INTERNAL USE ONLY

2015-06-04

# GitHub and Gerrit for Information Developers

# Content

# 1 Overview of GitHub

You are the information developer on a development team that is working on an open source application, and your project is due for release pretty soon. A Help icon has been added to the application, and you have created a PDF, and a web-help system, and you're pretty happy with yourself. So, you ask the developers where they want you to put your PDF and HTML files. They tell the project is being hosted on something called GitHub, and even give you a link to the folder on GitHub where they need your files. What now? That's they question you will be able to answer after you have read this document.

**What is GitHub?**

GitHub is free-to-use, publicly accessible library of open-source projects. It enables developers to upload the files that make up an project into a repository. Once in the repository, the files can be viewed and cloned by anybody else on the internet onto their own computers. Anyone can modify the cloned application if they want, and then submit the changes for integration to the original version, so that other users can take advantage of their bug fixes, enhancements, and new features. The owner of the repository can decide whether or not to accept or reject the changes. Before they make this decision, they of course have to be able to review the code, and most open source projects have integrated additional open source collaboration tools to GitHub to enable the to review submitted code before accepting it. Gerrit and NinjaReview are popular examples of these collaboration tools. The cloned version of the files for anything they want, once it does not breech the open source license agreement. There are various types of open source agreements, but generally they are not very restrictive.

# 2 GitHub Public Vs. GitHub Enterprise

Software projects hosted on GitHub are publicably available. Software projects can also be hosted on GitHub Enterprise for a fee, with the advantage that you can limit who can access your project. In other respects, the features and functions of GitHub and GitHub enterprise a similar.

# 3 Open Source Licenses

Open source projects on GitHub are generally subject to a licenense agreement. The owner of the product can choose from a variety fo license agreements available for open source projects ranging from very free to extremely restrictive.

**License Agreements**

The following are popular open source license agreement:

- GNU General Public License version 3 (http://gnu.org/licenses/gpl.htm l) - the most widely used free software license. It guarantees users, including organizations and companies the freedom to use, study, modify and share, the software, and stipulates that if the software is used in another piece of software, the rights to that piece of software are also covered by the GPL licnes
- Eclipse Public License (https://www.eclipse.org/legal/epl-v10.html ) -designed to be a business-friendly free software license and features weaker copyleft provisions than contemporary licenses such as the GNU General Public License (GPL). The receiver of EPL-licensed programs can use, modify, copy and distribute the work and modified versions, in some cases being obligated to release their own changes.
- Apache License (http://apache.org/licenses/LICENSE-2.0 ) - free software license that allows the user of the software the freedom to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software, under the terms of the license, without concern for royalties.
- MIT license (http://opensource.org/licenses/MIT ) - a free software license originating at the Massachusetts Institute of Technology (MIT). It is a permissive free software license, meaning that it permits reuse within proprietary software provided all copies of the licensed software include a copy of the MIT License terms and the copyright notice.

# 4 GitHub User Credentials

# 5 Make Changes on GitHub Repo

You can use GitHub Desktop to add, modify, and delete files in open source repos created by other GitHub users. The owener of the repo can decide whether or not to accept these changes.

To make changes on a GitHub repo that belongs to another GitHub user, do the following:

1. Sign up and log on to GitHub.
2. Download GitHub for Windows, which is an application that enables you to add, modify, and delete files in open source repos created by other GitHub users.
3. Configure GitHub for Windows settings and options so that it works with your GitHub account.
4. Search for a repo that was created by another GitHub user.
5. Use GitHub for Windows to clone the repo.
6. Use GitHub for Windows to create a local branch of the the repo.
7. Add, modify or delete files from the branch of the repo you have created.
8. Create a pull request, which is a request to the owner of the original, or master, version of the repo on GitHub to accept your changes.

The owner of the master version of the repo on GitHub now reviews your proposed changes, and either accepts or rejects them.

**Note**

Even if the owner of the master version of the repo rejects your changes, they have already been implement on your branch of your cloned version of the original.

## 5.1 Create a GitHub Account

In order to use GitHub, you must create a personal GiHub account on the GitHub website.

### Context

You create a GitHub account in order to be able to create new repos, or to clone and contribute to existing open source repos created by other GitHub users.

### Procedure

1. Enter https://github.com/ in the address bar of your internet browser.

The GitHub sign up page is displayed.

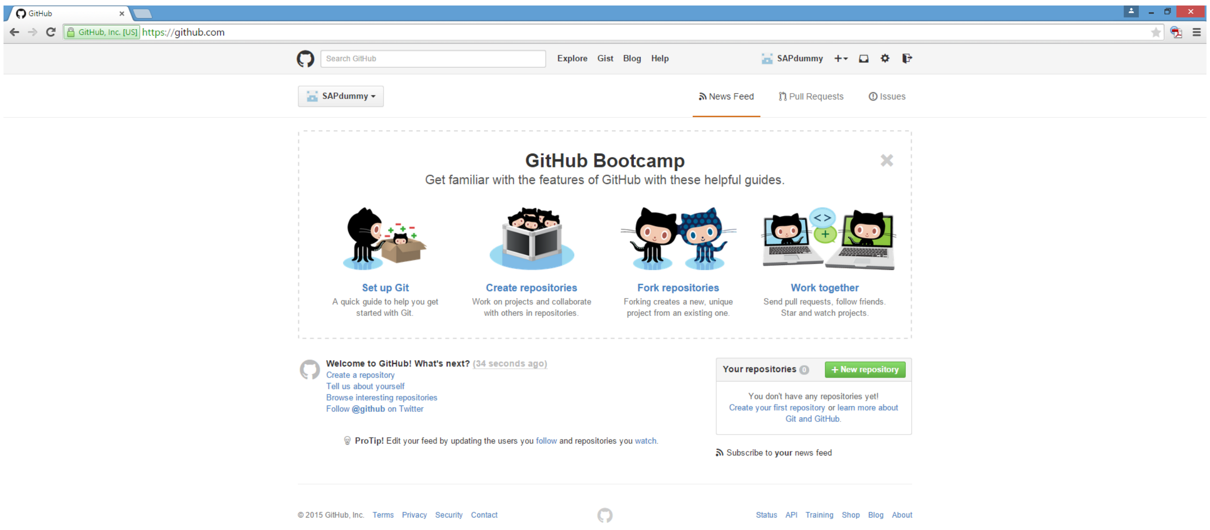
2. Enter a username, email, and password for your user account, and click the **Sign up for GitHub** button.

   The https://github.com/join/plan **Welcome to GitHub** page has been displayed.

3. Choose a plan by clicking the **Chosen** button.

   The features and functions of each plan is displayed in when you click each plan. If you do not intend on created a repo, but rather want to use your account to clone and contribute to exisiting open source repos created by other GitHub users, you can choose the **Free** plan.

   Your GitHub account is created and displayed.



GitHub Bootcamp

Get familiar with the features of GitHub with these helpful guides.

Set up Git
A quick guide to help you get started with Git.

Create repositories
Work on projects and collaborate with others in repositories.

Fork repositories
Forking creates a new, unique project from an existing one.

Work together
Send pull requests, follow friends. Star and watch projects.

**Welcome to GitHub! What's next?** (34 seconds ago)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow **@github** on Twitter

**ProTip!** Edit your feed by updating the users you follow and repositories you watch.

**Your repositories** 0 + New repository

You don't have any repositories yet!
Create your first repository or learn more about Git and GitHub.

Subscribe to **your** news feed

© 2015 GitHub, Inc. Terms Privacy Security Contact Status API Training Shop Blog About

## Results

You can now search for, clone, and contribute to exisiting open source repos created by other GitHub users.

## 5.2 Install and Configure GitHub for Windows

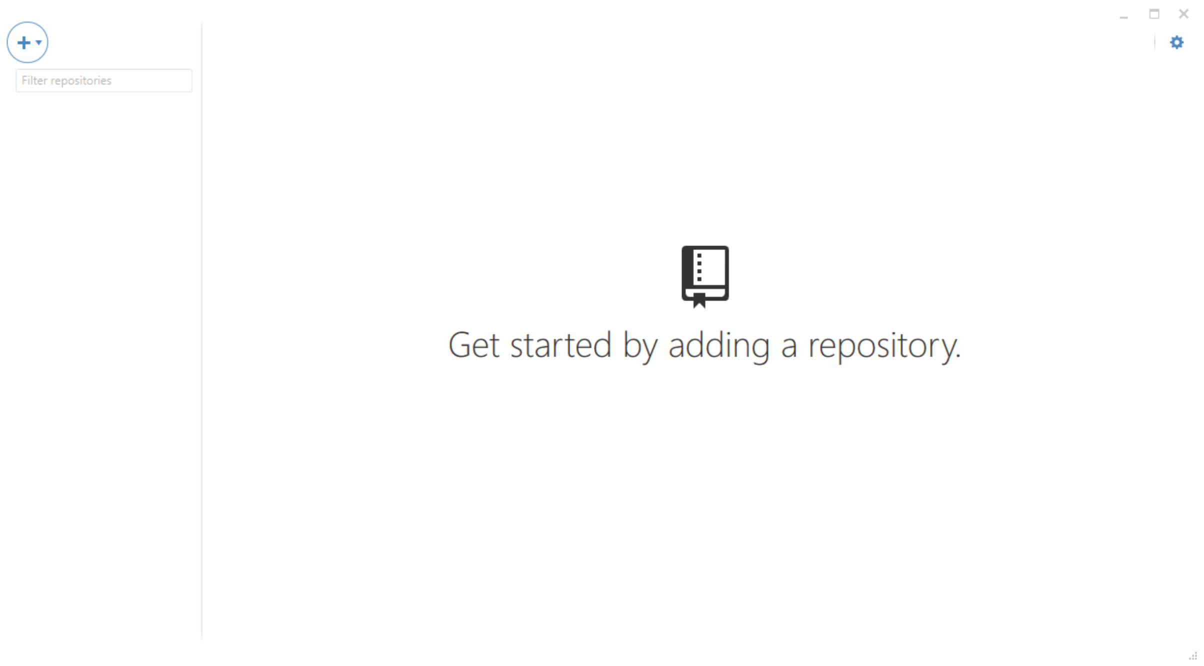
GitHub for Window is a free application that enables you to create, modify, and delete files on repos created and owed by other GitHub users.

### Procedure

1. Download GitHub from https://windows.github.com .
2. Click **Download GitHub for Windows**.

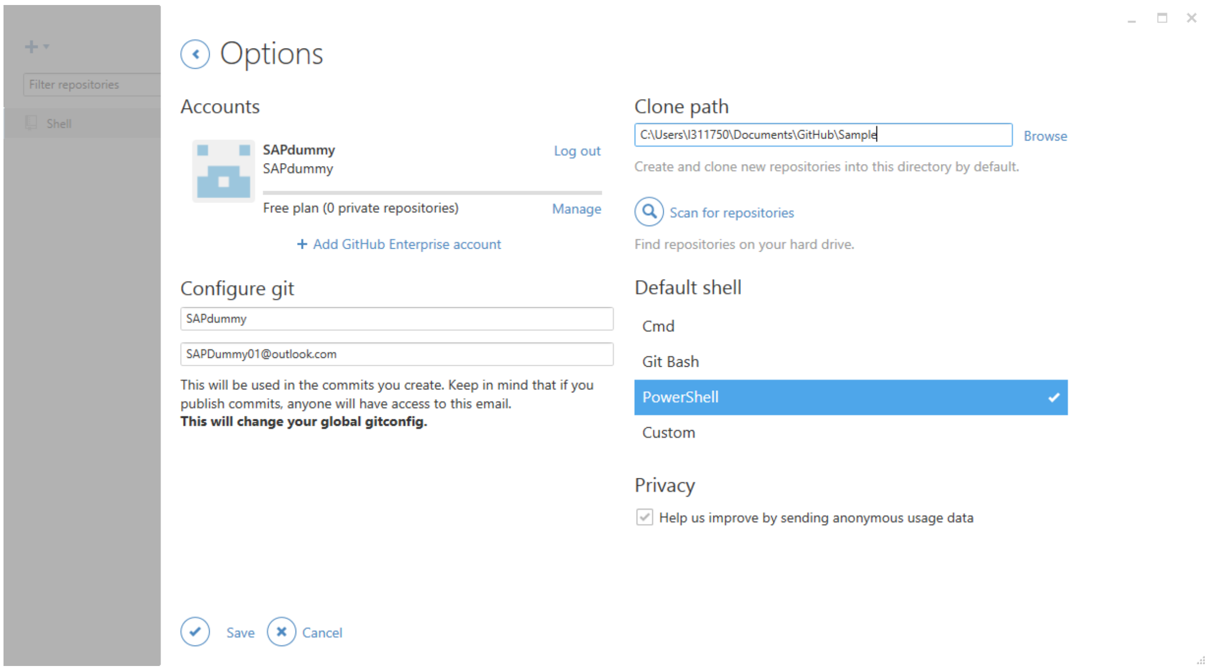
   The installation file, **GitHubSetup.exe**, is downloaded.
3. Double-click **GitHubSetup.exe**.

   **GitHub for Windows** is installed and displayed.



4. Click the **Settings** cog on the **GitHub for Windows** interface, and select the **Options** command.

   The **Options** page is displayed.
5. Complete the details on the **Options** page.



Accounts

SAPdummy
SAPdummy
Log out
Free plan (0 private repositories)
Manage
+ Add GitHub Enterprise account

Configure git

SAPdummy

SAPDummy01@outlook.com

This will be used in the commits you create. Keep in mind that if you publish commits, anyone will have access to this email.
**This will change your global gitconfig.**

Clone path

C:\Users\I311750\Documents\GitHub\Sample
Browse
Create and clone new repositories into this directory by default.

Scan for repositories
Find repositories on your hard drive.

Default shell

Cmd
Git Bash
[✓ PowerShell]
Custom

Privacy

✓ Help us improve by sending anonymous usage data

6. Click the **Save** button.

## Results

GitHub for Windows is now installed and ready to be used.

## 5.3 Clone a Repo

Before you can make changes to a repo, you must first clone a copy of the repo to your local computer enviroment.

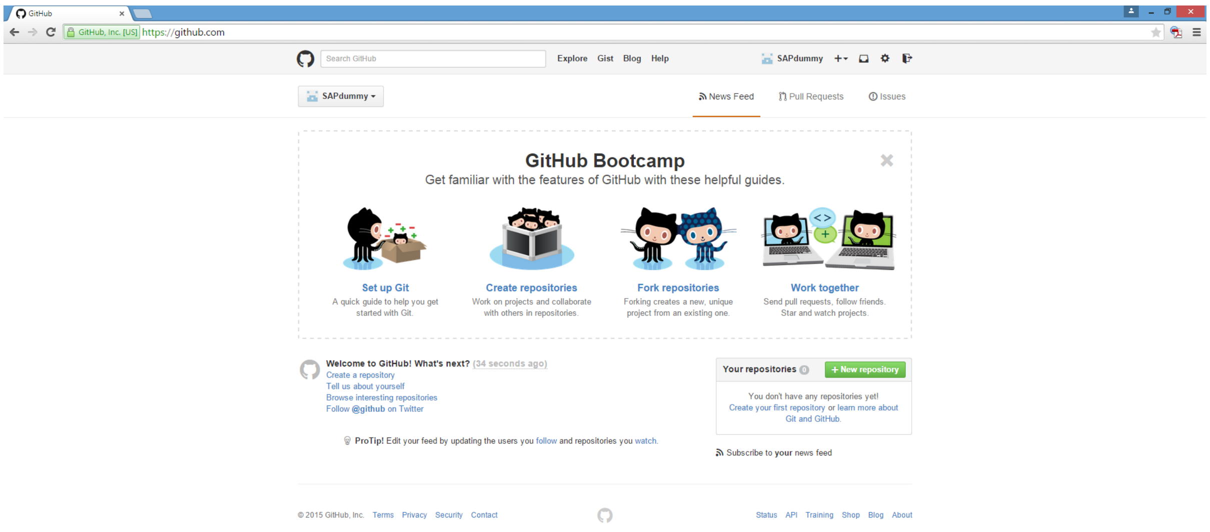
### Context

After you have signed up and logged on to GitHub, and installed GitHub for Windows, you can clone a repo to your local enviroment from GitHub.

### Procedure

1. Sign up and log in to GitHub.

The home page of your GitHub account is displayed.



GitHub Bootcamp

Get familiar with the features of GitHub with these helpful guides.

A quick guide to help you get started with Git.

Work on projects and collaborate with others in repositories.

Forking creates a new, unique project from an existing one.

Send pull requests, follow friends. Star and watch projects.

Welcome to GitHub! What's next? (34 seconds ago)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow @github on Twitter

ProTip! Edit your feed by updating the users you follow and repositories you watch.

Your repositories 0

You don't have any repositories yet! Create your first repository or learn more about Git and GitHub.

Subscribe to your news feed

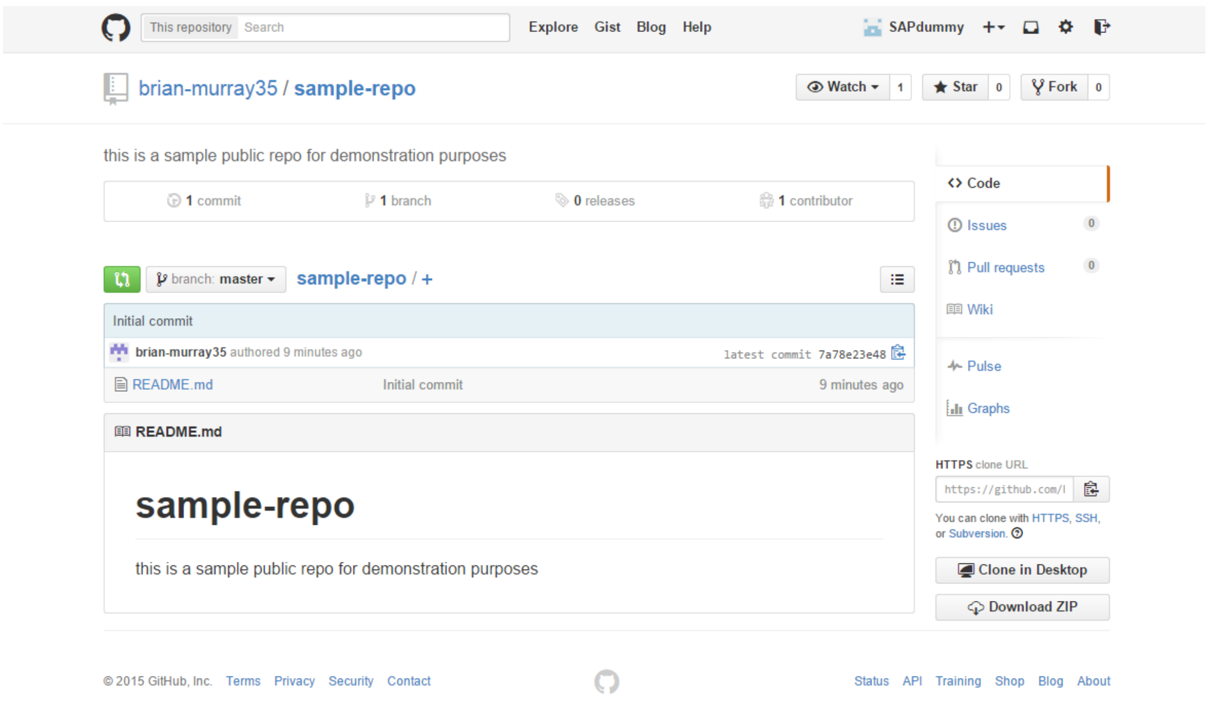
© 2015 GitHub, Inc. Terms Privacy Security Contact Status API Training Shop Blog About

2. Enter the name of the repo you want to clone in the **Search GitHub** search box and click **Enter**.

   A list of search results is displayed.

3. Click the repo you want to clone.

   The repo is displayed.



this is a sample public repo for demonstration purposes

1 commit | 1 branch | 0 releases | 1 contributor

Initial commit

brian-murray35 authored 9 minutes ago — latest commit 7a78e23e48

README.md — Initial commit — 9 minutes ago

README.md

sample-repo

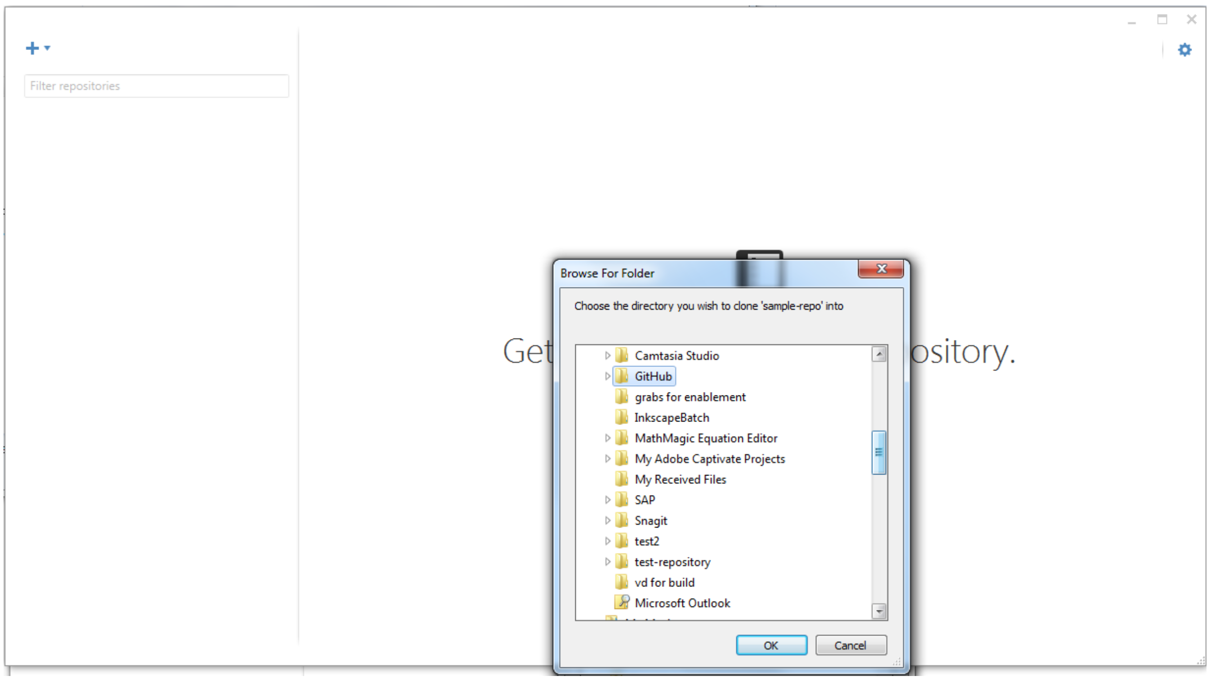
this is a sample public repo for demonstration purposes

You can clone with HTTPS, SSH, or Subversion.

© 2015 GitHub, Inc. Terms Privacy Security Contact Status API Training Shop Blog About

4. Click the **Clone in Desktop** button.

GitHub for Windows is displayed, and a dialog requiring you to create or identify the location on your local enviroment where you want to clone a copy of the repository is displayed.



Filter repositories

Browse For Folder

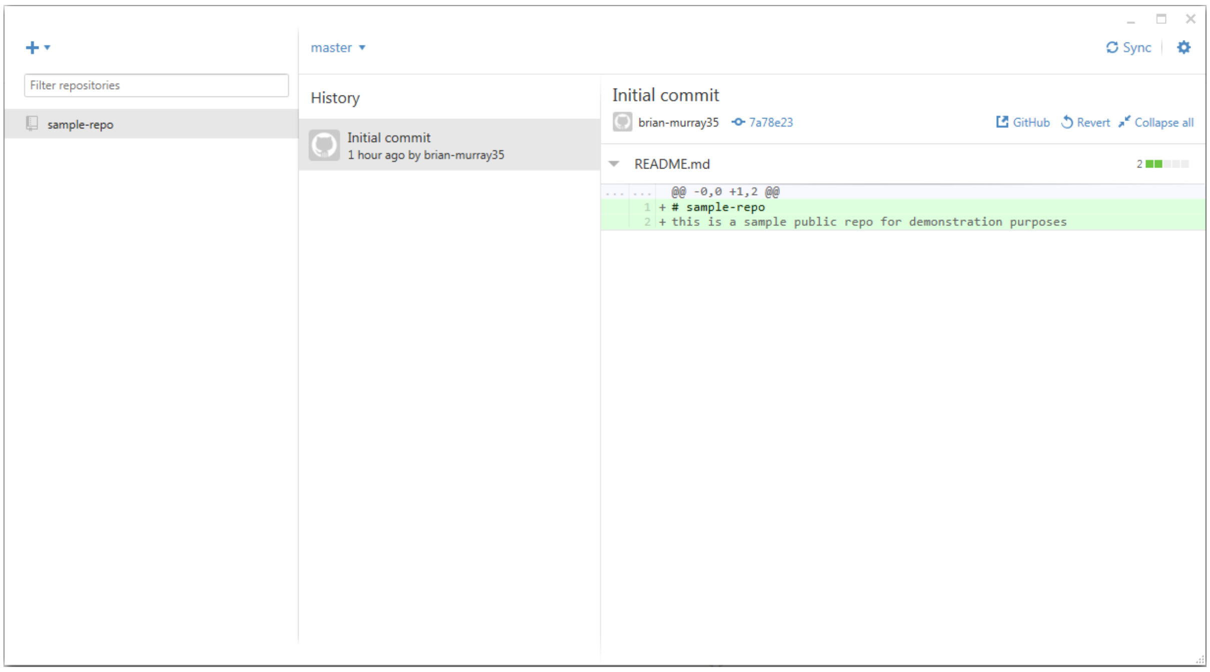
Choose the directory you wish to clone 'sample-repo' into

- Camtasia Studio
- GitHub
- grabs for enablement
- InkscapeBatch
- MathMagic Equation Editor
- My Adobe Captivate Projects
- My Received Files
- SAP
- Snagit
- test2
- test-repository
- vd for build
- Microsoft Outlook

OK Cancel

5. Browse to the location on your local computer where you want to clone the repo and click **Open**.

   The repo is cloned and displayed in the left-hand panel.



Filter repositories

sample-repo

master

History

Initial commit
1 hour ago by brian-murray35

Initial commit

brian-murray35 7a78e23

README.md 2

```
@@ -0,0 +1,2 @@
+ # sample-repo
+ this is a sample public repo for demonstration purposes
```

### Results

You can now create a branch from the cloned repo. You can then make changes to the branch in your local environment and submit these changes to be integrated into the the version on GitHub.

## 5.3.1 Troubleshooting: Cloning a Repo

In **GitHub for Desktop**, it may happen that clicking the **Clone in Desktop** button in a repo you want to clone does not work, do the following:

1. Ensure you are using Google Chrome. If this does not resolved the problem, try...
2. If you are connected to the internet behind a firewall (e.g. SAP Corporate), try another internet connection (e.g. SAP Internet). If this does not resolve the problem, try...
3. Drag the URL of the repo you want to clone from the address bar of your browser and drop it onto onto the **GitHub for Desktop** interface. This should automatically being the cloning process...

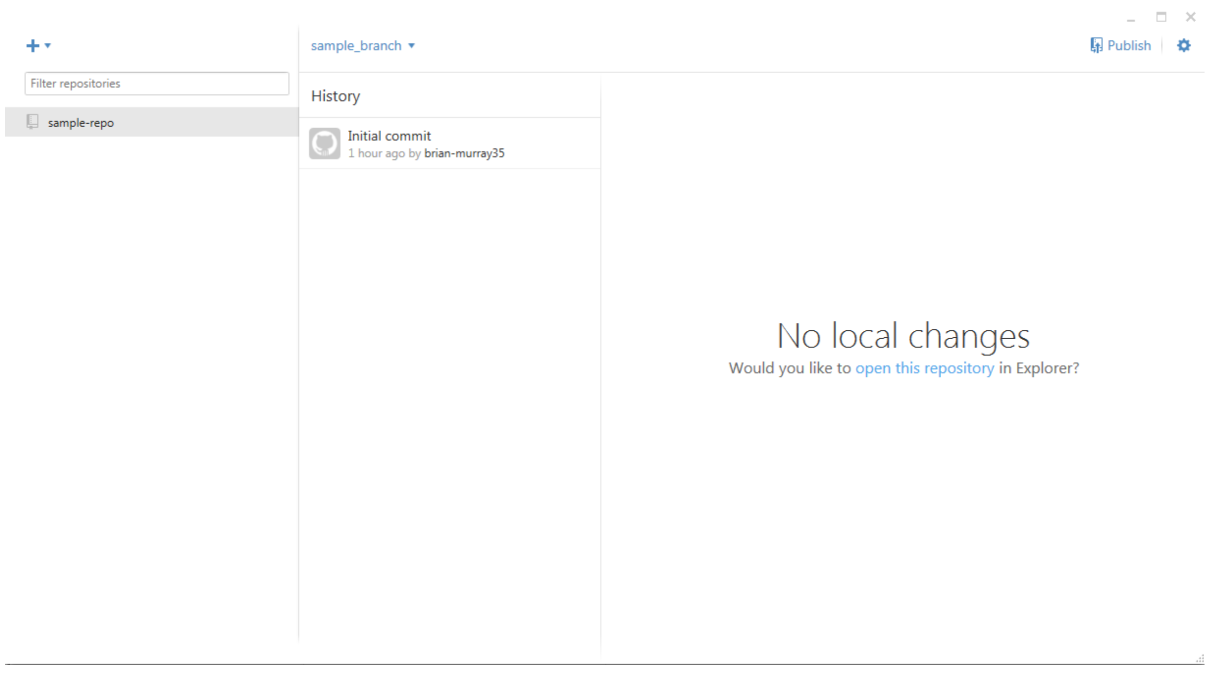
## 5.4 Create a Local Branch of the Repo

You can create a unique branch of the cloned repo, make changes, and create a pull request to add your changes to the repo on GitHub.

### Procedure

1. Click the repo in the left-hand panel on the **GitHub for Windows** interface.
2. Click the branch drop-down arrow for the repo and enter a name for a new repo.
3. Enter a name for your branch and press Enter.

   The branch is created.



### Results

You can now select which branch is displayed at any time by clicking the branch downwards arrow and selecting a branch. Each branch is a unique version of the cloned repo, so you can implement different changes on different branches if you want.

## 5.5 Add, Modify, or Delete Files from a Repo

You can add, modify, or delete files in a branch of a cloned repo in GitHub. Then you can submit a pull request to have your changes merged into the master branh of the version of the repo on GitHub.

### Context

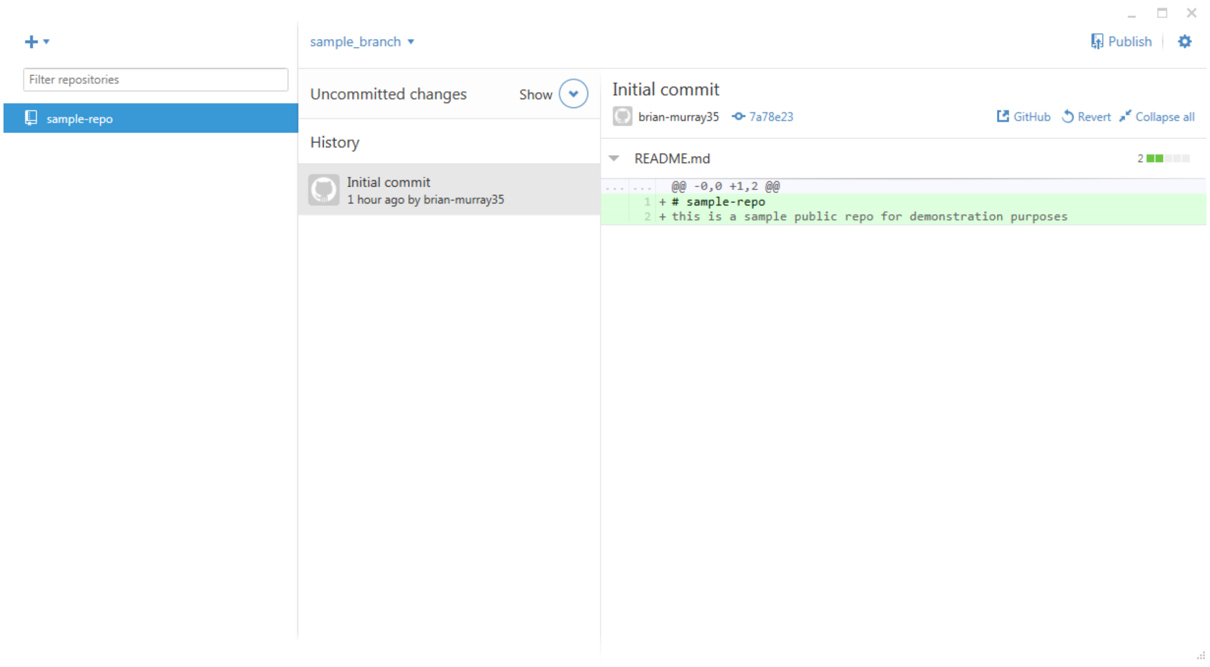
Before you make changes to a repo, you must have cloned the repo to your local environment using GitHub for Windows, and created a branch from the cloned version.

## Procedure

1. Click the branch drop-down on GitHub for Windows and select the branch you have created.
2. Right-click the repo in the right-hand panel and select **Open in Explorer**.

   The directory structure of the repo is displayed in Windows Explorer.
3. Add, modify or delete files as required.

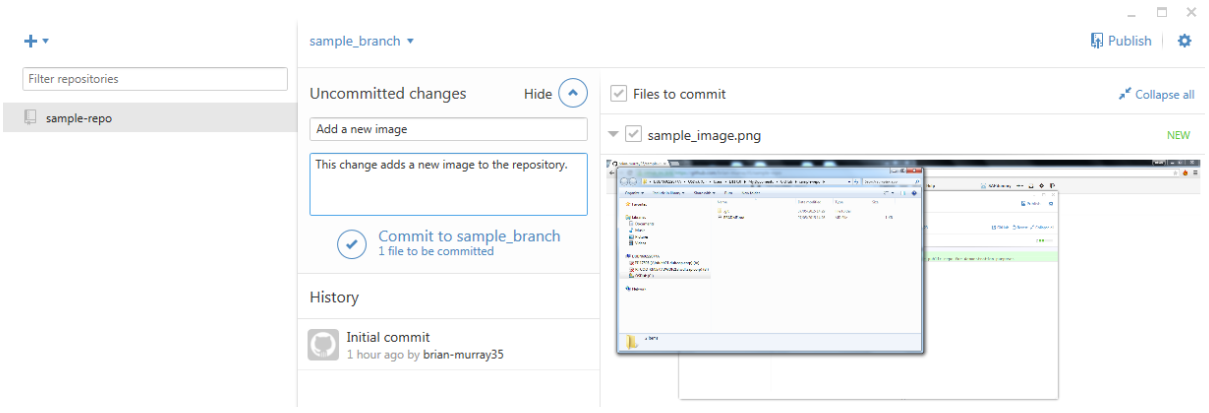
   In GitHub for Windows, an **Uncommitted changes** messages is displayed.



```
@@ -0,0 +1,2 @@
+ # sample-repo
+ this is a sample public repo for demonstration purposes
```

4. Click the **Show** arrow on the **Uncommitted changes** message.

   The name and description fields for the change are displayed.



5.
6. Enter a name a description for your change.

   This information is presented to the reviewer of the pull request in GitHub, so ensure that they describe the purpose of the pull request clearly. Each project has different guidelines for writing these names and descriptions. However, you will have the opportunity to modify this text before the pull request is created.

7. Click the **Commit to [***branch***]** button.

   The change is committed.

### Results

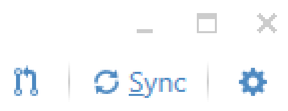
Once you have made your changes on the branch of the cloned version of the repo in GitHub desktop, you can submit a pull request to the repo on GitHub to have your changes merged with the master branh of the version of the repo on GitHub.

## 5.6 Create a Pull Request in GitHub for Windows

You can create a pull request in GitHub for Windows or on www.GitHub.com.

### Context

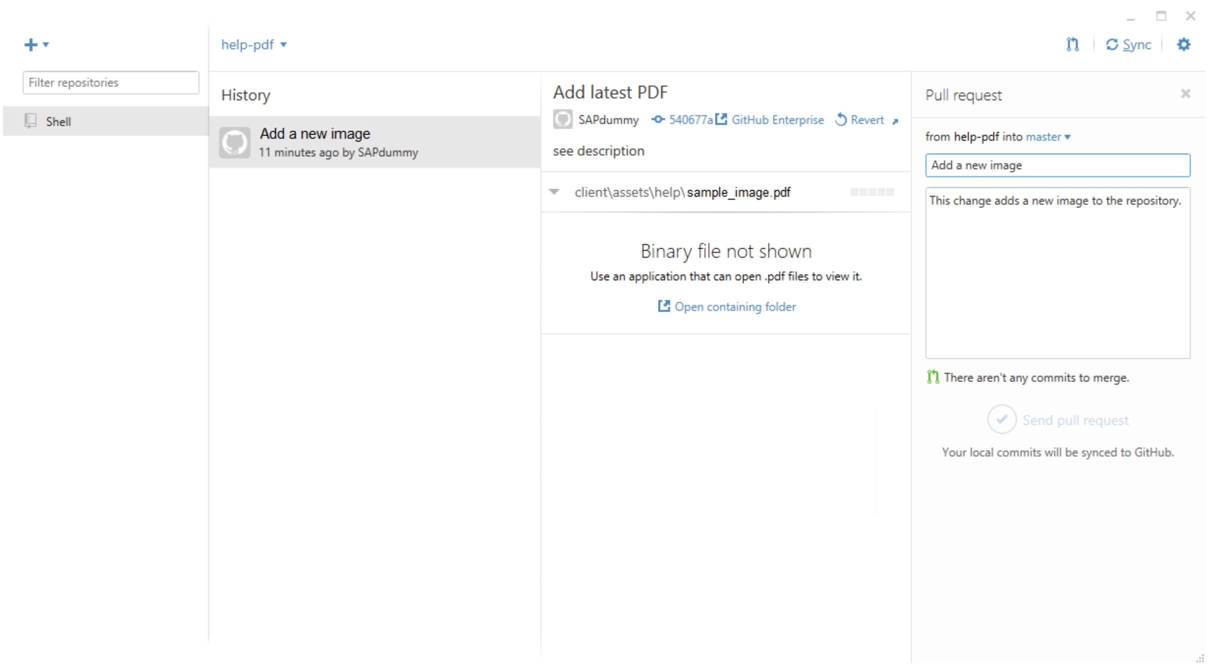
When you have commited changes to a repo in GitHub for Windows, the **Pull Request** icon is displayed in the top-right of the window.



### Procedure

1. In GitHub for Windows click the **Pull Request** icon.

   The **Pull Request** panel is displayed.



Add a new image
11 minutes ago by SAPdummy

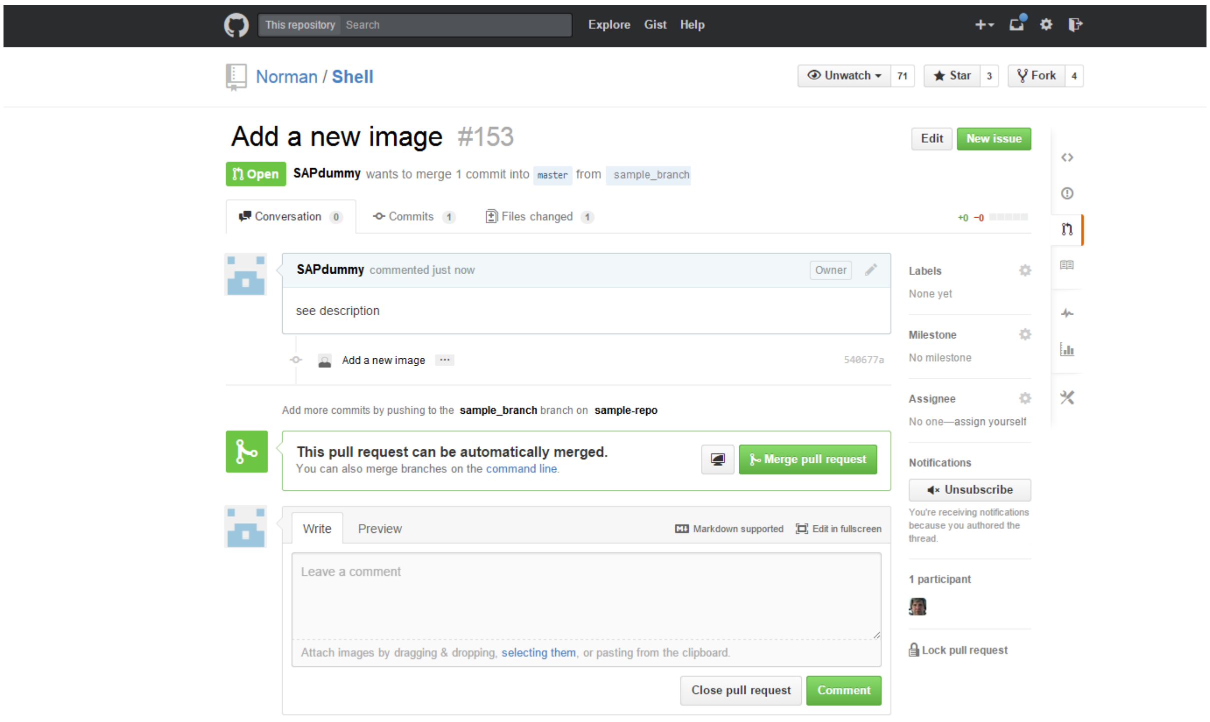
Add latest PDF
SAPdummy 540677a GitHub Enterprise Revert
see description
client\assets\help\sample_image.pdf
Binary file not shown
Use an application that can open .pdf files to view it.
Open containing folder

from help-pdf into master
Add a new image
This change adds a new image to the repository.
There aren't any commits to merge.
Send pull request
Your local commits will be synced to GitHub.

2. Select the branch into which you want merge your changes by clicking the drop-down arrow abover the **Name** field in the **Pull Request** panel, and select a branch. In this example, the **master** branch is selected by default.
3. Enter a name and description for the pull request.

   The name and description you entered for the change when you were committing it is displayed.
4. Click the **Send pull request** button.

   The pull request is created in GitHub.



Add a new image #153
Open SAPdummy wants to merge 1 commit into master from sample_branch

SAPdummy commented just now
see description
Add a new image 540677a

Add more commits by pushing to the sample_branch branch on sample-repo

This pull request can be automatically merged.
You can also merge branches on the command line.

You're receiving notifications because you authored the thread.

Leave a comment
Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

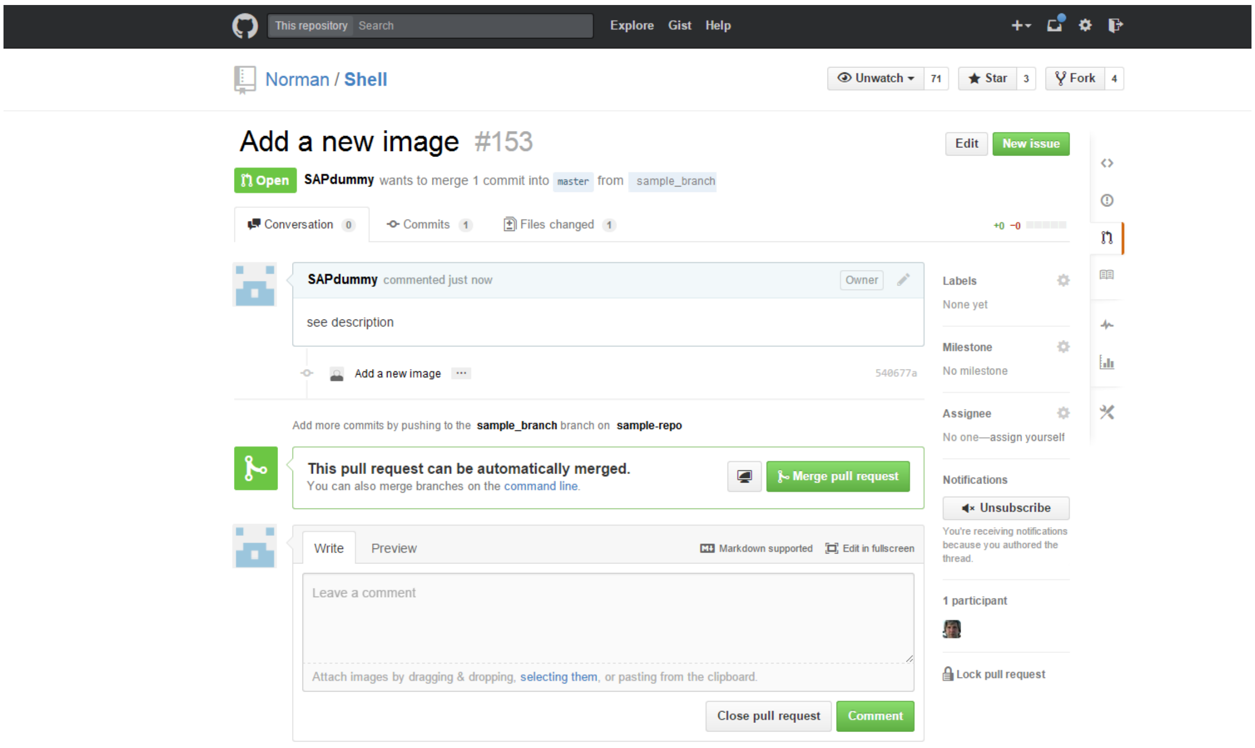
## 5.7 Merge a Pull Request in Git Hub

When you, or another user, have created a pull request to have changes included in a repo on GitHub, GitHub users with the appropriate credentials can merge or reject the changes.

### Context

When a pull request is created, it is sent to the repo on GitHub, and from there it can be merged merged a GitHub user with the approriate credentials. However, in some environments where Gerrit code reviews are required, pull requests can only be merged when the code review is sucessful. For more information about Gerrit code reviews, see Overview of Gerrit. [page 23]

When a pull request has been created, you see the following:



Add a new image #153

SAPdummy wants to merge 1 commit into master from sample_branch

SAPdummy commented just now

see description

Add a new image 540677a

Add more commits by pushing to the sample_branch branch on sample-repo

This pull request can be automatically merged.
You can also merge branches on the command line.

You're receiving notifications because you authored the thread.

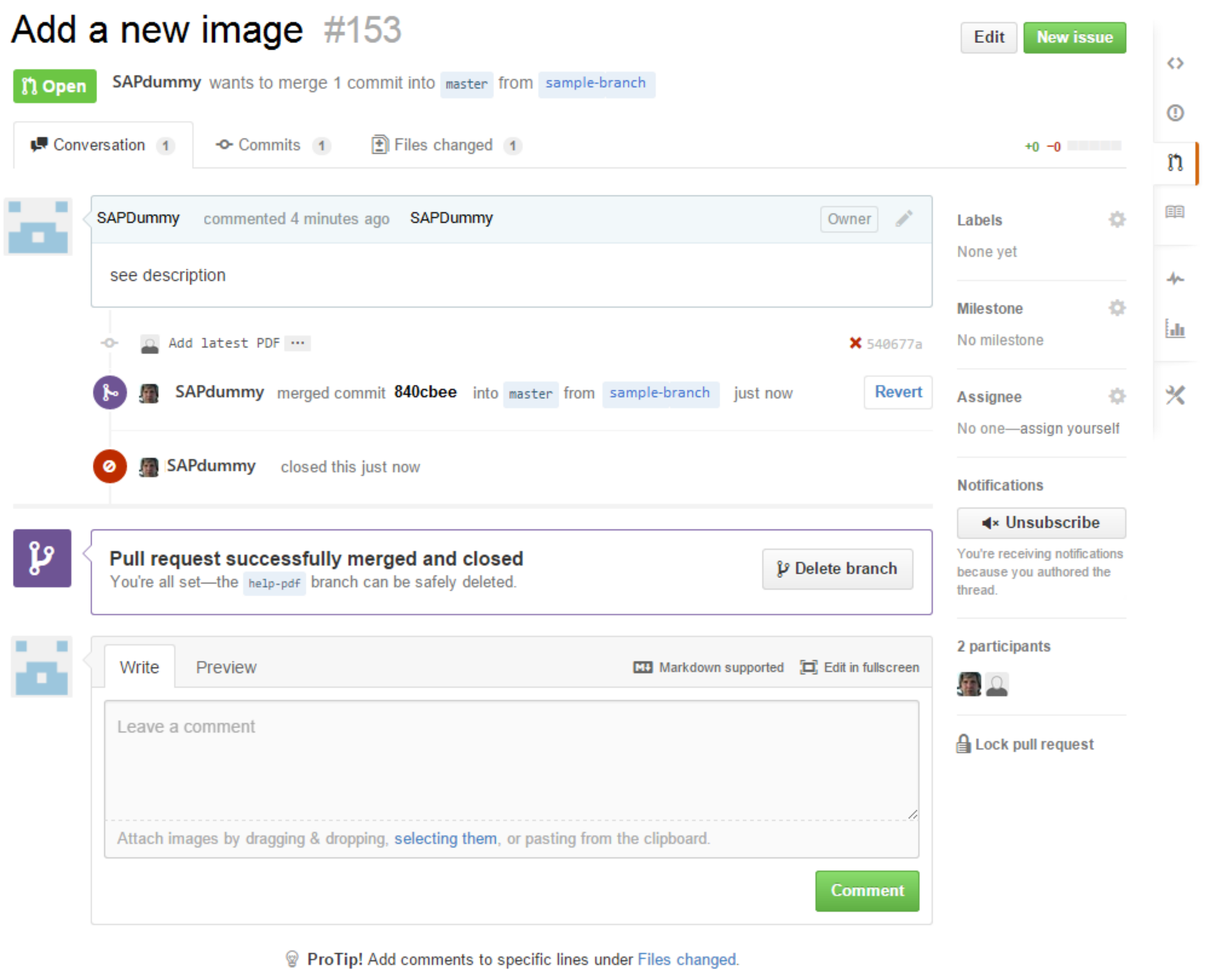
Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

In this instance, a user with the approriate user credentials (usually the owner)can merge the pull request by doing the following:

### Procedure

Click the **Merge and Pull Request** button.

The pull request is meged:



# Add a new image #153

Edit | New issue

Open SAPdummy wants to merge 1 commit into `master` from `sample-branch`

Conversation 1 | Commits 1 | Files changed 1

+0 −0

SAPDummy commented 4 minutes ago SAPDummy Owner

see description

`Add latest PDF` ... 540677a

SAPdummy merged commit **840cbee** into `master` from `sample-branch` just now Revert

SAPdummy closed this just now

**Pull request successfully merged and closed**

You're all set—the `help-pdf` branch can be safely deleted.

Delete branch

Write | Preview | Markdown supported | Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

Comment

**ProTip!** Add comments to specific lines under Files changed.

Labels: None yet

Milestone: No milestone

Assignee: No one—assign yourself

Notifications: Unsubscribe

You're receiving notifications because you authored the thread.

2 participants

Lock pull request

## 5.8 Ask Questions, Create Bugs and Feature Requests in GitHub

In addition to creating pull requests, you can use the issue tracker in a GitHub repo to ask questions, create bugs and feature requests to the owners of the repo. This is possible on a GitHub.com account, and on GitHub Enterprise accounts.

### Context

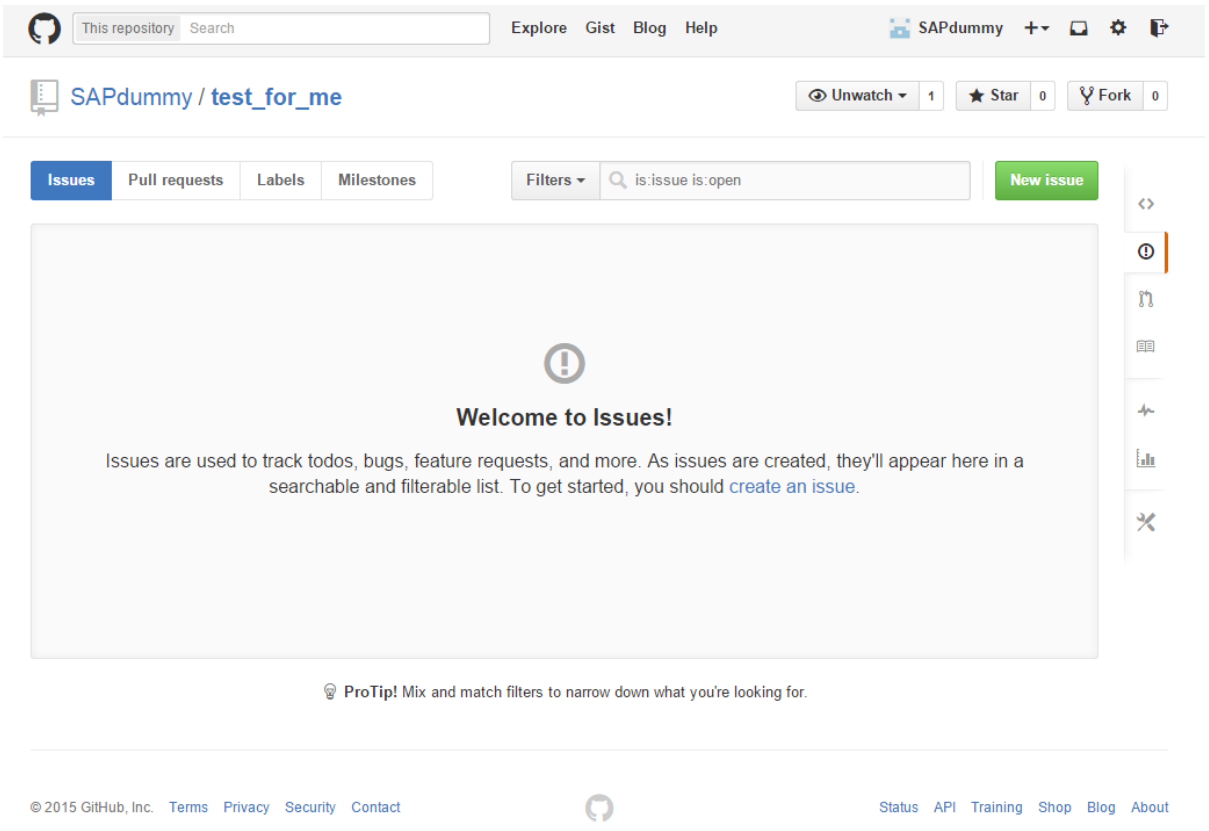
To create an issue in a repo in GitHub.com, do the following:

### Procedure

1. Sign into GitHub using a private account or an enterprise account.
2. Search for the repo by entering the repo name in the **Search** text box.
3. Select the repo.

   The repo is displayed.
4. Click the Issues button on the right-hand side.

   The **Issue Tracker** is displayed.



Welcome to Issues!

Issues are used to track todos, bugs, feature requests, and more. As issues are created, they'll appear here in a searchable and filterable list. To get started, you should create an issue.

**ProTip!** Mix and match filters to narrow down what you're looking for.

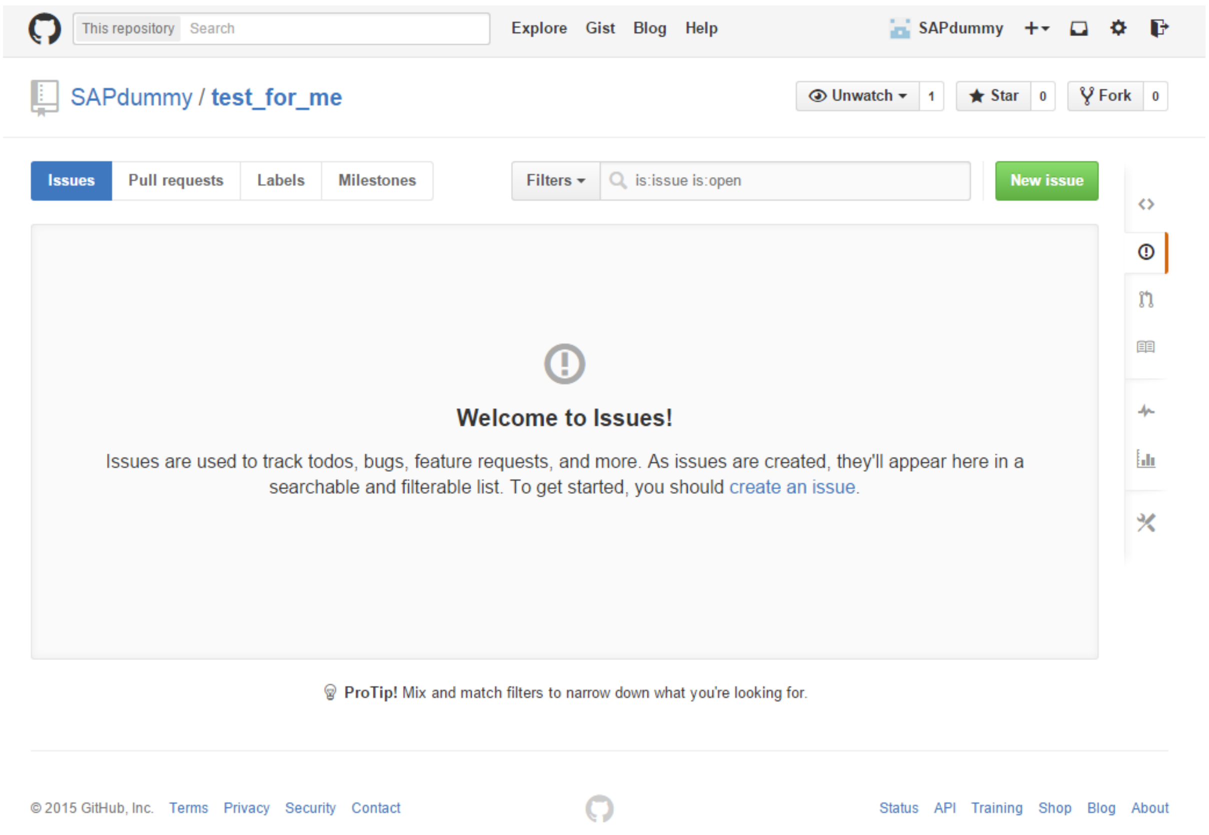
© 2015 GitHub, Inc. Terms Privacy Security Contact
Status API Training Shop Blog About

5. A new issue is displayed.

   The number of this issues is displayed in the format #[*Issue Number*].
6. Enter a name and description for the issue, and click the **Submit new issue** button.

   The issue is created. Now other users of GitHub can access the issue.

7. You, or another user with appropriate permissions, can set a label(s) for the bug, by clicking the



**Welcome to Issues!**

Issues are used to track todos, bugs, feature requests, and more. As issues are created, they'll appear here in a searchable and filterable list. To get started, you should create an issue.

**ProTip!** Mix and match filters to narrow down what you're looking for.

© 2015 GitHub, Inc. Terms Privacy Security Contact Status API Training Shop Blog About

button, and selecting the relevent label(s).

The label is assigned to the issue. Labelling issues makes it easier for the repo owners to group, filter, and manage the issues. You can create labels by clicking the Issues button on the right-hand side of the repo, and clicking the **Labels** button.

8. You, or other users, can enter comments in response to the issue, or close the issue, as required.

## Related Information

GitHub Public Vs. GitHub Enterprise [page 4]

# 6 Overview of Gerrit

• Gerrit is a Git server which allows management of multiple repositories in one central system

• Gerrit provides access authorization to each repository (even on branch level)Ge

• Gerrit facilitates the code review process (four-eyes principle) which is very common in OpenSource communities

The central LeanDI Gerrit Server can be reached at https://git.wdf.sap.corp:8080 (bad naming!)

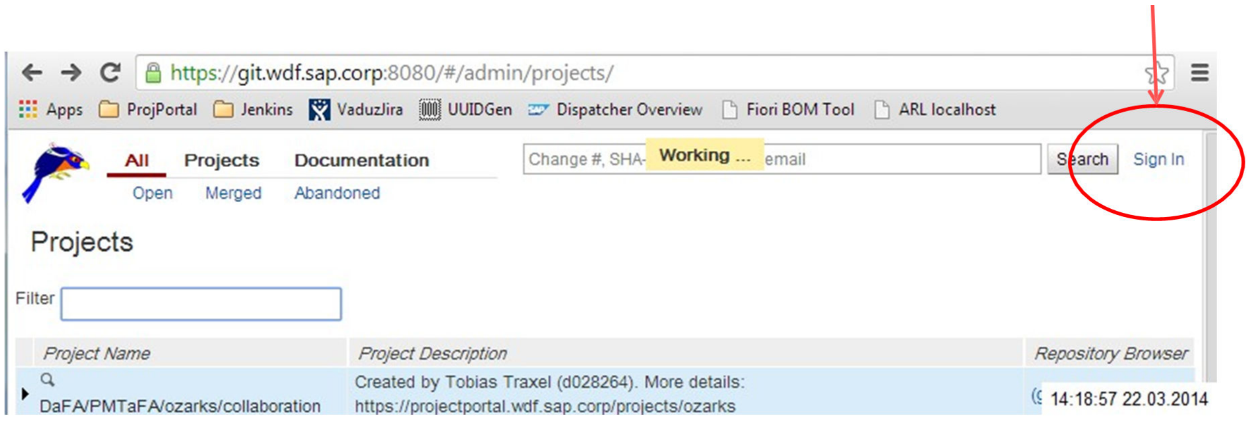
Logon with your SAP_ALL/GLOBAL account (passwords are being synchronized automatically)

## 6.1 Common Pitfalls

Performance of table rendering in Internet Explorer (IE) is very bad.

If you go to https://git.wdf.sap.corp:8080 manually, you are typically not signed it automatically. You have to do so manually.

If you are not doing so, still the menu looks like as if you were logged on, but you cannot find your projects/ changes/… □□confusing



| Project Name | Project Description | Repository Browser |
|---|---|---|
| DaFA/PMTaFA/ozarks/collaboration | Created by Tobias Traxel (d028264). More details: https://projectportal.wdf.sap.corp/projects/ozarks | |

14:18:57 22.03.2014

As there is automatic type-ahead on searching, using Gerrit with IE may be sluggish and cumbersome. Therefore, you should use an alternative browser, such as Chrome or Firefox when using Gerrit.

## 6.2 Code Reviews

A coder is implementing changes to a project, but is not allowed to ship it to the customer immediately.

Before that peers are looking into the changes and signing it off.

Being a good code reviewer is another skill than being a good coder

**Environment**

- non-threating style
- Goal is cooperation, not fault-finding

**Benefits**

- Catches bugs early (even before they reached the mainline)
- Distribution of knowledge across all team members
- Enforces consistency

## 6.3 Gerrit Roles

**Administrator**: A user who is allowed to grant all authorizations to a third person (only LeanDI system administrators)

**Owner**: A user who has the “owner” privilege which allows to grant/revoke authorizations for the git repository; also allowed to create/delete branches,

**Committer**: A user who has the “submit” privilege which allows to approve (i.e. submit) a review – typically this is the list of team members

**Pusher**: A user who has the “push” privilege which allows to bypass the review process (not something what you typically want)

**General**: Every user (i.e. everyone in SAP_ALL/GLOBAL) is entitled to send a change to Gerrit for review!

## 6.4 Voting System - Verified and Code Review

Table 1:

| Reviewer | Verified | Code Review |
|---|---|---|
| **Reviewer A** | Voting for “technical verification”, i.e. correctness of syntax, the change can be integrated, doesn’t break any Unit Tests<br>• typically automated<br>• “fiorivoter” (Jenkins) Requirement for submission: “+1” | Voting for “semantical verification”, i.e. semantic consistency, checking of read-ability/comprehension, comments in the coding,<br>• manual process<br>Requirement for submission: “+2”<br>The number of reviewers is not limited and reviewers can be added at any time |
| **Reviewer B** | Voting for “technical verification”, i.e. correctness of syntax, the change can be integrated, doesn’t break any Unit Tests<br>• typically automated<br>• “fiorivoter” (Jenkins) Requirement for submission: “+1” | Voting for “semantical verification”, i.e. semantic consistency, checking of read-ability/comprehension, comments in the coding,<br>• manual process<br>Requirement for submission: “+2”<br>The number of reviewers is not limited and reviewers can be added at any time |

## 6.5 Voting System - Rules of Engagement

For “Verified”, the reviewer may vote

- +1 - I have tested it and think that it’s good
- 0 - no opinion
- -1 - test fails, please do not submit
  For “Code Review”, the reviewer may vote
  +2 - I agree to this change and want to allow submitting it (only committers are allowed to do that)
  +1 - I agree to this change (I do not have any objections, any reviewer may do so) • 0 □□no opinion so far (or undecided)
  -1 - This change isn’t ripe to be included yet
  -2 - veto, fundamental disagree (only committers are allowed to do that) change is prohibited from being submitted in any case

> **Note**
>
> Warning! 1 + 1 does not make +2!

There must be at least one committer with +2 before the change may be submitted.

## 6.6 Voting System in Open Source Projects

It is very common that you will first get a “code review -1” for your change!

This is not a rejection, but only an indication that something should be fixed.

A “code review -2” is a clear indication that the change will be rejected altogether due to fundamental objections.

- Contact the person directly for clarification.

Most projects do not apply the “four-eyes principle”, but at least “six-eyes”

- The first reviewer will only cast a vote “code review +1” the second reviewer then will cast “code review +2”, if the change is ok.

## 6.7 Integration Voter / Jenkins

Voting for “Verified” is typically automated.

For this, the integration server (LeanDI/Fiori: Jenkins) is used on which a job with several checks is executed to verify the consistency of the build. Typically this includes:

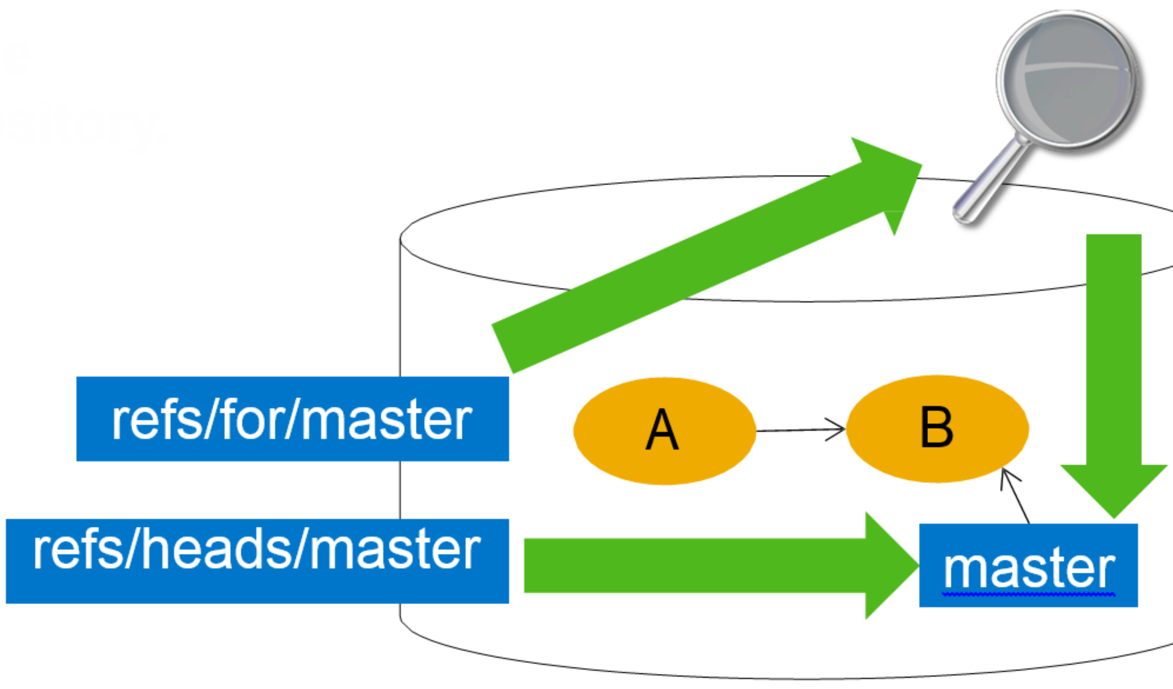
- Running Unit Tests (QUnit, JsTestDriver, ...)
- - Code Scanning Checks (jsLint)

If there is at least one major error, the automated voter will cast a vote of “Verified -1”. If no major error was found, it will vote for “Verified +1”.

An automated voter always will only vote for “Verified”, but never for “code review” (will leave it on “0”).

## 6.8 Integration of Git and Gerrit

Technically, the Gerrit is a remote repository for your local git repository.



**Speciality**

For each physical branch in the repository, there are two virtual branches which serve as interfaces:

- the push branch: refs/heads/master (a.k.a. master)
    - a **push** will go directly into the repository (bypassing code review)
    - user must be a “pusher”, i.e. push authorization is required
- the **code review** branch: refs/for/master
    - a push will trigger a code review process (or update an existing one)
    - any Gerrit user can do it

## 6.9 Typical Error

If you are getting an error message like:

```
! [rejected]    HEAD -> master (non-fast-forward)
error: failed to push some refs to
'ssh://xxxxx@git.wdf.sap.corp:29418/fnf/QD/your_project.git'
hint: Updates were rejected because a pushed branch tip is behind its remote hint:
counterpart. Check out this branch and integrate the remote changes hint: (e.g.
'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

then you might have tried to push to “refs/heads/master” instead of “refs/for/master”, and you do not have the “push” authorization.

Note

Check your branch setup to see to which branch at the remote repository it tries to push your changes

## 6.10 Change ID in Commit Message

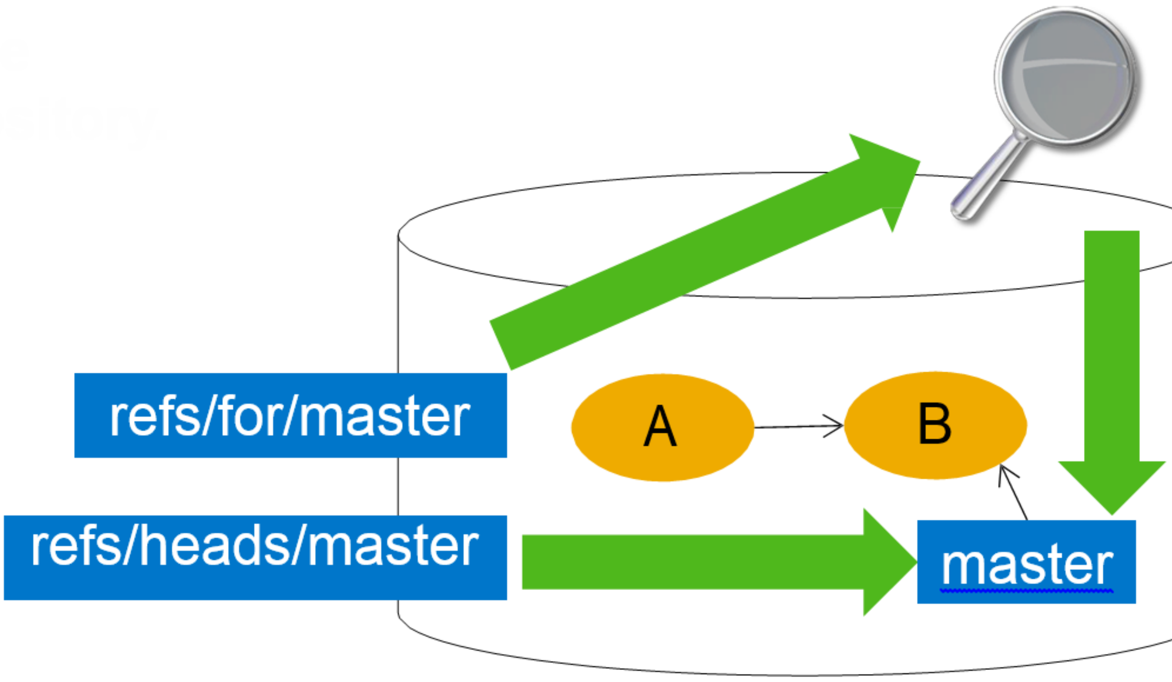
Commit which are pushed to a refs/for/... branch require a Change ID in their commit message.

Commits without Change-IDs will result in an error message on pushing

Note

If you know that you want to push a commit later on, always add a Change-ID immediately; adding it later on may become cumbersome!

Ensure to have it pressed to insert Change-ID



Here: Dummy change-ID - will be generated on 'Commit'

## 6.11 PatchSet

For a given code review in Gerrit, multiple versions (i.e. implemeting corrections during the code review) can exist.

These iterations are called PatchSets.

PatchSets are created automatically by pushing a new version of the same correction.

This is detected by having the same “Change ID” in the commit message.

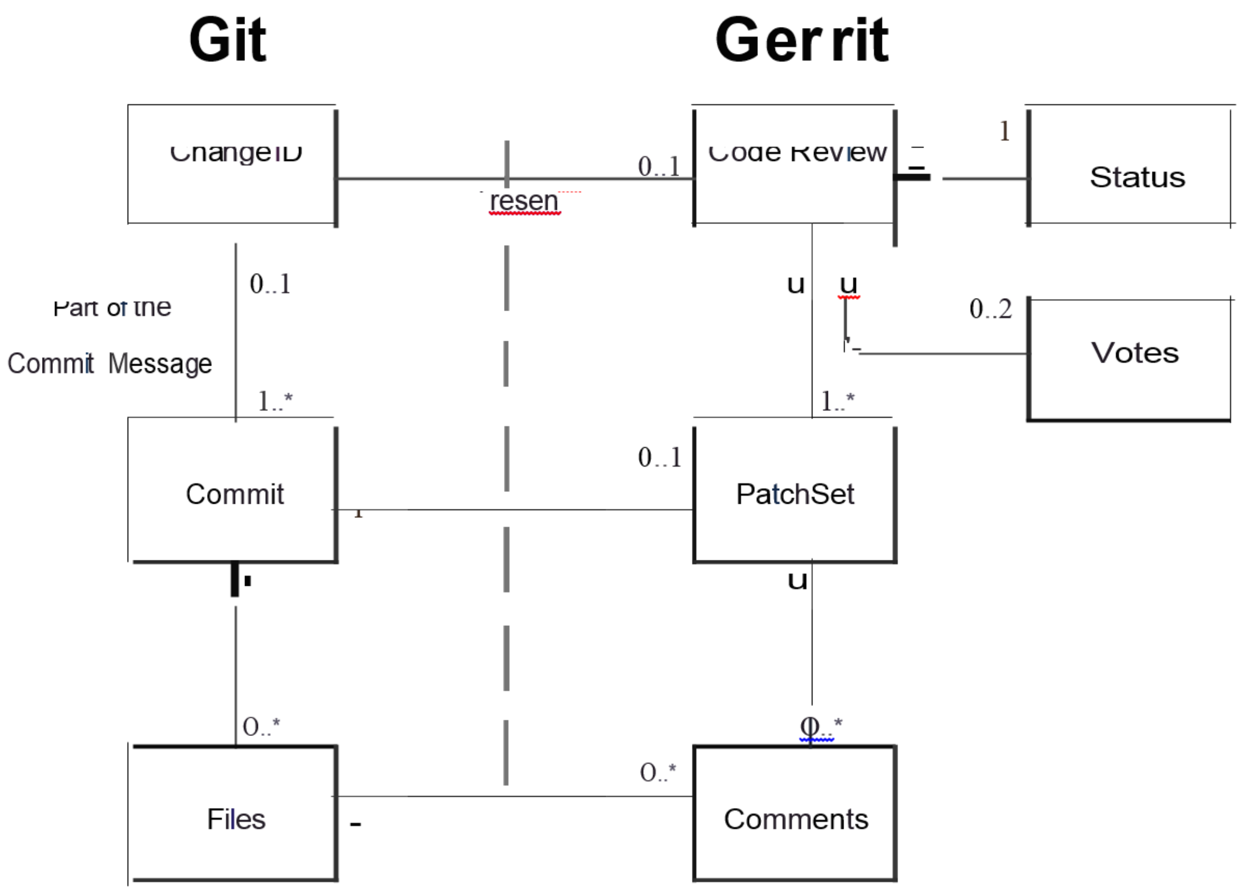
Note

If correcting your code review change, ensure that your ChangeID remains stable!

- Use “Rebase” for correcting your code review

Only the last PatchSet of a code review counts for the submission. All others are just for historical reasons (traceability).

## 6.12 Mapping of Entities Git / Gerrit



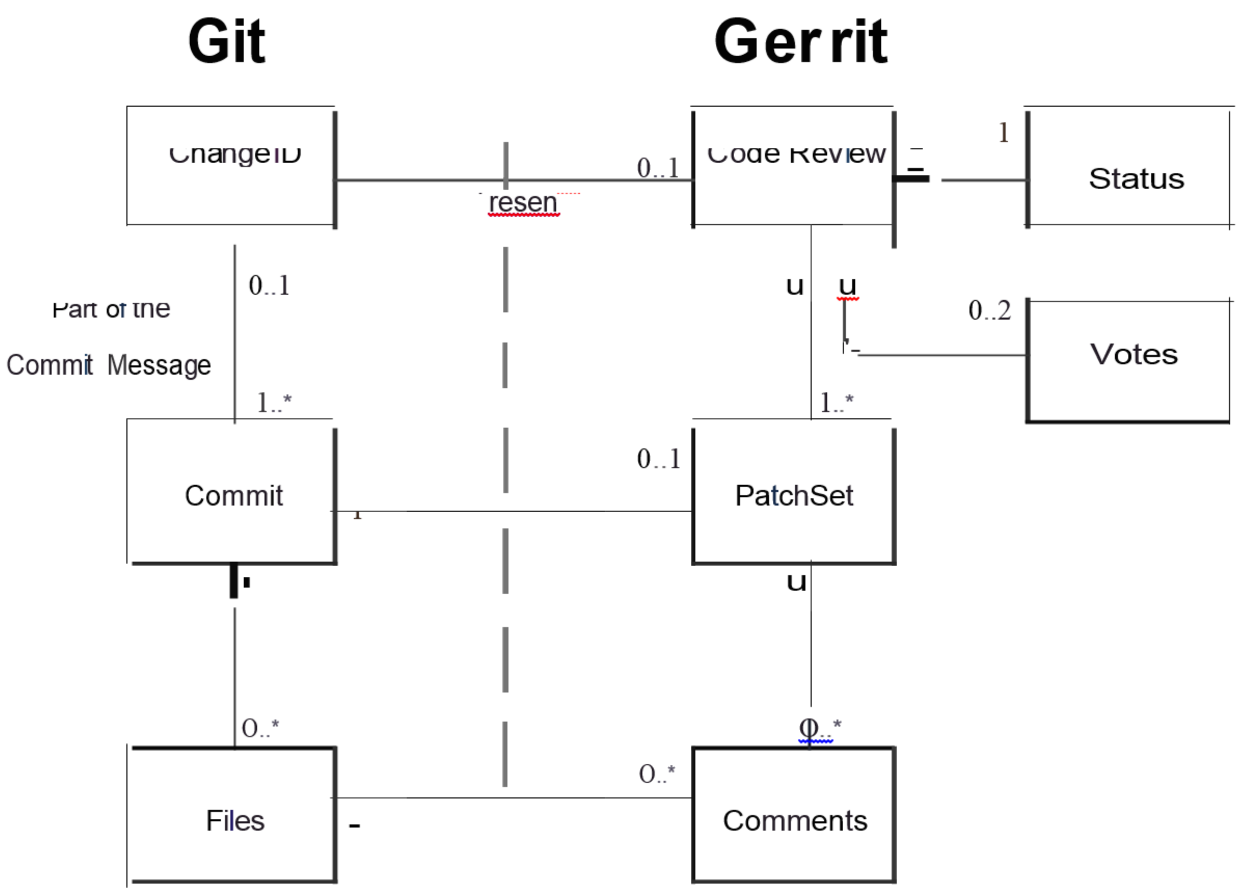
## 6.13 Common Pitfall: Push of Multiple Commits

Each commit (with a unique Change-ID), which you push to refs/for/..., will result in a new Code Review! Dependency of these commits is ensured by Gerrit.

However, this means that you cannot use “Merge” with Gerrit in general:

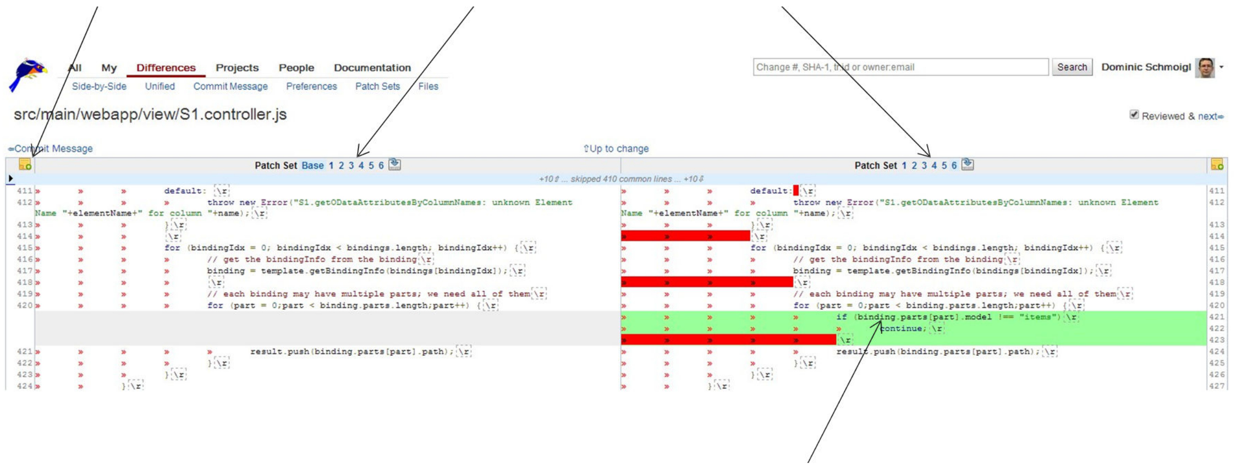
- Merge Commits usually do not have a Change-ID by default
  - push would be rejected (none of the commits will be accepted)
- Entering a new Change-ID into the merge commit
- 
  - new Code Review
- • Entering the same Change-ID into the merge commit
  - new Patchset; Changes would be overwritten

## 6.14 Mapping of Entities Git / Gerrit



## 6.15 Inline Code Review

Add a note for the entire file Compare the different PatchSets of the same file

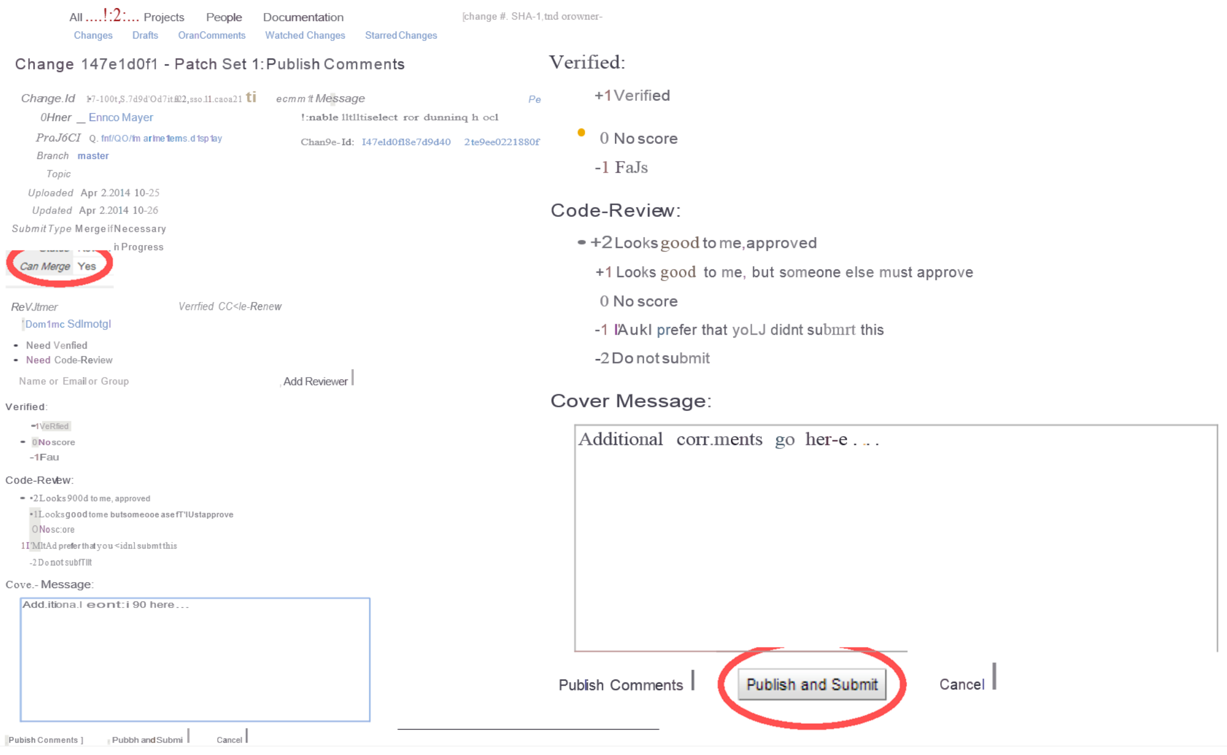


src/main/webapp/view/S1.controller.js

```
            default: \r
                throw new Error("S1.getODataAttributesByColumnNames: unknown Element Name "+elementName+" for column "+name);\r
            }\r
            \r
            for (bindingIdx = 0; bindingIdx < bindings.length; bindingIdx++) {\r
                // get the bindingInfo from the binding\r
                binding = template.getBindingInfo(bindings[bindingIdx]);\r
                \r
                // each binding may have multiple parts; we need all of them\r
                for (part = 0;part < binding.parts.length;part++) {\r
                    result.push(binding.parts[part].path);\r
                }\r
            }\r
        }\r
```

```
            default: \r
                throw new Error("S1.getODataAttributesByColumnNames: unknown Element Name "+elementName+" for column "+name);\r
            }\r
            \r
            for (bindingIdx = 0; bindingIdx < bindings.length; bindingIdx++) {\r
                // get the bindingInfo from the binding\r
                binding = template.getBindingInfo(bindings[bindingIdx]);\r
                \r
                // each binding may have multiple parts; we need all of them\r
                for (part = 0;part < binding.parts.length;part++) {\r
                        if (binding.parts[part].model !== "items") \r
                            continue; \r
                        \r
                        result.push(binding.parts[part].path);\r
                }\r
            }\r
        }\r
```

## 6.16 Review and Submit



Change 147e1d0f1 - Patch Set 1:Publish Comments

Change.Id ⊦7-100t,S.7d9d'Od7it⊞2,sso.11.caoa21 ti
0Hner _ Ennco Mayer
PraJ6CI Q. fnf/QO/fn arlme tems.d1sp1ay
Branch master
Topic
Uploaded Apr 2.2014 10-25
Updated Apr 2.2014 10-26
Submit Type Merge if Necessary
h Progress
Can Merge Yes

ecmm ‡t Message
!:nable lltlltiselect ror dunninq h ocl
Chan9e-Id: I47eld0fl8e7d9d40 2te9ee0221880f

ReVJtmer Verrfied CC<le-Renew
'Dom1mc Sdlmotgl
- Need Venfied
- Need Code-Review

Name or Email or Group

Verified:
- 1 VeRfied
- 0 No score
- -1 Fau

Code-Rewew:
- •2 Looks 900d to me, approved
- •1 Looks good tome butsomeooe asef T'IUstapprove
- 0 Nosc:ore
- 1I'MltAd prefer that you <idnl submt this
- -2 Do not subfTllt

Cove.- Message:

Add.itiona.l eont:i 90 here...

Verified:

+1 Verified

0 No score

-1 FaJs

Code-Review:

+2 Looks good to me,approved

+1 Looks good to me, but someone else must approve

0 No score

-1 I'Aukl prefer that yoLJ didnt submrt this

-2 Do not submit

Cover Message:

Additional corr.ments go her-e . . .

# Important Disclaimers and Legal Information

## Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

## Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of wilful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

## Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

## Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).

**www.sap.com/contactsap**