Bishop Blanchet 2022 FRC Rapid React Coding – Getting Started Guide

Follow the steps below to onboard the software development aspects for the 2022 Rapid React challenge. The topics generally include becoming familiar with the resources necessary to being your software development journey. The information included will help in learning more about the expected program flow that the Blanchet 2022 Rapid React robot will use during competition, practicing and some testing.

Contents

[Prerequisites 1](#_Toc93157940)

[Computer Software Setup 1](#_Toc93157941)

[Program Structure / Flow 2](#_Toc93157942)

[Required Reading 2](#_Toc93157943)

[Going Deeper 2](#_Toc93157944)

[Source Code Access 3](#_Toc93157945)

[GitHub Checkin Instructions 4](#_Toc93157946)

[Terminology / Glossary 8](#_Toc93157947)

# Prerequisites

The expected prerequisites for this area of the robot build are described below.

1. Extended access PC or Mac
   1. Requires USB
   2. If you don’t have one, Kyle may be able to loan one out
   3. PC likely a plus as Mac may have some quirks not outlined in this document
2. Internet connection
3. Basic working knowledge of Java a plus
4. Be ready to help correct this document where flaws are found, please keep this updated!

# Computer Software Setup

The steps below generally outline the requirements for your development computer. Follow the directions to install and properly configure your computer. Because some of the subsystems and control code will be developed throughout the FRC season, check back with this guide to get the latest instructions on how to setup your computer.

1. Game tools - <https://docs.wpilib.org/en/latest/docs/zero-to-robot/step-2/labview-setup.html>
   1. Older NI Components - <https://www.ni.com/en-us/support/downloads/drivers/download/packaged.frc-game-tools.369633.html>
2. Java - <https://java.com/en/download/>
3. WPILib / VS Code – this one takes a while
   1. <https://docs.wpilib.org/en/latest/docs/zero-to-robot/step-2/wpilib-setup.html>
   2. For 2022 season - <https://github.com/wpilibsuite/allwpilib/releases/tag/v2022.2.1>
      1. 2021 can also be installed from - <https://github.com/wpilibsuite/allwpilib/releases/tag/v2021.3.1>
4. Third party libraries
   1. <https://docs.wpilib.org/en/stable/docs/software/vscode-overview/3rd-party-libraries.html>
5. Phoenix / CTRE (Cross The Road Electronics)
   1. Library / Tool Install (e.g., Phoenix Tuner)
      1. <https://github.com/CrossTheRoadElec/Phoenix-Releases/releases>
   2. Documentation
      1. <https://docs.ctre-phoenix.com/en/stable/ch05a_CppJava.html>
         1. Example: <https://docs.ctre-phoenix.com/en/stable/ch23_AddResource.html#phoenix-c-java-api-documentation>
      2. <https://store.ctr-electronics.com/content/api/java/html/index.html>
   3. Firmware
      1. <https://github.com/CrossTheRoadElec/Phoenix-Releases/blob/master/ctr-device-firmware.zip?raw=true>
6. SPARK Max and Neo motors
   1. REV Hardware Client install - <https://docs.revrobotics.com/sparkmax/rev-hardware-client/getting-started-with-the-rev-hardware-client>
   2. Connecting a SPARK Max - <https://docs.revrobotics.com/rev-hardware-client/spark-max/connecting-a-spark-max>
   3. Updating a SPARK Max - <https://docs.revrobotics.com/rev-hardware-client/spark-max/updating-spark-max>
   4. Install instructions - <https://docs.revrobotics.com/sparkmax/software-resources/spark-max-api-information#java-api-installation-instructions>
   5. REV Java API - javadoc - <https://codedocs.revrobotics.com/java/com/revrobotics/package-summary.html>
   6. REV Java code examples - <https://github.com/REVrobotics/SPARK-MAX-Examples/tree/master/Java>
7. Gyros / NavX MXP
   1. Gyros - <https://docs.wpilib.org/en/stable/docs/hardware/sensors/gyros-hardware.html>
   2. NavX page - <https://pdocs.kauailabs.com/navx-mxp/>
   3. Field oriented drive - <https://pdocs.kauailabs.com/navx-mxp/examples/field-oriented-drive/>
   4. Examples - <https://pdocs.kauailabs.com/navx-mxp/examples/>
   5. User Guide - <https://pdocs.kauailabs.com/navx-mxp/wp-content/uploads/2020/09/navx2-mxp_robotics_navigation_sensor_user_guide-8.pdf>

NOTE: The steps above are expected to be mostly correct. As the software is in constant development from both FRC and hardware vendors (REV and CTRE) it is expected some of the above may get out of date, as discrepancies are found, please update the documentation here!

# Program Structure / Flow

The 2022 Blanchet Rapid React robot is expected to generally follow the command-based programming paradigm created by FRC contributors over the past few years. The command-based program flow will be used for competitions, robot practice and integration testing. It may or may not be used at component level testing. Any developer on the 2022 Blanchet Rapid React team effort is required to learn aspects of this programming model. Some aspects are not required but are recommended learning.

## Required Reading

All members of the ‘coding’ team are expected to be familiar with the command based programming model described in the links below.

1. Command based description
   1. <https://docs.wpilib.org/en/stable/docs/software/commandbased/what-is-command-based.html>
2. Subsystems
   1. <https://docs.wpilib.org/en/stable/docs/software/commandbased/subsystems.html>
3. Commands
   1. <https://docs.wpilib.org/en/stable/docs/software/commandbased/commands.html>

## Going Deeper

All members of the ‘coding’ team are encouraged to learn more about the command based programming model.

1. General concepts
   1. Design Pattern - <https://en.wikipedia.org/wiki/Design_pattern>
   2. Declarative - <https://en.wikipedia.org/wiki/Declarative_programming>
      1. Note: many aspects of the code that team members will write will tend to be imperative, so folks should not expect this to be a purist exercise
      2. Imperative - <https://en.wikipedia.org/wiki/Imperative_programming>
2. FRC details on command system
   1. Index - <https://docs.wpilib.org/en/stable/docs/software/commandbased/index.html>
   2. Command Groups - <https://docs.wpilib.org/en/stable/docs/software/commandbased/command-groups.html>
   3. Command Scheduler - <https://docs.wpilib.org/en/stable/docs/software/commandbased/command-scheduler.html>

# Source Code Access

To get access to the Java codebase that the team will be building, writing and contributing to everyone will need to learn how to use a source code control system. This system will help all team members efficiently be able to keep track of their individual changes and generally keep us all organized. Although not explicitly required to adjust source code, the steps below should generally be completed after your computer software setup from the previous section is complete.

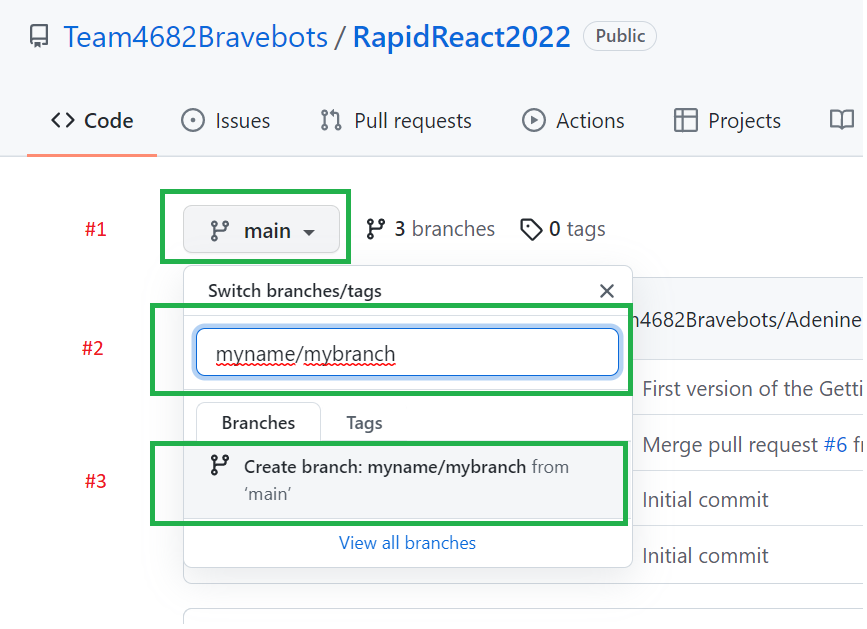
1. Setup a GitHub account
   1. <https://github.com/join>
2. Get GitHub access for the <https://github.com/Team4682Bravebots> ‘organization’
   1. Contact Kyle or Mike McAdams and tell them your GitHub logon (likely your Blanchet email)
3. Download and install git on your computer
   1. Windows - <https://git-scm.com/download/win>
   2. Mac - <https://git-scm.com/download/mac>
4. Download and install ‘GitHub Desktop’
   1. <https://desktop.github.com/>
5. Clone the ‘RapidReact2022’ repo to your machine
   1. GitHub Desktop method
      1. File -> Clone Repository
      2. GitHub.com Tab
      3. Select Team4682Bravebots/RapidReact2022
   2. cmd line method
      1. cd to a directory where you want the files to be placed
      2. git clone <https://github.com/Team4682Bravebots/RapidReact2022.git>
6. Git documentation / commands
   1. Git book - <https://git-scm.com/book/en/v2>
   2. Videos - <https://git-scm.com/videos>

# GitHub Checkin Instructions

All members of the coding team are expected to contribute their ‘robot ready’ code to the GitHub repo (https://github.com/Team4682Bravebots/RapidReact2022) that is being used for the 2022 season. The main branch of this repo will represent the code that should go on the robot for integration, practice and competitions. Using a centralized place for the code is critical so that various team members have access to the most up-to-date functionality even when other team members are not present.

The following steps summarize how to get updated code into the main branch

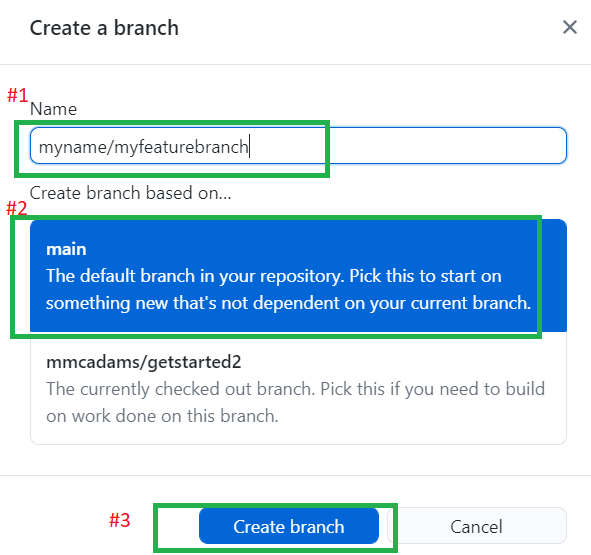
1. Get access to GitHub and the source code (see above section)
2. Create a feature branch based on the most up-to-date main branch
   1. Use the web UX



* 1. Use GitHub desktop

Graphical user interface, application

Description automatically generated



1. Update the source code and/or documents as appropriate (e.g., make your code changes, additions and removals)
   1. Then add your changes (at cmd line: git add)

Text

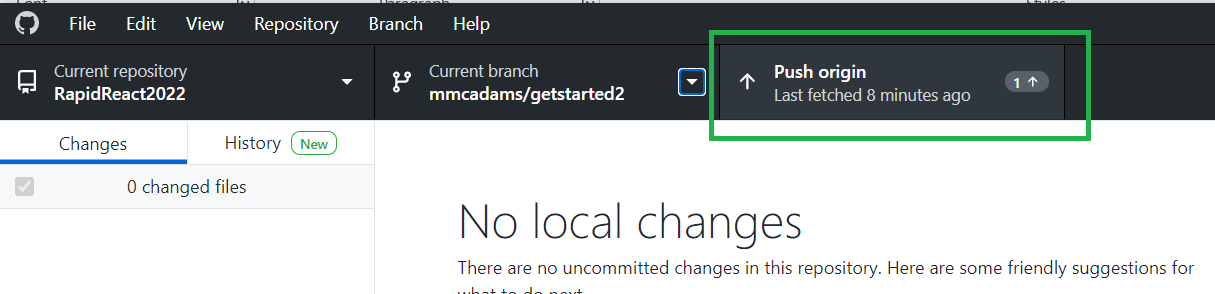
Description automatically generated

* 1. Then commit your changes (at cmd line: git commit -m “<your message here about what your change is about”)

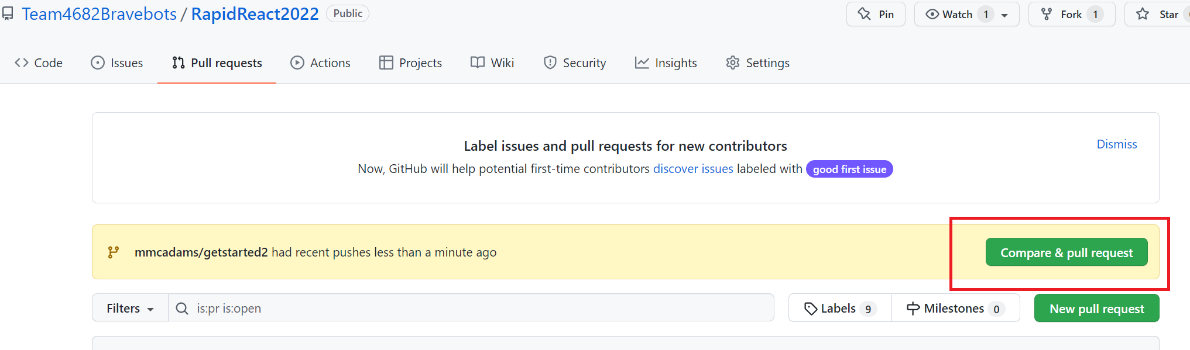
Text

Description automatically generated

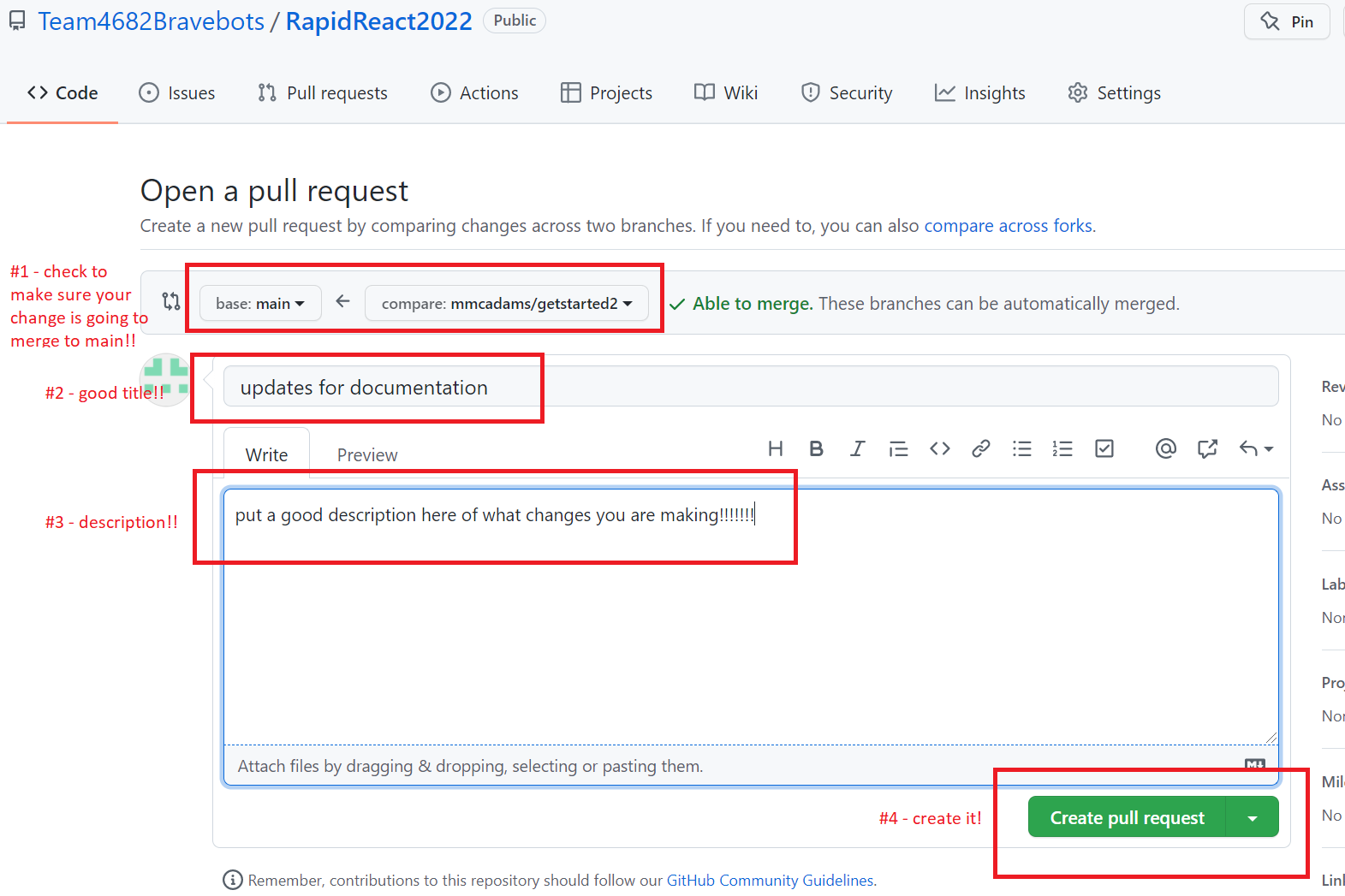
1. Publish feature branch to GitHub
   1. Use GitHub Desktop



1. Create a pull request from your feature branch to the main branch
   1. Navigate to: <https://github.com/Team4682Bravebots/RapidReact2022/pulls>
   2. Click on:

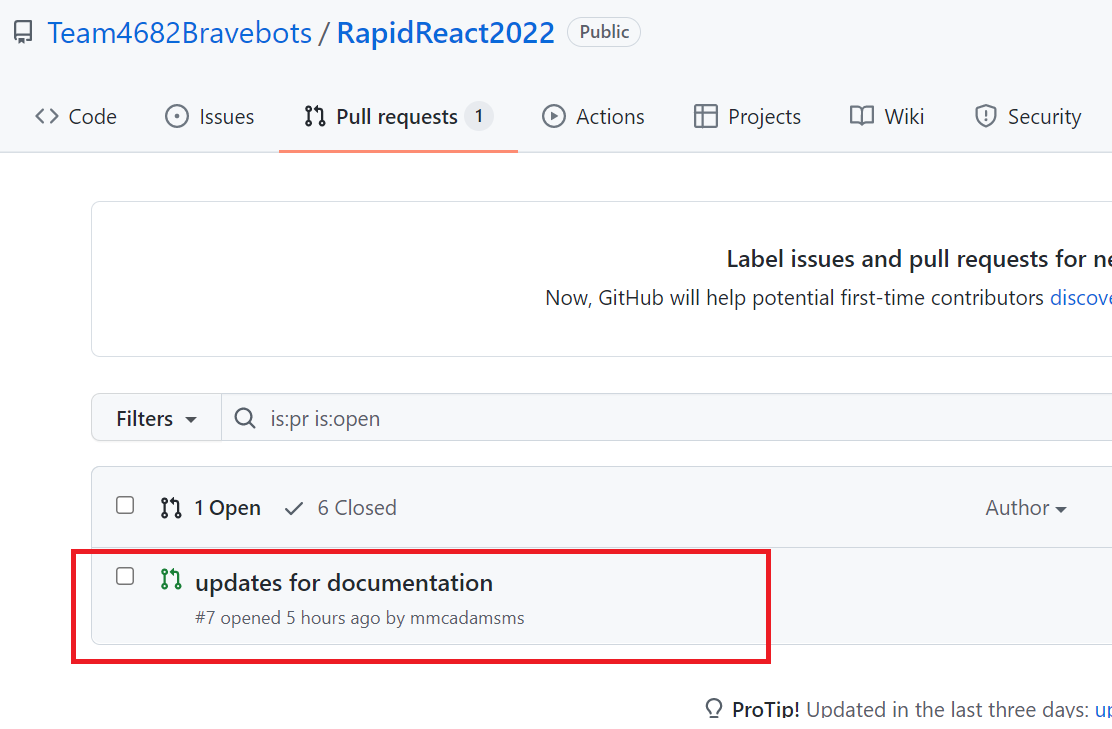


* 1. Create a PR:



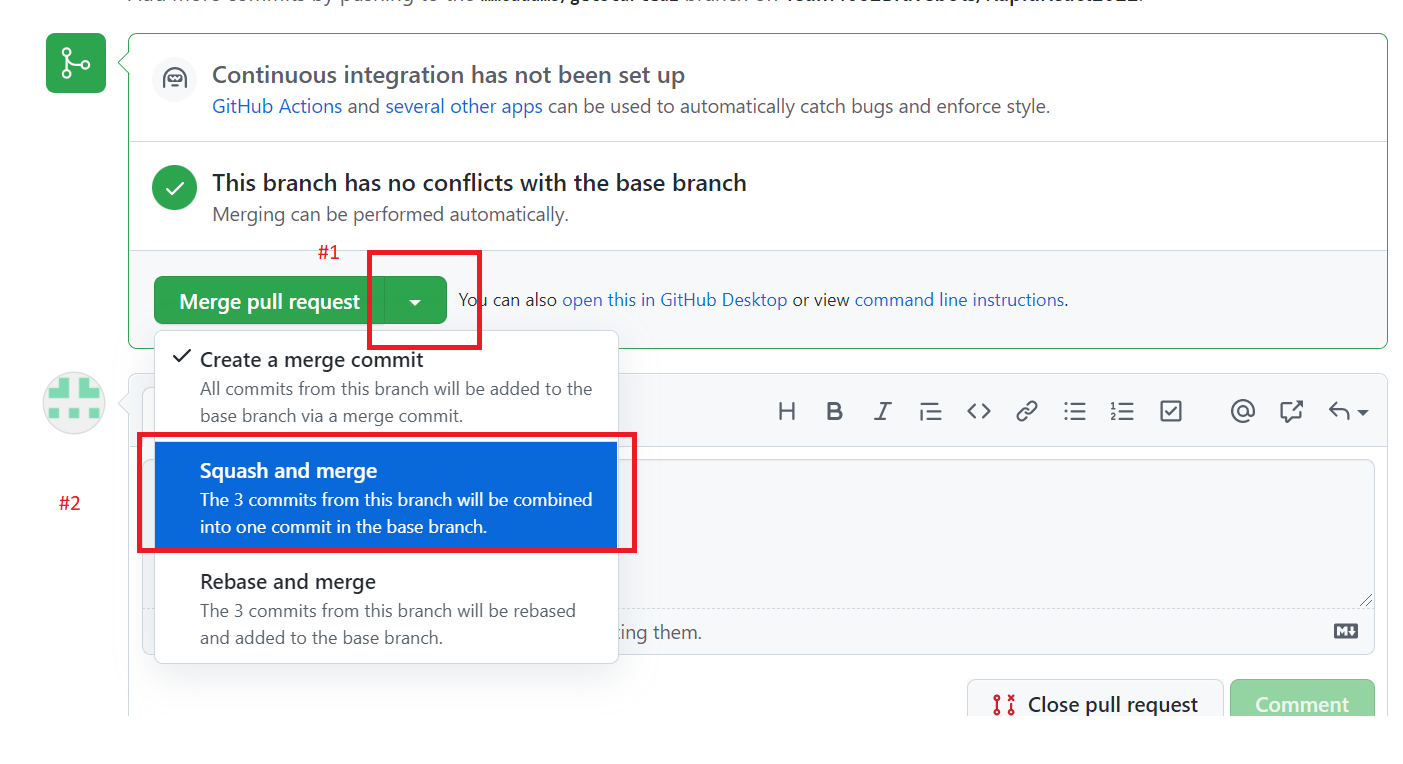
* 1. Make sure to merge/rebase main to your feature branch so that you will address merge conflicts
     1. For more info see: [How To Resolve Merge Conflicts in Git {Step-by-Step Guide} (phoenixnap.com)](https://phoenixnap.com/kb/how-to-resolve-merge-conflicts-in-git)
  2. The PR can and should be updated to accommodate changes suggested by reviewers by pushing updates to the feature branch in a similar way as in step 4 above.

1. Get someone on the team to review and approve your changes
   1. Reviewers can find all pull requests at: <https://github.com/Team4682Bravebots/RapidReact2022/pulls>
   2. Reviewers can then select the current PR:

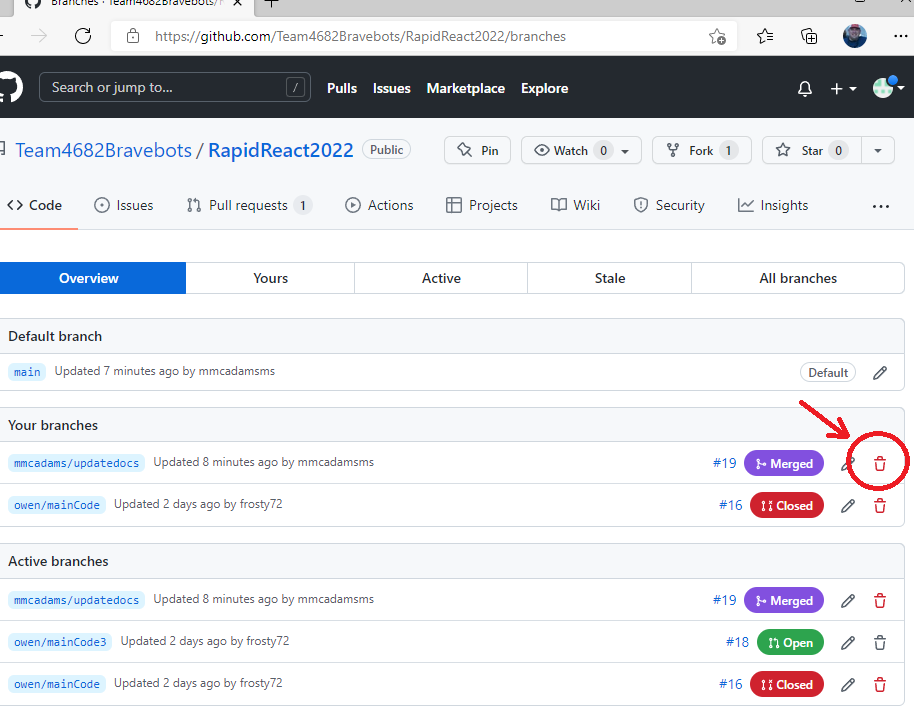


* 1. Reviewers should add comments where changes are necessary within the code.

1. Complete the pull request
   1. After making changes based on comments the author should complete the PR. This can be done via GitHub in the PR by clicking on ‘Merge pull request’
      1. Using ‘Squash and merge’ saves storage space and keeps the history cleaner so it is advised …



1. Integration test the main branch code on the robot
2. Next delete your branch (not absolutely required – but generally folks should not try to merge from the same branch to main more than once)



1. You are done – now get to work on the next thing you have to do!!!

# Terminology / Glossary

Terminology and acronyms are described in the following table.

|  |  |
| --- | --- |
| **Term / Acronym** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |