center.

2. Add a title and optional due date and optional description.
3. Click Create Milestone.

You see a list of milestones

Make GitHub Work for you

Collaborating Outside of GitHub –

Chatting It Up:

For many teams, especially distributed teams, chat is a powerful way for members of the team to collaborate and coordinate their efforts. Chat in this context does not refer to sipping tea on a porch talking about how their day went. Chat refers to text-based tools, such as Slack, used by teams to communicate both synchronously and asynchronously.

Many teams find it helpful to have GitHub post important notifications into a chat room so teams are kept apprised of what's going on with a repository. In this section, I set up a GitHub integration with one popular chat software, Slack.

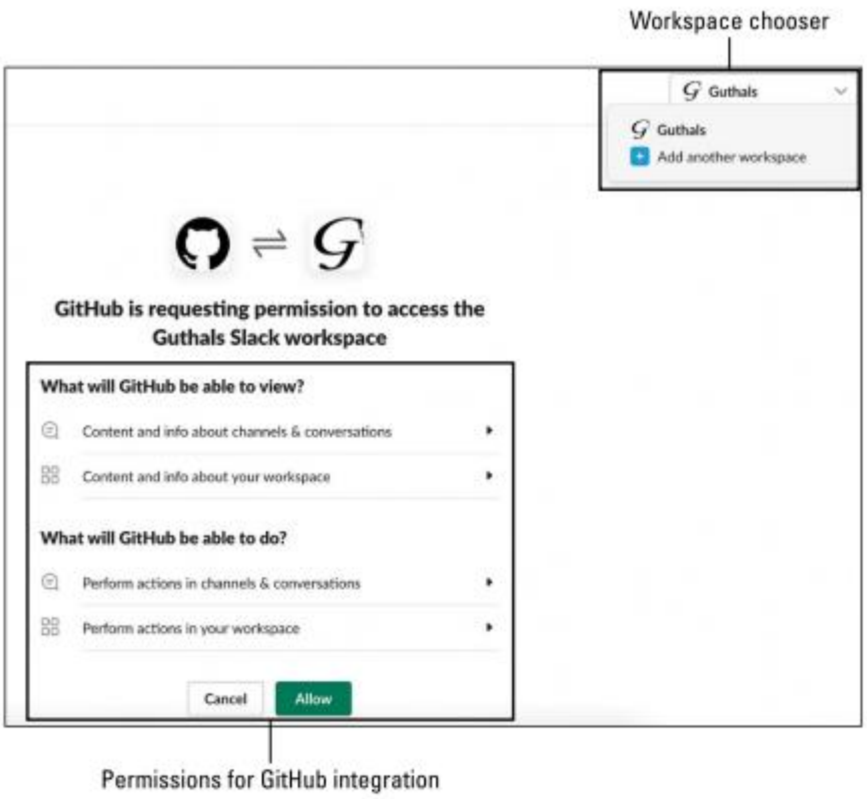
Before you install the integration, you need to be the admin of a Slack workspace. You can create a free Slack workspace at https://slack.com.

After you set up your Slack workspace, installing the GitHub for Slack integration requires two key steps:
1. Install the GitHub app for Slack in the Slack workspace.
2. Add the Slack App for GitHub to your GitHub account.
The following sections cover these steps in detail.

1. Installing the GitHub app for Slack

To install the GitHub app for Slack:

1. **Go to https://slack.github.com and click the Add to Slack button in the center of the browser window.**
2. **Click Allow, shown in Figure 12-1, to be taken to Slack to authenticate with GitHub.**
   As shown in Figure 12-2, this DM prompts you to sign in with GitHub.
3. **Click Connect to GitHub Account, and then enter the post authentication code.**
4. **Check out the instructions for how use the GitHub Slack integration.**
   As shown in Figure 12-3, you can subscribe specific Slack channels to specific repositories, create issues, and manage notifications from inside Slack



GitHub is requesting permission to access the Guthals Slack workspace

What will GitHub be able to view?

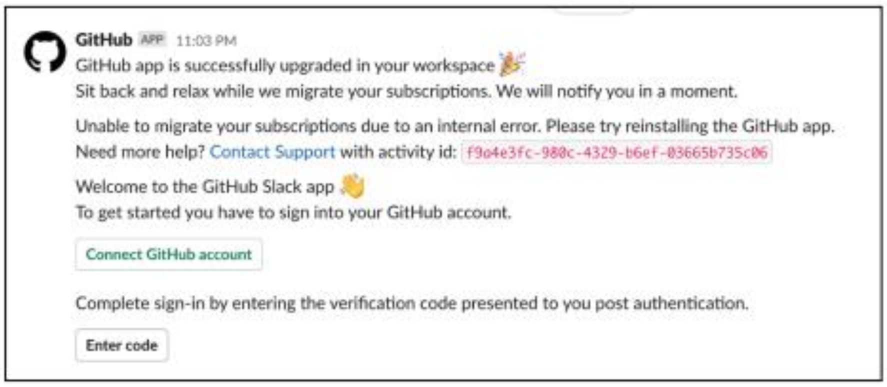
- Content and info about channels & conversations
- Content and info about your workspace

What will GitHub be able to do?

- Perform actions in channels & conversations
- Perform actions in your workspace

Cancel | Allow

FIGURE 12-1: Confirmation page for the GitHub app.



GitHub APP 11:03 PM

GitHub app is successfully upgraded in your workspace 🎉
Sit back and relax while we migrate your subscriptions. We will notify you in a moment.

Unable to migrate your subscriptions due to an internal error. Please try reinstalling the GitHub app.
Need more help? Contact Support with activity id: `f9a4e3fc-980c-4329-b6ef-03665b735c06`

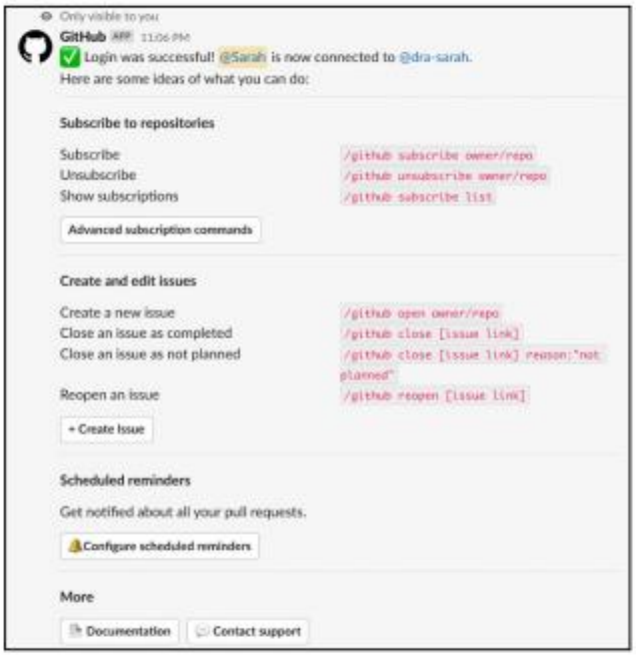
Welcome to the GitHub Slack app 👋
To get started you have to sign into your GitHub account.

Connect GitHub account

Complete sign-in by entering the verification code presented to you post authentication.

Enter code

FIGURE 12-2: Authenticate with GitHub from Slack.


Only visible to you

GitHub APP 11:06 PM

✅ Login was successful! @Sarah is now connected to @dra-sarah.

Here are some ideas of what you can do:

Subscribe to repositories

Subscribe	`/github subscribe owner/repo`
Unsubscribe	`/github unsubscribe owner/repo`
Show subscriptions	`/github subscribe list`

Create and edit issues

Create a new issue	`/github open owner/repo`
Close an issue as completed	`/github close [issue link]`
Close an issue as not planned	`/github close [issue link] reason:"not planned"`
Reopen an issue	`/github reopen [issue link]`

Scheduled reminders

Get notified about all your pull requests.

More

FIGURE 12-3: GitHub Slack direct message with instructions on how to use the integration.

2.Subscribing to a repository in a Slack channel:

After you install the GitHub app for Slack, you can subscribe to notifications for a GitHub repository from within a Slack channel, by typing:

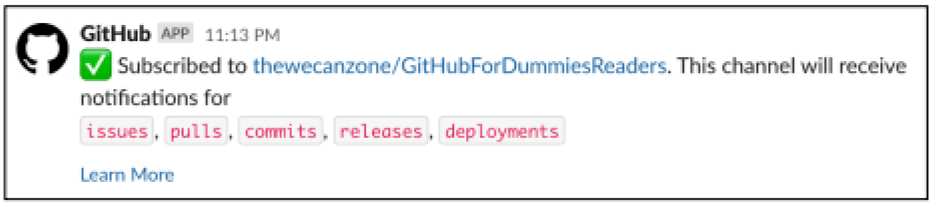
/github subscribe owner/repository

For example, to subscribe to the repository I created for the readers of this book,

https://github.com/thewecanzone/GitHubForDummiesReaders, you would type the following in a Slack channel:

/github subscribe TheWecanZone/GitHubForDummiesReaders

When you've successfully subscribed to a specific repository, you see the confirmation message shown in Figure 12-4.


GitHub APP 11:13 PM

✅ Subscribed to thewecanzone/GitHubForDummiesReaders. This channel will receive notifications for `issues`, `pulls`, `commits`, `releases`, `deployments`

FIGURE 12-4: A message from GitHub in Slack confirming subscription to a repository.

3.Trying out the GitHub Slack integration:

Trying out the GitHub Slack integration

With the installation complete, you can now subscribe to GitHub repositories in your Slack channels. To see the full list of Slack commands, type the following:

```
/github help
```

The output of this command is the same output you get when you first install the integration (refer to Figure 12-3).

To test the GitHub app and open a new issue:

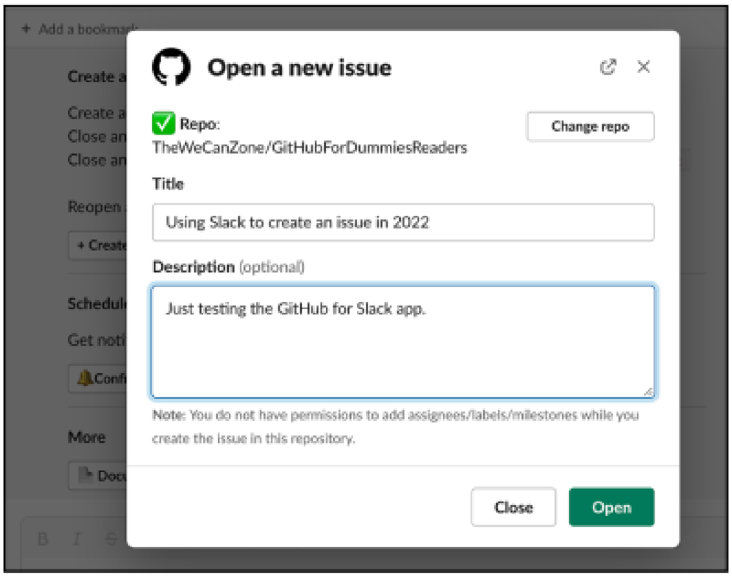
1. **Run the following command:**

   ```
   /github open TheWeCanZone/GitHubForDummiesReaders
   ```

   A Slack dialog box appears. You can use this dialog box to create a new issue, as shown in Figure 12-5.

2. **Fill in the dialog box and click the Open button.**

   Clicking the Open button creates the issue on GitHub. And because I'm subscribed to that repository, I get a Slack message in the channel that the issue was created, as shown in Figure 12-6.


Open a new issue

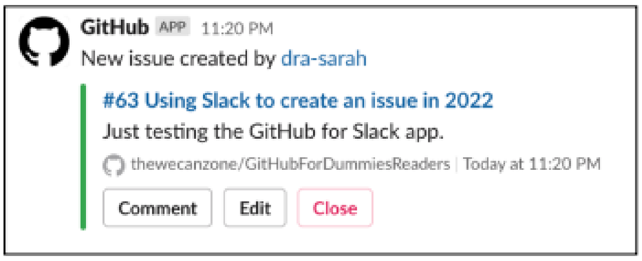
Repo: TheWeCanZone/GitHubForDummiesReaders

Title: Using Slack to create an issue in 2022

Description (optional): Just testing the GitHub for Slack app.

Note: You do not have permissions to add assignees/labels/milestones while you create the issue in this repository.

FIGURE 12-5: Slack dialog to create an issue on GitHub.


GitHub APP 11:20 PM
New issue created by dra-sarah

#63 Using Slack to create an issue in 2022
Just testing the GitHub for Slack app.
thewecanzone/GitHubForDummiesReaders | Today at 11:20 PM

FIGURE 12-6: Slack message with information about a newly created GitHub issue.

The /github subscribe command by default subscribes a channel to notifications for the following features of a repository:

» issues: Opened or closed issues

» pulls: New or merged pull requests

» commits: New commits on the default branch (usually main)

» releases: Published releases

» deployments: Updated status on deployments

» reviews: New reviews completed on pull requests

You can remove a single feature by using the /github unsubscribe owner/repo [feature] command. For example, to remove commit notifications on the default branch, run the following command. /github unsubscribe TheWeCanZone/GitHubForDummiesReaders commits

Getting Trello and GitHub Integrated

1. Installing the GitHub power-up:

The following installation instructions assume that you've already signed up for https://trello.com and created a project board:

1. With a board open, make sure the menu is open.

If not, click in the top right to show the menu.

2. **Click the Power-Ups section of the menu, as shown in Figure 12-7.**

Clicking the Power-Ups button brings up a search dialog box for power-ups.

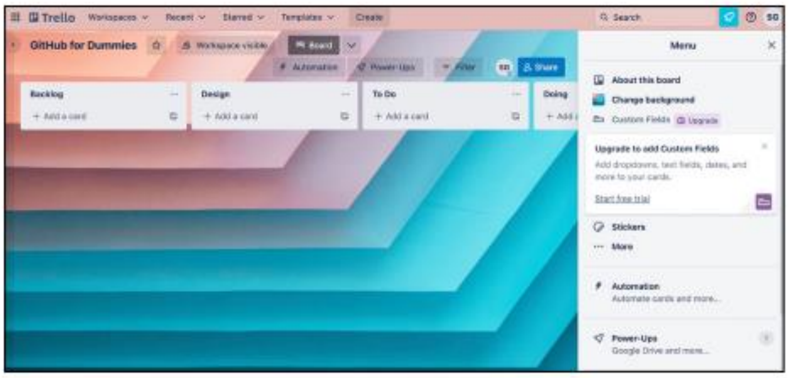


FIGURE 12-7: The power-ups section of the menu.

3**. Search for GitHub to find GitHub related power-ups.**

4. Click the Add button for the GitHub power-up to enable it.

5. After you enable the power-up, click the settings button to configure it.

You see a menu with the option to authorize or disable the power-up.

6. Click Authorize Account.

An option to link your GitHub account appears.

7. Click Link Your GitHub Account.

GitHub.com launches in your browser and prompts you to Authorize Trello, as shown in Figure 12-8

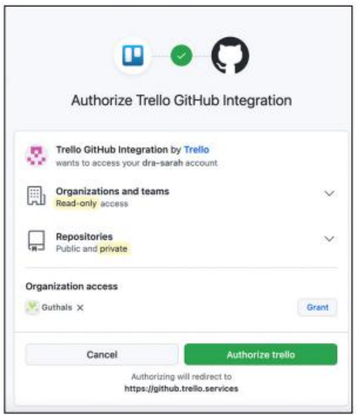

Authorize Trello GitHub Integration
Trello GitHub Integration by Trello
wants to access your dra-sarah account
Organizations and teams
Read-only access
Repositories
Public and private
Organization access
Guthals
Grant
Cancel
Authorize trello
Authorizing will redirect to
https://github.trello.services

FIGURE 12-8: Authorize Trello on GitHub.

8. **Click the Grant button next to any organizations that you want to connect with Trello.**
In my case, I granted Trello access to the guthals organization.
9. Click the Authorize Trello button to make the power-up active.

2. Using the GitHub power-up:

The GitHub power-up is accessed via the power-up button on the back of any Trello card. If you haven't already, go ahead and create a couple of cards.
To use the GitHub power-up on your Trello board, follow these steps:

1. **Click the card to access the back of the card.**
Figure 12-9 shows a card that I created. The GitHub power-up shows up in the bottom-right corner.
2. **Click the GitHub Power-Up button.**
Four menu options appear:
• Attach Branch
• Attach Commit
• Attach Issue
• Attach Pull Request
3. **Click Attach Issue to open a repository search dialog box**.
4. **Find the repository that contains the issue you want to attach to the card**.
After you select the repository, you see a list of issues, as shown in
Figure 12-10. You can also search for issues.
5. **Select the issue from the drop-down list and it attaches to the Trello card.**
After you attach the issue, the issue is displayed on the back of the Trello card

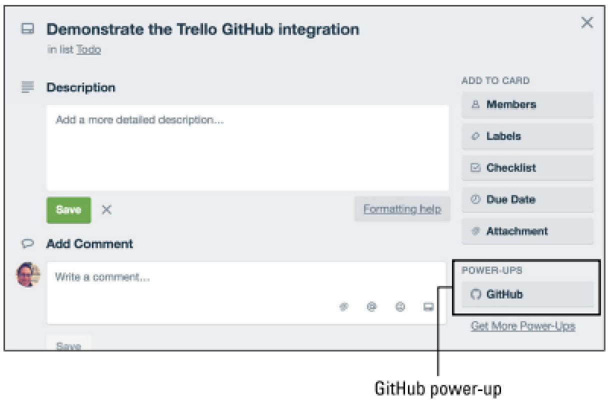


FIGURE 12-9: Trello card with a GitHub power-up.

A Trello card may be attached to multiple GitHub items. For example, repeat the previous steps, but choose Attach Pull Request instead of Attach Issue to attach a pull request to an issue. When you are done, you see both an issue and a pull request attached to the Trello card, as shown in Figure 12-11.

The front of the card shows a couple icons that indicate that this card is attached to GitHub. It shows an Octocat icon with a count of GitHub items attached to the card. It also shows pull request icon with a count to indicate the number of pull requests attached, as shown in Figure 12-12

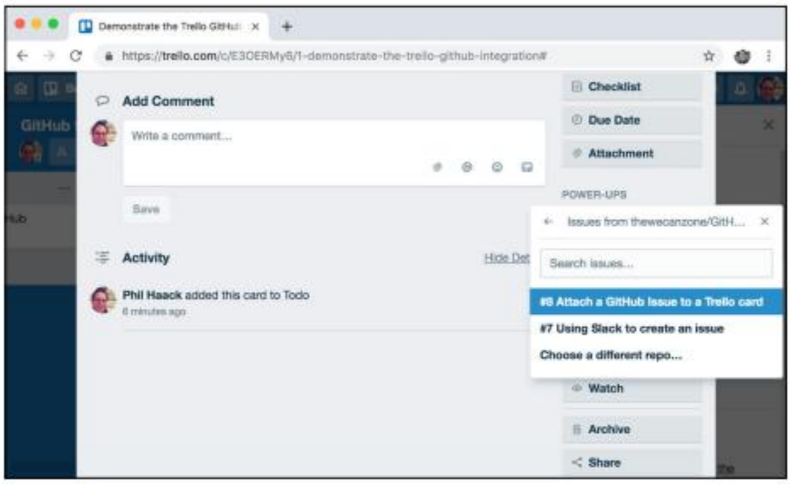

Demonstrate the Trello GitHub
https://trello.com/c/E3OERMy6/1-demonstrate-the-trello-github-integration#
Add Comment
Write a comment...
Save
Activity
Hide Det
Phil Haack added this card to Todo
6 minutes ago
Checklist
Due Date
Attachment
POWER-UPS
Issues from thewecanzone/GitH...
Search issues...
#8 Attach a GitHub Issue to a Trello card
#7 Using Slack to create an issue
Choose a different repo...
Watch
Archive
Share

FIGURE 12-10: Selecting the issue to attach.

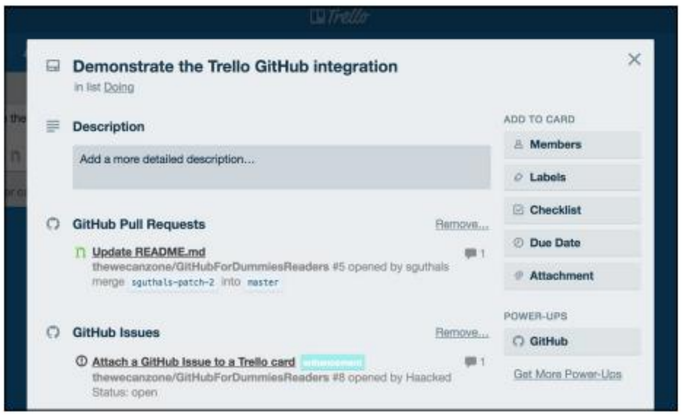

Demonstrate the Trello GitHub integration
in list Doing
Description
Add a more detailed description...
GitHub Pull Requests
Remove...
Update README.md
thewecanzone/GitHubForDummiesReaders #5 opened by sguthals
merge sguthals-patch-2 into master
GitHub Issues
Remove...
Attach a GitHub Issue to a Trello card
thewecanzone/GitHubForDummiesReaders #8 opened by Haacked
Status: open
ADD TO CARD
Members
Labels
Checklist
Due Date
Attachment
POWER-UPS
GitHub
Get More Power-Ups

FIGURE 12-11: Trello card with an issue and pull request attached.

When you visit the issue or pull request on GitHub.com, you can see that the attachment is bidirectional. The GitHub issue now has a link to the Trello board, as shown in Figure 12-13.

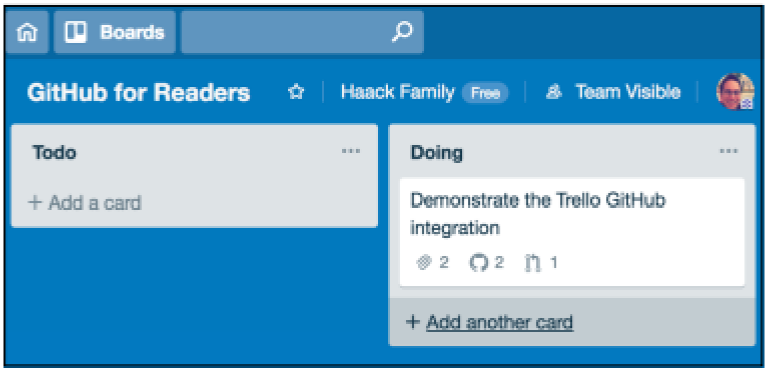
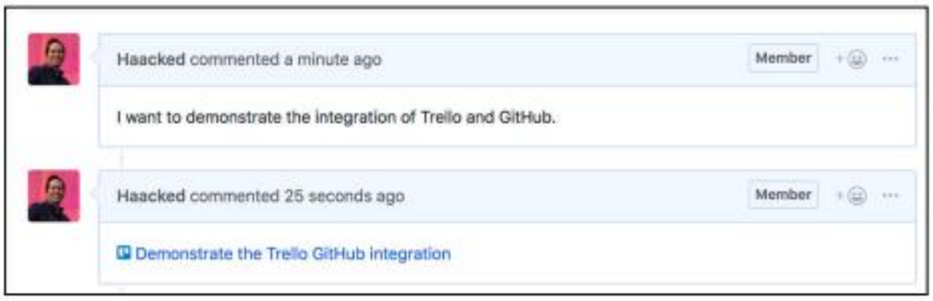


FIGURE 12-12: Front of a card with an issue and pull request attached.


Haacked commented a minute ago Member

I want to demonstrate the integration of Trello and GitHub.

Haacked commented 25 seconds ago Member

Demonstrate the Trello GitHub integration

FIGURE 12-13: GitHub issue with a link to the Trello board.

Managing Notifications with Octobox:

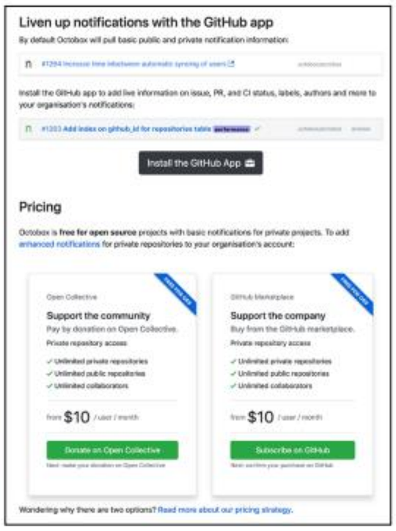
Earlier in this chapter, I cover a couple of integrations that bring GitHub information into other collaboration tools. GitHub integrations help teams work together. In this section, I cover a GitHub app that's a little different. It's a tool to help individuals manage the flow of GitHub notifications. As you participate in more and more GitHub repositories, the number of notifications can start to get overwhelming. Octobox provides an email client style view of your notifications.

Installing Octobox is pretty straightforward:

1. Go to https://octobox.io and scroll down to the button labeled Install the GitHub App.

Some pricing options appear, as shown in Figure 12-14. Octobox is free for open source projects.

2. Click Install the GitHub App to continue with the installation process.
3. Authorize the application the same way you authorized Slack and Trello, earlier in this chapter.



Liven up notifications with the GitHub app

By default Octobox will pull basic public and private notification information:

#1294 Increase time inbetween automatic syncing of users

Install the GitHub app to add live information on issue, PR, and CI status, labels, authors and more to your organisation's notifications:

#1203 Add index on github_id for repositories table performance ✓

Pricing

Octobox is **free for open source** projects with basic notifications for private projects. To add enhanced notifications for private repositories to your organisation's account:

Open Collective

Support the community

Pay by donation on Open Collective.

Private repository access

- ✓ Unlimited private repositories
- ✓ Unlimited public repositories
- ✓ Unlimited collaborators

from $10 / user / month

Next: make your donation on Open Collective

GitHub Marketplace

Support the company

Buy from the GitHub marketplace.

Private repository access

- ✓ Unlimited private repositories
- ✓ Unlimited public repositories
- ✓ Unlimited collaborators

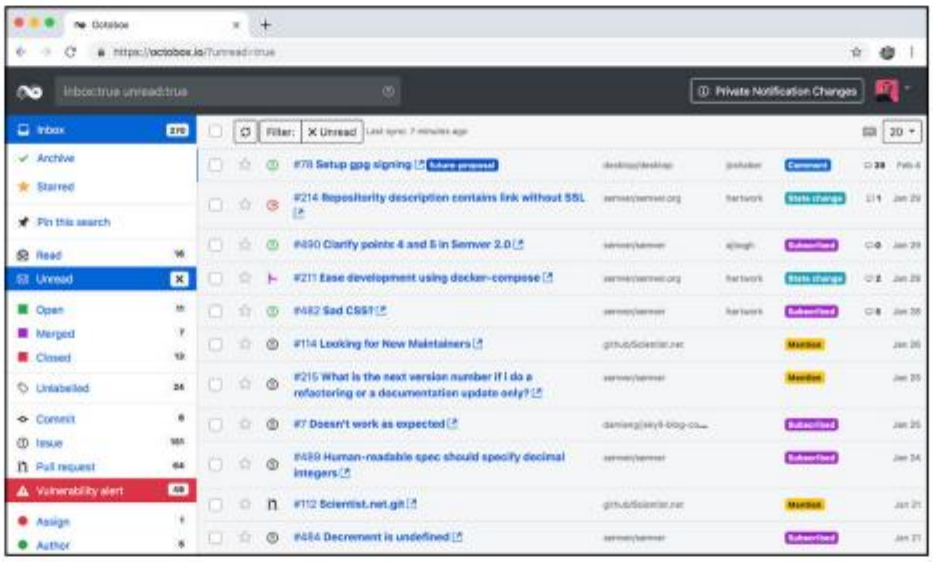
from $10 / user / month

Next: confirm your purchase on GitHub

Wondering why there are two options? Read more about our pricing strategy.

FIGURE 12-14: GitHub app download button and pricing options for Octobox.

After the installation and authorization steps are complete, you're taken to your Octobox inbox. The first time it runs, it takes a moment to synchronize your notifications. When it's done, you should see something like Figure 12-15.



#78 Setup gpg signing

#214 Repository description contains link without SSL

#490 Clarify points 4 and 5 in Semver 2.0

#211 Ease development using docker-compose

#482 Sad CSS?

#114 Looking for New Maintainers

#215 What is the next version number if I do a refactoring or a documentation update only?

#7 Doesn't work as expected

#489 Human-readable spec should specify decimal integers

#112 Scientist.net.git

#484 Decrement is undefined

FIGURE 12-15: Octobox inbox.

After Octobox is installed and synchronized, you can use it to manage your notifications. It allows you to search and filter your notifications by repository, organization, type, action, status, and so on. You can set Octobox to automatically synchronize on an interval in its Settings page. As the status for issues and pull requests change on GitHub, synchronizing Octobox displays those changes in Octobox. Octobox also provides archiving and muting for notifications, which is a nice way of staying on top of notifications, especially if you work on multiple active projects on GitHub.

GitHub Workflow Integrations –
Using GitHub for Visual Studio Code:

1. **Interacting with pull requests in VS Code**
   After you're signed in, when you go to the Source Control tab on the left side of VS Code, you should see all the pull requests associated with this repo. Pull requests are grouped into five different sections:
   » Local Pull Request Branches: Ones that you currently have checked out on your machine
   » Waiting for My Review: Ones where you are marked as a reviewer
   » Assigned to Me: Ones that you're assigned to
   » Created by Me: Ones that you created
   » All Open: A list of all pull requests that are open for the repository

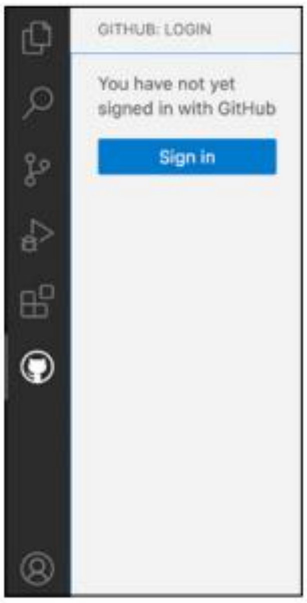
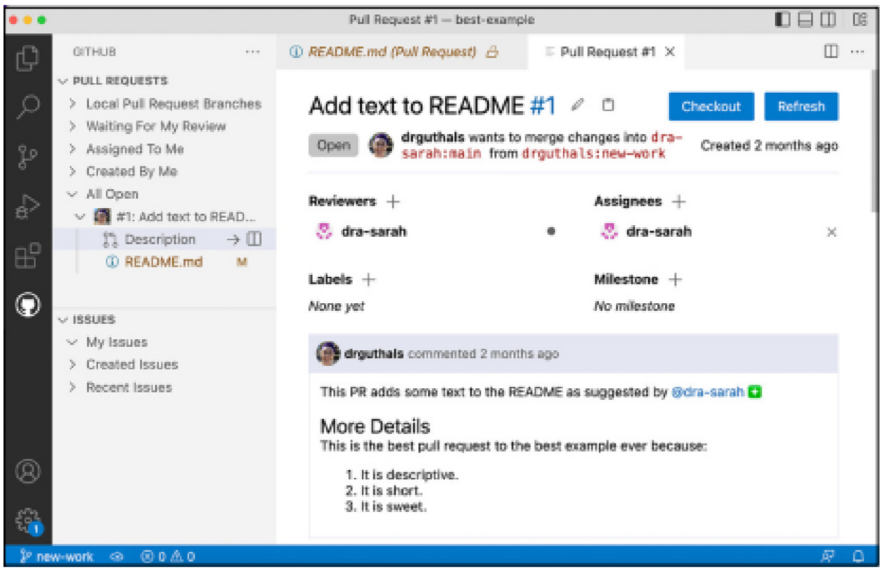


FIGURE 13-2: Initiate the GitHub sign-in process.

When you unroll a specific pull request, you see the description of the pull request, along with all the modified, added, or deleted files. Clicking the description of the pull request displays the description as you would see it on GitHub, as shown in Figure 13-3. From this page, you can check out or refresh the pull request; leave a comment; modify reviewers, assignees, labels, or milestones; merge or update the pull request; or complete a review. You have the entire pull request experience right inside of VS Code! Another feature of this extension is the ability to add inline comments to the diff. Clicking a specific modified file shows you the side-by-side diff, just as it would look on GitHub.com. From here, you can add a comment to any of the modified lines, as shown in Figure 13-4. All these actions are reflected on GitHub.com.



Add text to README #1

Open drguthals wants to merge changes into dra-sarah:main from drguthals:new-work Created 2 months ago

Reviewers: dra-sarah
Assignees: dra-sarah
Labels: None yet
Milestone: No milestone

drguthals commented 2 months ago

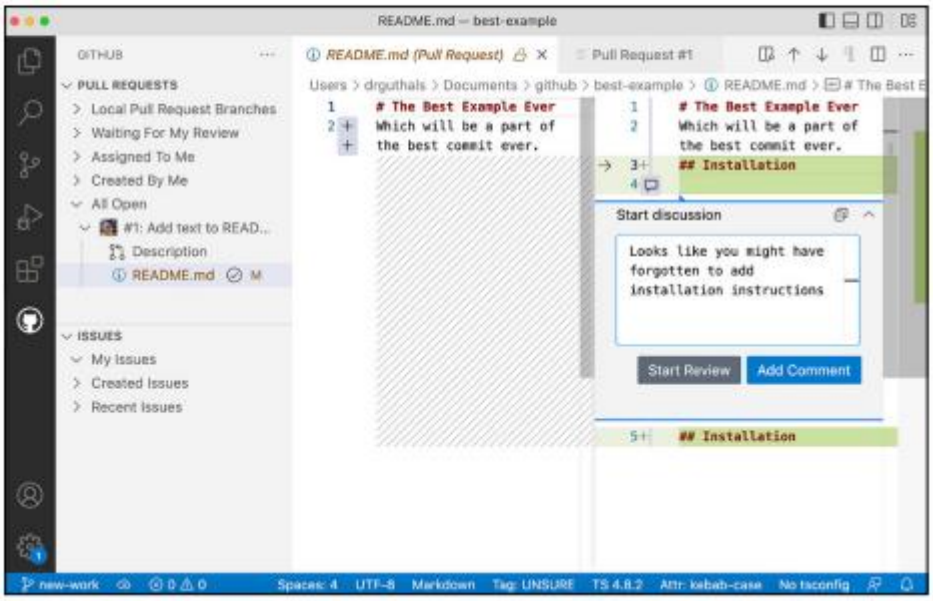
This PR adds some text to the README as suggested by @dra-sarah

More Details

This is the best pull request to the best example ever because:

1. It is descriptive.
2. It is short.
3. It is sweet.

FIGURE 13-3: Interacting with pull requests in VS Code.



```
# The Best Example Ever
Which will be a part of
the best commit ever.
## Installation
```

Looks like you might have forgotten to add installation instructions

FIGURE 13-4: Adding an inline comment in VS Code.

Following the GitHub for VS Code pull requests extension Because this extension is also open source, you can follow the development, report issues, or even contribute to it on GitHub.com. Go to https://github.com/

Using GitHub for Visual Studio:

Visual Studio is different from Visual Studio Code. Visual Studio is an integrated deveopment environment (IDE) and is a full-featured application to support developers in writing code, while VS Code is an editor that has an extensive list of extensions that a developer can add to create the experience they need. Visual Studio 2015, 2017, and 2019 have a specific GitHub integration, while Visual Studio 2022 has Git and GitHub integrated out of the box. It is recommend that you install and use Visual Studio 2022; however, if you have to use an older version, this section shows you how to integrate GitHub with Visual Studio 2017.

TIP

If you do install Visual Studio 2022, you can use GitHub very similarly to how to use it in VS Code. Head to `https://visualstudio.microsoft.com/vs/github/` for guided instructions on the newest version control features of Visual Studio 2022.

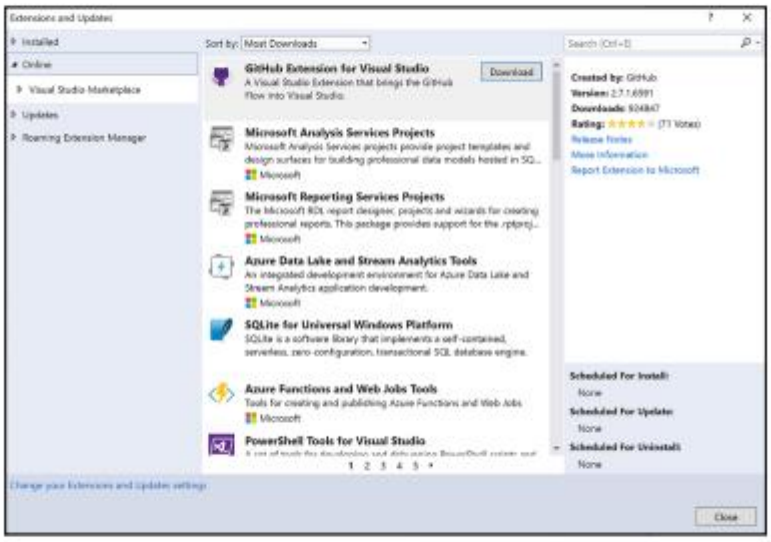
After you install Visual Studio, choose Tool⇨Extensions and Updates. A pop-up window appears with all the extensions you currently have installed, plus the marketplace of additional extensions. If you click Online, the top choice is the GitHub Extension for Visual Studio extension, as shown in Figure 13-5.

Click the Download button and close Visual Studio. When Visual Studio closes, the VSIX Installer starts and ask whether it can modify Visual Studio. Click Yes and Modify, and the extension begins to install. After it installs, click the Connect link in the Team Explorer tab and connect to your GitHub account, as shown in Figure 13-6. When you click the Connect link, a pop-up window asks you to sign in to GitHub. If you have two-factor authentication set up, you're also asked for your 2FA code.

Once connected, you can clone a repo, create a new repo, or sign out from GitHub, all from the Connect page on the Team Explorer pane.

Viewing, creating, and reviewing pull requests in Visual Studio

When you have the project open in Visual Studio that is connected to a GitHub repo, you see additional project options on the home page of the Team Explorer pane, as shown in Figure 13-7.


Extensions and Updates

Installed
Online
Visual Studio Marketplace
Updates
Roaming Extension Manager

Sort by: Most Downloads

GitHub Extension for Visual Studio
A Visual Studio Extension that brings the GitHub Flow into Visual Studio.

Microsoft Analysis Services Projects
Microsoft Analysis Services projects provide project templates and design surfaces for building professional data models hosted in SQ...
Microsoft

Microsoft Reporting Services Projects
The Microsoft RDL report designer, projects and wizards for creating professional reports. This package provides support for the .rptproj...
Microsoft

Azure Data Lake and Stream Analytics Tools
An integrated development environment for Azure Data Lake and Stream Analytics application development.
Microsoft

SQLite for Universal Windows Platform
SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

Azure Functions and Web Jobs Tools
Tools for creating and publishing Azure Functions and Web Jobs
Microsoft

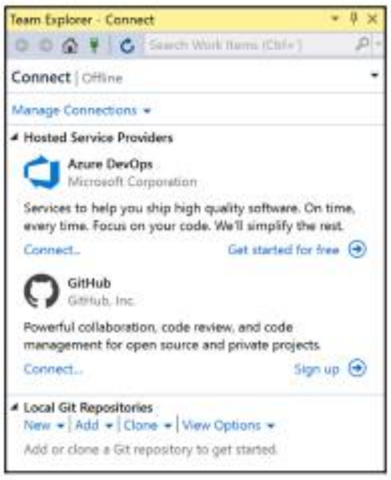
PowerShell Tools for Visual Studio

Created by: GitHub
Version: 2.7.1.6991
Downloads: 924847
Rating: (71 Votes)
Release Notes
More Information
Report Extension to Microsoft

Scheduled For Install:
None
Scheduled For Update:
None
Scheduled For Uninstall:
None

Change your Extensions and Updates settings

FIGURE 13-5: The GitHub for Visual Studio extension in the Visual Studio Marketplace.


Team Explorer - Connect

Connect | Offline

Manage Connections

Hosted Service Providers

Azure DevOps
Microsoft Corporation

Services to help you ship high quality software. On time, every time. Focus on your code. We'll simplify the rest.

Connect... Get started for free

GitHub
GitHub, Inc.

Powerful collaboration, code review, and code management for open source and private projects.

Connect... Sign up

Local Git Repositories
New | Add | Clone | View Options

Add or clone a Git repository to get started.

FIGURE 13-6: The Team Explorer pane with the GitHub Connect section.

Clicking Pull Requests opens the GitHub pane where all the pull requests on this repo are listed, as shown in Figure 13-8. At the top of the list, you can choose to see all the open pull requests, closed pull requests, or just all pull requests. You can also sort by author.

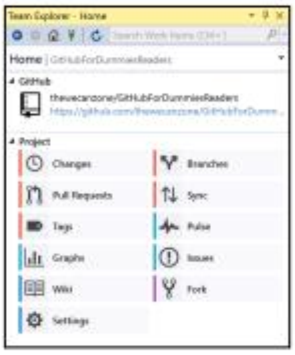


FIGURE 13-7: The Team Explorer pane with additional GitHub project options.

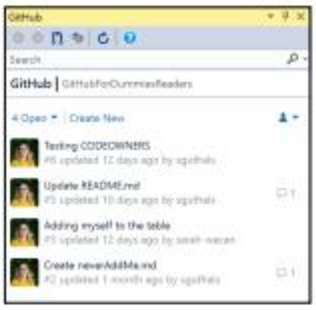


FIGURE 13-8: A list of pull requests on the GitHub pane.

If you double-click one of the pull requests, the details open. From here, you can see the description, the target and base branch, the current state, the list of reviews, and the list of files changed. You can also click the View on GitHub link to open a browser window to view this pull request on github.com. Click the Checkout <*branch-name*> button to check out the branch associated with this pull request. Click the Add Your Review link to add your own review with inline and overall comments. When you add a review, you can mark it as comment only, approve it, or request changes. If you double-click one of the changed files, the diff opens in the editor area. If you hover over one of the changed lines, you can add an inline comment, very similar to how it is done in VS Code (refer to Figure 13-4). You can see all of this in Figure 13-9.

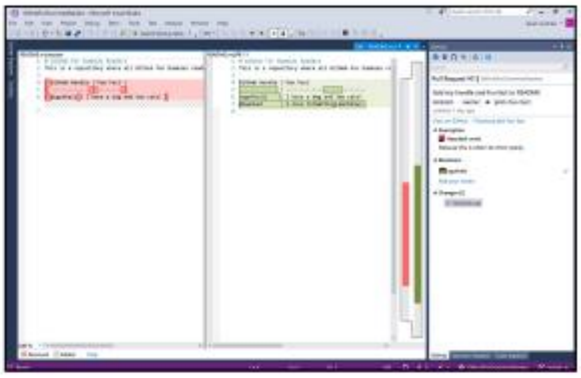


FIGURE 13-9: Interacting with a pull request in Visual Studio.

Following the GitHub for Visual Studio extension

The GitHub for Visual Studio extension is open source, so while it is highly recommended that you install and use the integrated Git experience for Visual Studio 2022, if you want to make improvements on the 2015, 2017, 2019 GitHub integration, you still can! Go to `https://github.com/github/visualstudio` to find documentation for how to use the GitHub for Visual Studio extension to improve your development workflow.

Using GitHub for XCode, Using GitHub for IntelliJ.

Personalizing GitHub - Using Browser Extensions, GitHub Apps and Probot, Taking Action with GitHub Actions.