Continuous Integration with MATLAB and GitHub® Actions Workshop Homework

The Continuous Integration with MATLAB and GitHub Actions Workshop Homework is a follow-up to the Continuous Integration with MATLAB and GitHub Actions Workshop.

Homework Goal:

Use everything you've learned during the Continuous Integration with MATLAB and GitHub Actions Workshop to:

- add new files to your repository
- find and fix a bug in one of the files
- get a passing GitHub Actions result and badge

Table of Contents

Workshop Homework Prerequisites.......... 1
Homework Task 1: Add new code and tests to your repository.......... 2
Homework Task 2: Push your changes to GitHub and view the GitHub Actions test results.......... 6
Homework Task 3: Fix the bug that is causing the test failure.......... 11
Homework Task 4: Commit, push, and view results on GitHub.......... 14
Workshop Homework Wrap-Up.......... 16

Workshop Homework Prerequisites

In order to complete the workshop homework, you will need the following things:

REQ1) MathWorks and GitHub accounts

- Create MathWorks account: **https://www.mathworks.com/mwaccount/register**
- Create GitHub account: **https://github.com/signup**

REQ2) A cloned fork of the CI Configuration Examples repository

- Forking the repository is covered in **Workshop Guide Part 2**

- Cloning the repository is covered in **Workshop Guide Part 5**

REQ3) Access to MATLAB

- You can use MATLAB Online (**https://matlab.mathworks.com**) or a desktop installation of MATLAB for the workshop homework
- ***Note:*** The majority of the workshop homework and screenshots depict the MATLAB Online interface

REQ4) Workshop Homework files

- The workshop homework leverages two pre-made files that you will add to your repository
- The workshop files can be found at: **https://github.com/mathworks/ci-with-matlab-and-github-actions-workshop**
- The necessary function and test files can be found within the "`code/CodeWithoutBug`" folder of the above repository

Homework Task 1: Add new code and tests to your repository

For this homework task, we have provided you with pre-prepared function and test files to add to your repository.

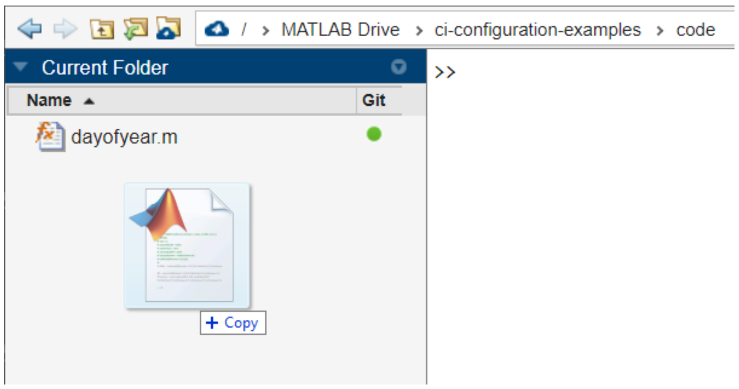
Note: In order for the code in this task to work, you will need to add the shared documents to your MATLAB Drive ***(see Workshop Homework REQ3)*** and execute the code from the root of your repository.

H1A) Copy the code and test files from the GitHub Actions Workshop files to your repository

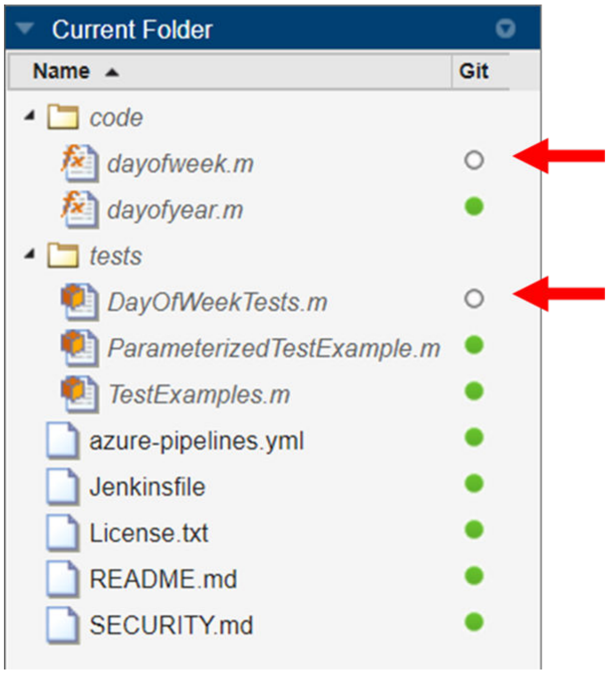
If you have not done so already, head over to **https://github.com/mathworks/ci-with-matlab-and-github-actions-workshop** and download a copy of the repository.

Extract the workshop files from the ZIP file, and drag and drop the files into your repository on MATLAB Online.

- Copy "`dayofweek.m`" to the "`code`" folder
- Copy "`DayOfWeekTests.m`" to the "`tests`" folder



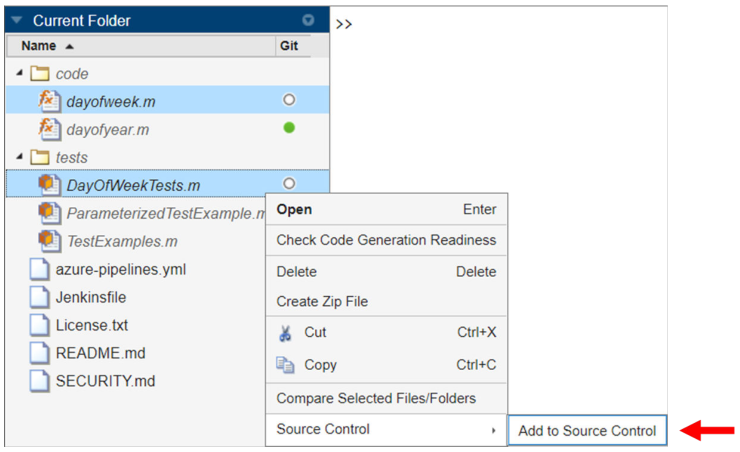
Both of the new files will have an empty circle in their Git status column, signifying that they are not currently being tracked by Git in our repository. We will add these files to the repository in the next step.



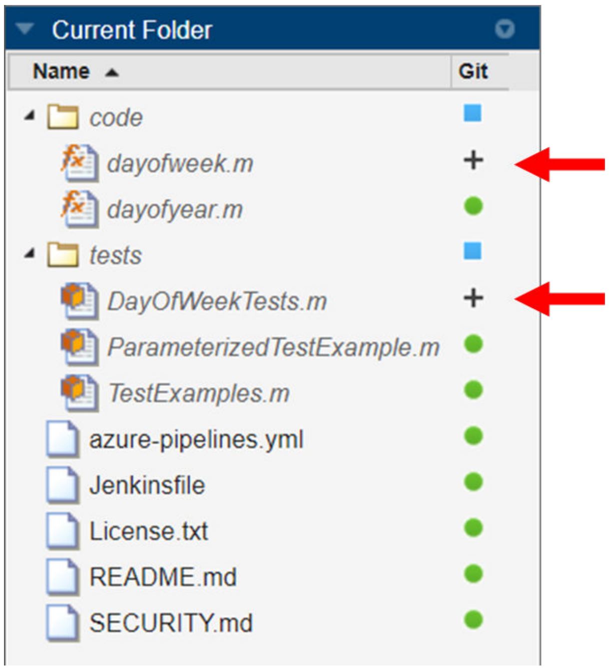
H1B) Add both new files to Git

For each file:

- Right-click the file > Source Control > Add to Source Control

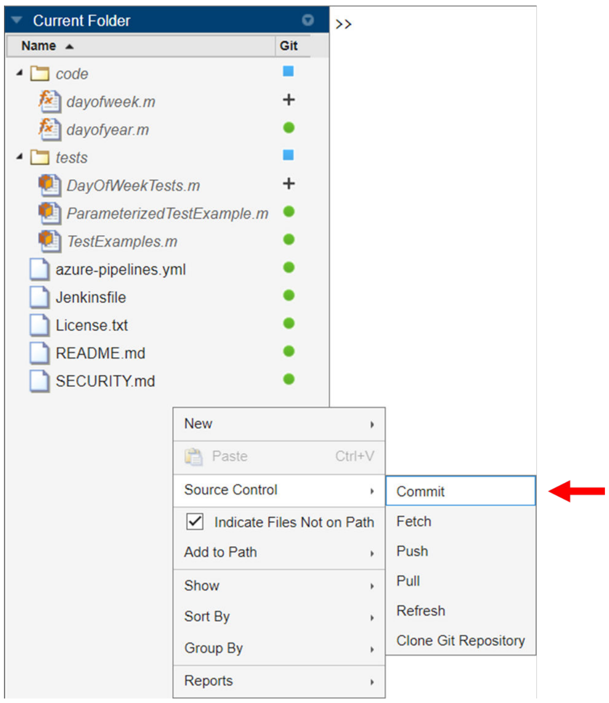


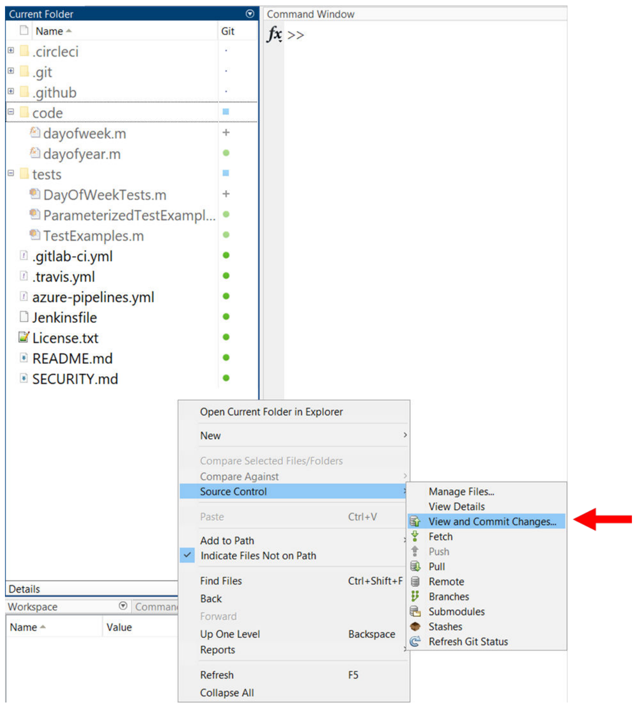
Once added, the files will have a "+" icon in their Git status column.



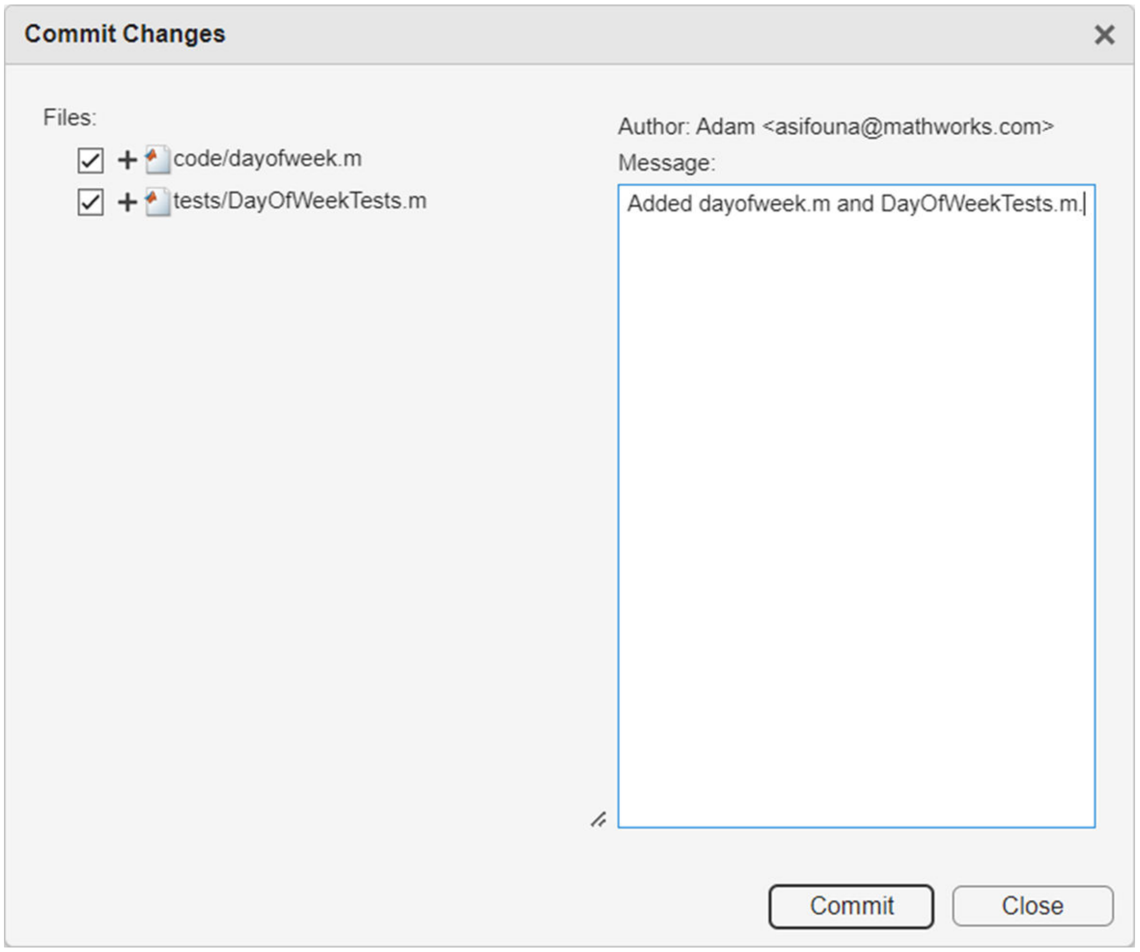
H1C) Right-click in the whitespace of the Current Folder Browser > Source Control > Commit

<u>MATLAB Online</u> <u>Desktop MATLAB</u>





H1D) Add a commit message



Files:

- ☑ + code/dayofweek.m
- ☑ + tests/DayOfWeekTests.m

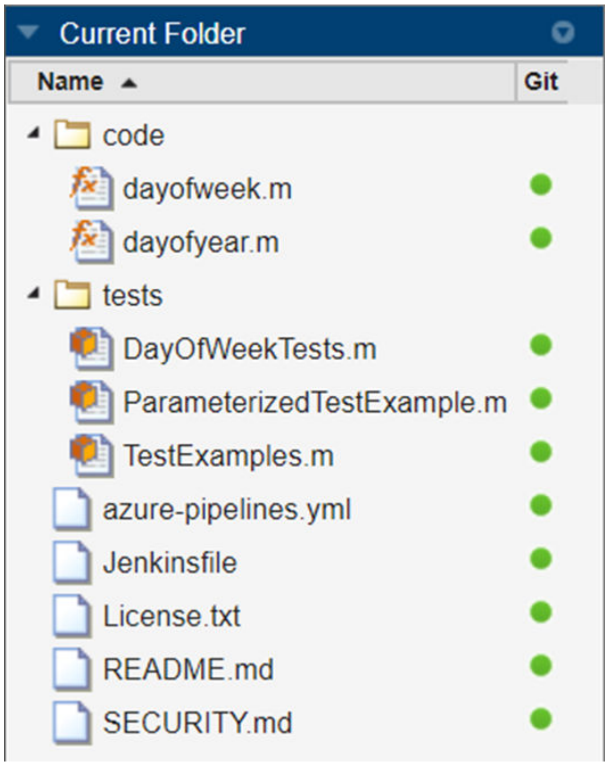
Author: Adam <asifouna@mathworks.com>

Message:

Added dayofweek.m and DayOfWeekTests.m.

H1E) Press the "Commit" button

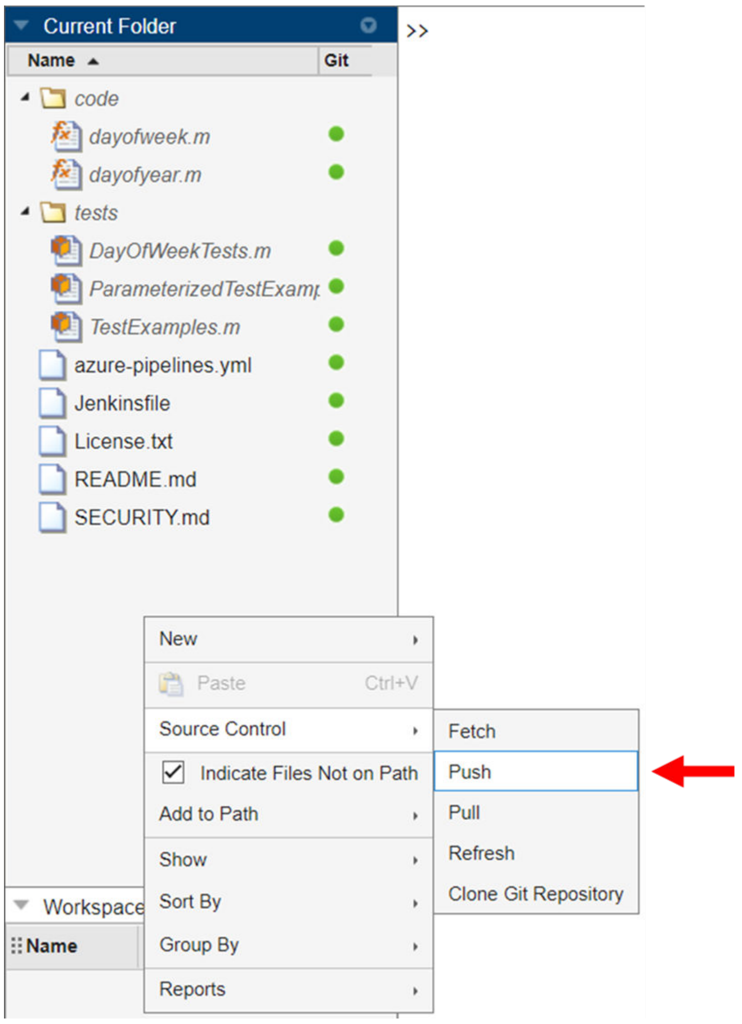
All of your changes are now recorded in source control, and all of the Git status icons should be green.



Homework Task 2: Push your changes to GitHub and view the GitHub Actions test results

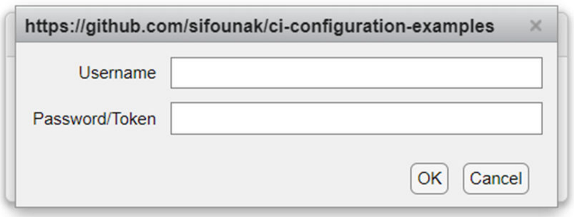
Just like during the workshop, the GitHub Actions configuration we provided will automatically run all of your tests every time a new commit is pushed to the main branch of your repository. Let's go ahead and push our changes to GitHub and see the test results there.

H2A) Right-click a blank area in the Current Folder Browser > Source Control > Push



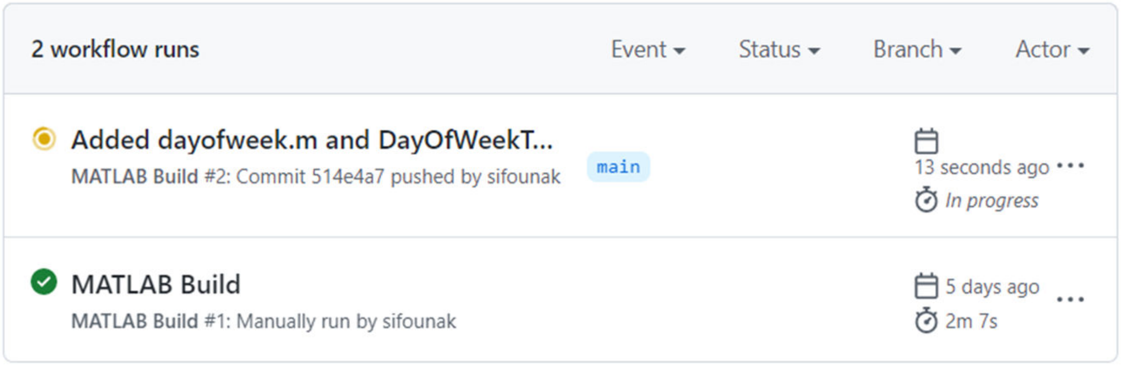
If you are using MATLAB Online:

- You may need to generate a new personal access token from GitHub to complete the push dialog below
- Please see **Workshop Guide Parts 9B-9F** for instructions on generating a new personal access token



H2B) In your GitHub repository, go to the "Actions" section to see your automated testing in action

You can see your latest code changes are automatically being tested by GitHub Actions.

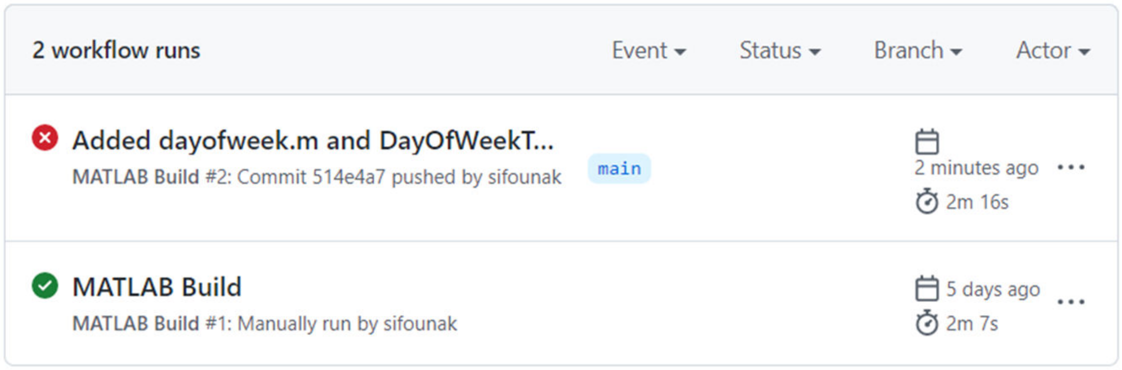

2 workflow runs

Event ▾ Status ▾ Branch ▾ Actor ▾

Added dayofweek.m and DayOfWeekT...
MATLAB Build #2: Commit 514e4a7 pushed by sifounak main
13 seconds ago
In progress

MATLAB Build
MATLAB Build #1: Manually run by sifounak
5 days ago
2m 7s

After about 1-2 minutes, you should see:


2 workflow runs

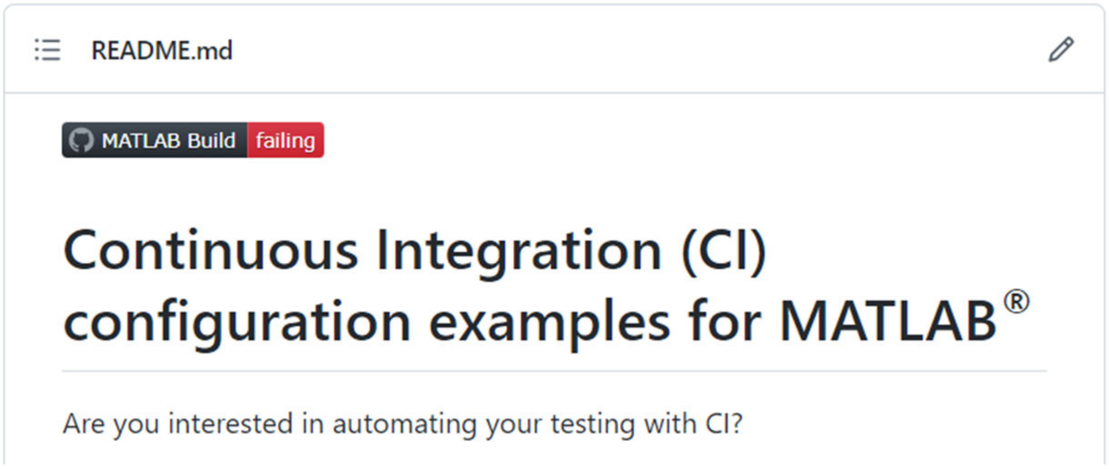
Event ▾ Status ▾ Branch ▾ Actor ▾

Added dayofweek.m and DayOfWeekT...
MATLAB Build #2: Commit 514e4a7 pushed by sifounak main
2 minutes ago
2m 16s

MATLAB Build
MATLAB Build #1: Manually run by sifounak
5 days ago
2m 7s

Oh no! It looks like at least one of our tests failed!

The badge on your repository main page should also reflect the test failure.


README.md

MATLAB Build failing

Continuous Integration (CI) configuration examples for MATLAB®

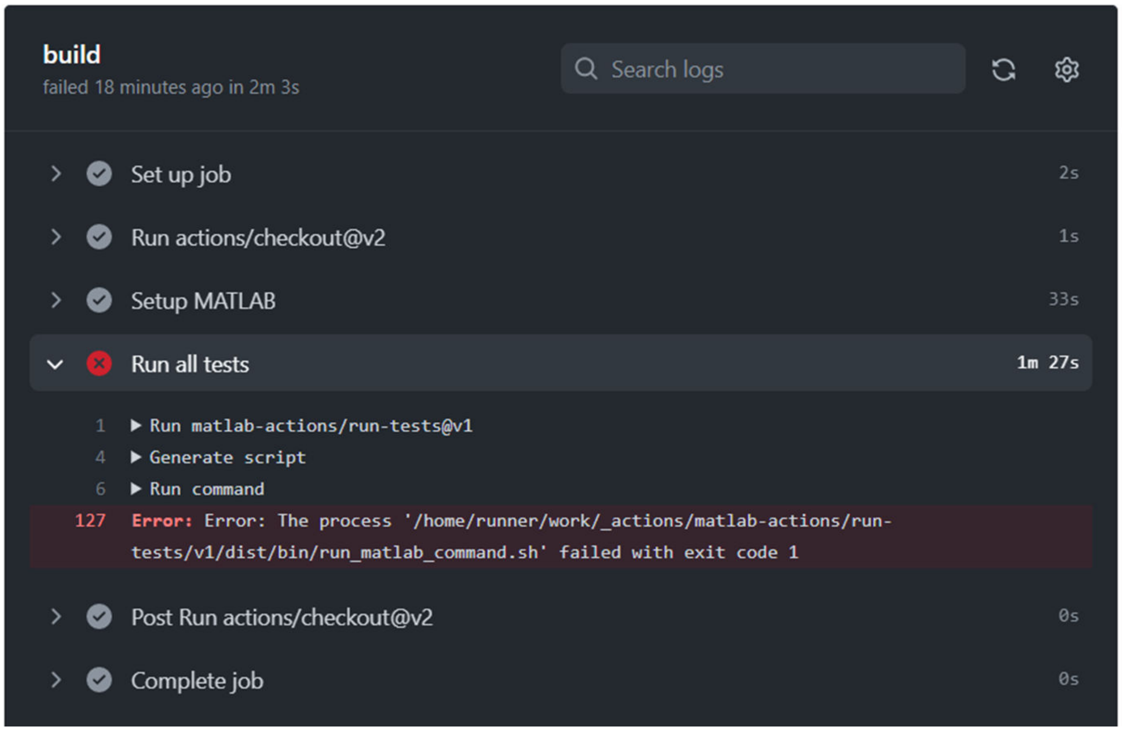
Are you interested in automating your testing with CI?

Note: The badge on your repository's main page may not update right away due to web browser image caching. To force the badge image to be reloaded, you may do either of the following actions:

- Hold the "`Shift`" key as you press the refresh icon
- Press "`Shift + F5`"

H2C) Investigate the errors

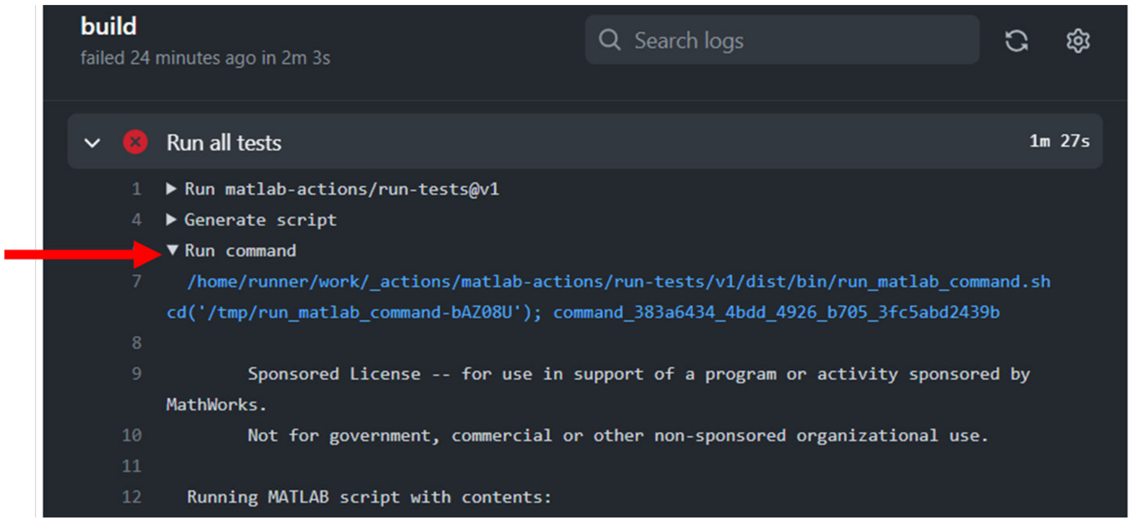
Click through the build items in the "Actions" area until you reach the build log.



build
failed 18 minutes ago in 2m 3s
Search logs
Set up job 2s
Run actions/checkout@v2 1s
Setup MATLAB 33s
Run all tests 1m 27s
Post Run actions/checkout@v2 0s
Complete job 0s

```
▶ Run matlab-actions/run-tests@v1
▶ Generate script
▶ Run command
Error: Error: The process '/home/runner/work/_actions/matlab-actions/run-
tests/v1/dist/bin/run_matlab_command.sh' failed with exit code 1
```

Expand the "Run all tests > Run command" section to examine the testing diagnostics.

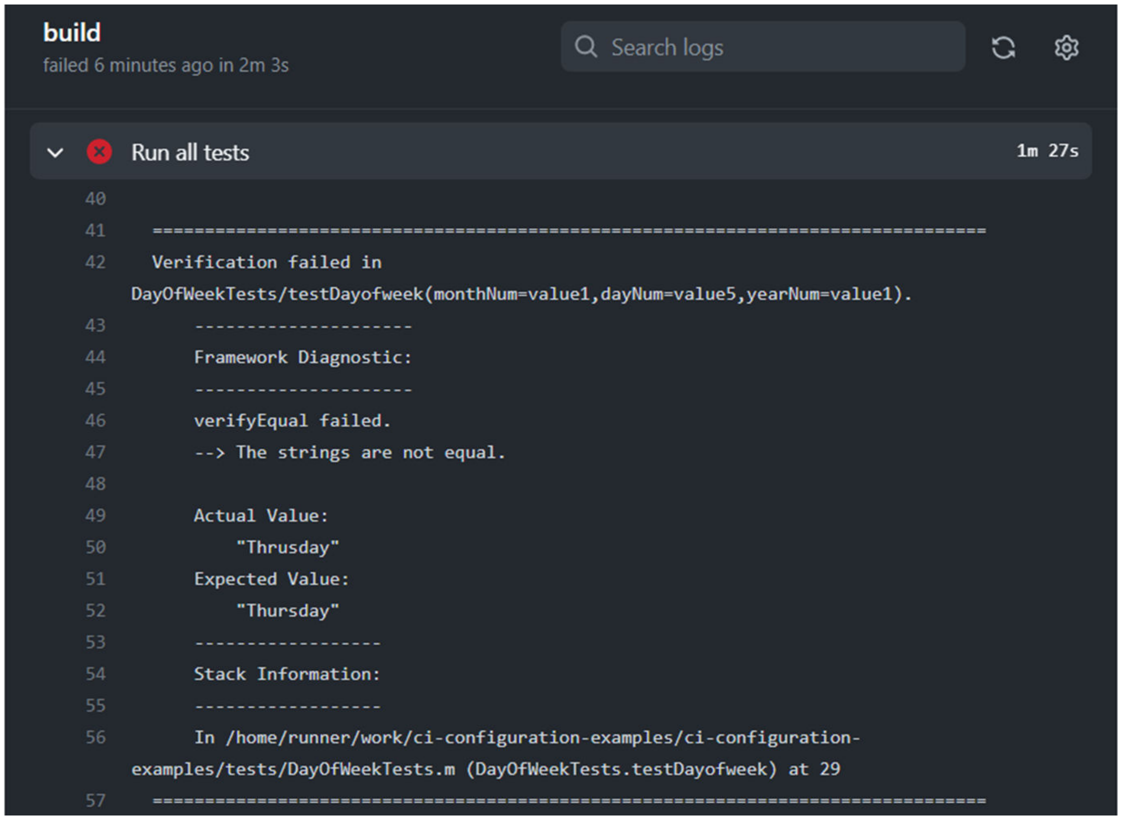


```
▶ Run matlab-actions/run-tests@v1
▶ Generate script
▼ Run command
  /home/runner/work/_actions/matlab-actions/run-tests/v1/dist/bin/run_matlab_command.sh
cd('/tmp/run_matlab_command-bAZ08U'); command_383a6434_4bdd_4926_b705_3fc5abd2439b

        Sponsored License -- for use in support of a program or activity sponsored by
MathWorks.
        Not for government, commercial or other non-sponsored organizational use.

  Running MATLAB script with contents:
```

As you look through the test diagnostics, you will notice the following failure from a test in `DayOfWeekTests.m`.


build
failed 6 minutes ago in 2m 3s

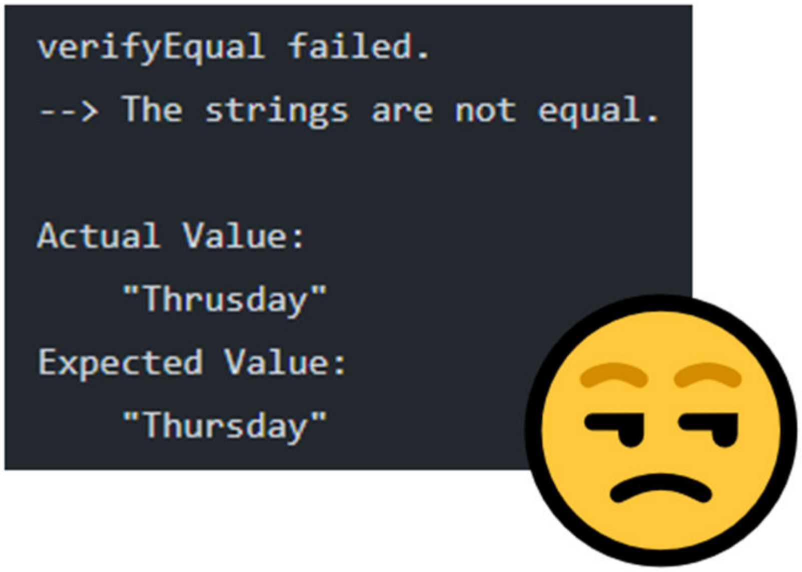
Run all tests 1m 27s

```

================================================================================
Verification failed in DayOfWeekTests/testDayofweek(monthNum=value1,dayNum=value5,yearNum=value1).
    --------------------
    Framework Diagnostic:
    --------------------
    verifyEqual failed.
    --> The strings are not equal.

    Actual Value:
        "Thrusday"
    Expected Value:
        "Thursday"
    ------------------
    Stack Information:
    ------------------
    In /home/runner/work/ci-configuration-examples/ci-configuration-examples/tests/DayOfWeekTests.m (DayOfWeekTests.testDayofweek) at 29
================================================================================
```

Looking closer, we notice:


```
verifyEqual failed.
--> The strings are not equal.

Actual Value:
    "Thrusday"
Expected Value:
    "Thursday"
```

It looks like the cause of this error is a misspelling related to the testing of the "`dayofweek`" algorithm.

But was it misspelled in the source code (`dayofweek.m`) or the test (`DayOfWeekTests.m`)?

Most people assume that all bugs come from the source code, but sometimes the bugs are in the test itself. The MATLAB Unit Test Framework uses a convention to make it easy to detect whether the bug coming from the source code or the test:

Syntax: **`verifyEqual(testCase, actual, expected)`**

- `actual` = Value returned by your source code
- `expected` = Value your test is comparing your source code against

If the "`actual`" value is wrong, the bug is in your source code. If the "`expected`" value is wrong, the bug is in your test.

In our case, we can see the spelling mistake is in the actual value, so we know that the bug is in the "`dayofweek.m`" source code.

Homework Task 3: Fix the bug that is causing the test failure

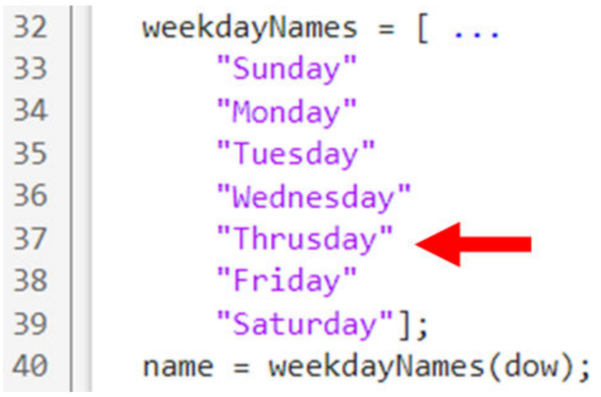
H3A) In MATLAB Online, open `dayofweek.m`

From our investigation within GitHub Actions, we know that the issue is in `dayofweek.m`.

If you'd like to see the test results directly in MATLAB Online, you can simply run the tests again in MATLAB Online ***(see Step H3C)***.

H3B) Find and fix the bug in `dayofweek.m`

The misspelling can be found on line 37.

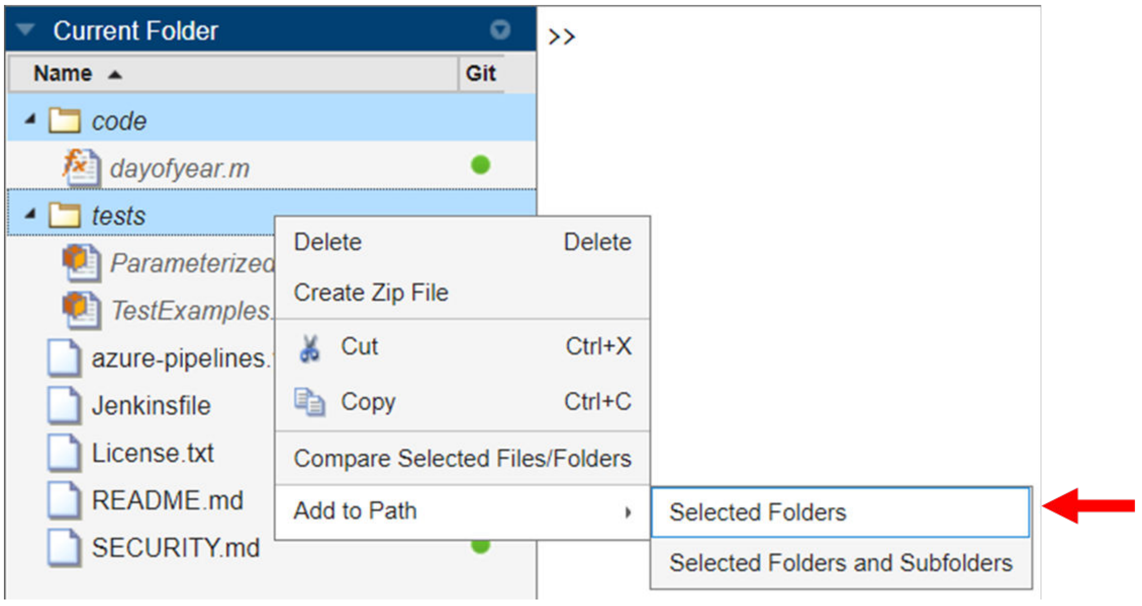


```
weekdayNames = [ ...
    "Sunday"
    "Monday"
    "Tuesday"
    "Wednesday"
    "Thrusday"
    "Friday"
    "Saturday"];
name = weekdayNames(dow);
```

H3C) Run your tests to ensure you fixed the bug

Earlier in the workshop, we ended up with a bug and failed tests in our published GitHub repository because we submitted code updates without testing them first. Let's not make that mistake again!

First, let's add the "`code`" and "`tests`" folders to the path by right-clicking on the folders > Add to Path > Selected Folders.



Next, we can run the tests in the codebase by executing any of the following pieces of code:

Option 1) Go into the "`tests`" folder and execute "`runtests`"

```
cd tests
runtests
```

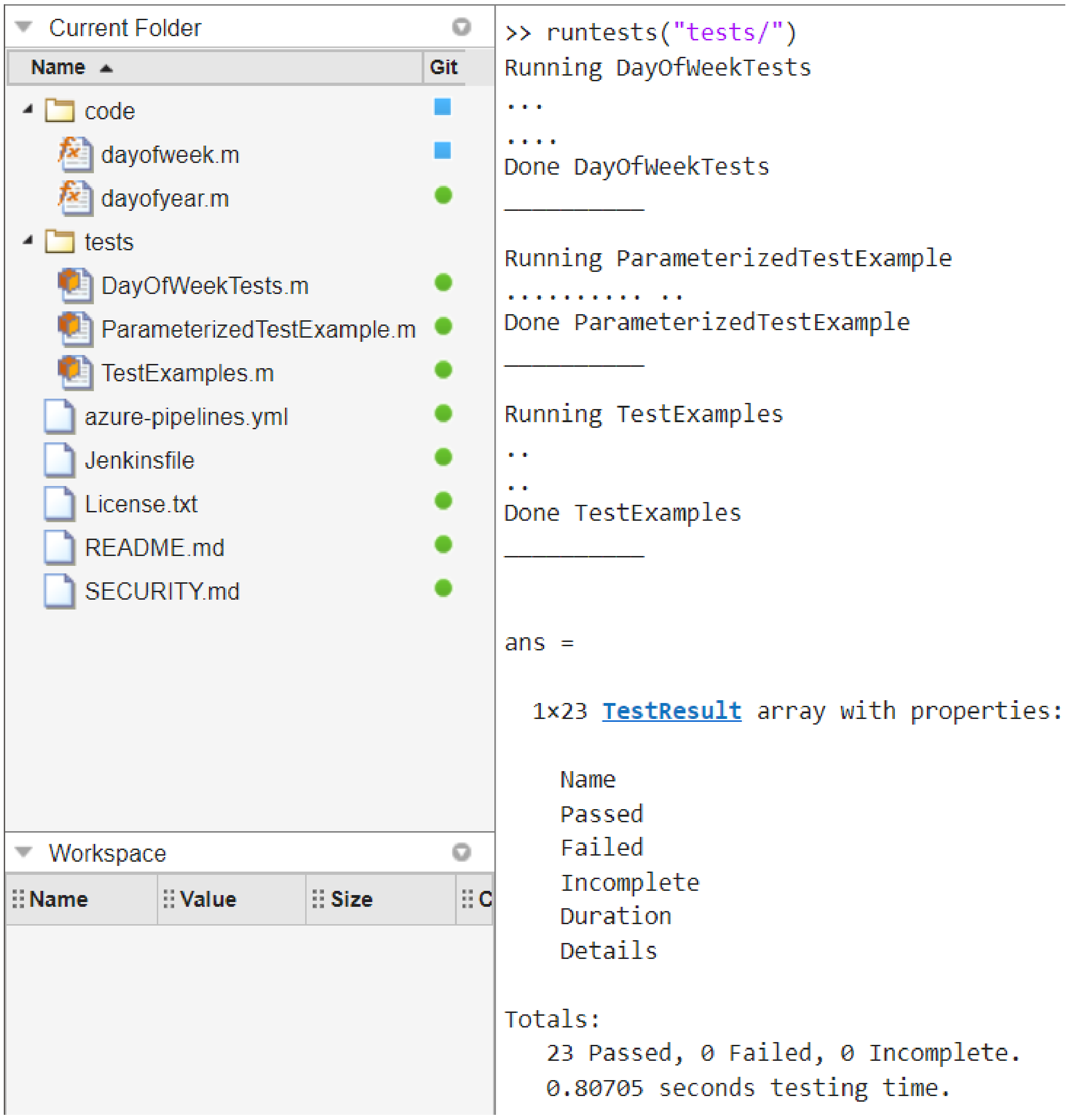
Option 2) Provide `runtests` with the path to the folder containing your tests

```
runtests("tests/")
```

Option 3) Ask `runtests` to search all subfolders for tests and automatically run them

```
runtests(IncludeSubfolders=true)
```

The output of your test run should look like:



```
>> runtests("tests/")
Running DayOfWeekTests
...
....
Done DayOfWeekTests
__________

Running ParameterizedTestExample
.......... ..
Done ParameterizedTestExample
__________

Running TestExamples
..
..
Done TestExamples
__________


ans =

  1×23 TestResult array with properties:

    Name
    Passed
    Failed
    Incomplete
    Duration
    Details

Totals:
   23 Passed, 0 Failed, 0 Incomplete.
   0.80705 seconds testing time.
```

It looks like we fixed the bug and all of our tests now pass! Yay!

Note: If you are starting the homework after completing the workshop, remember to also fix the bug we introduced during **Workshop Guide Part 7**.

Homework Task 4: Commit, push, and view results on GitHub

Now that we've verified our bug fix, we're ready to commit and push the changes to our repository.

H4A) Commit the changes to your repository

- Right-click in the whitespace of the Current Folder Browser > Source Control > Commit
- Add a useful commit message
- Press "Commit"

H4B) Push the changes to your repository

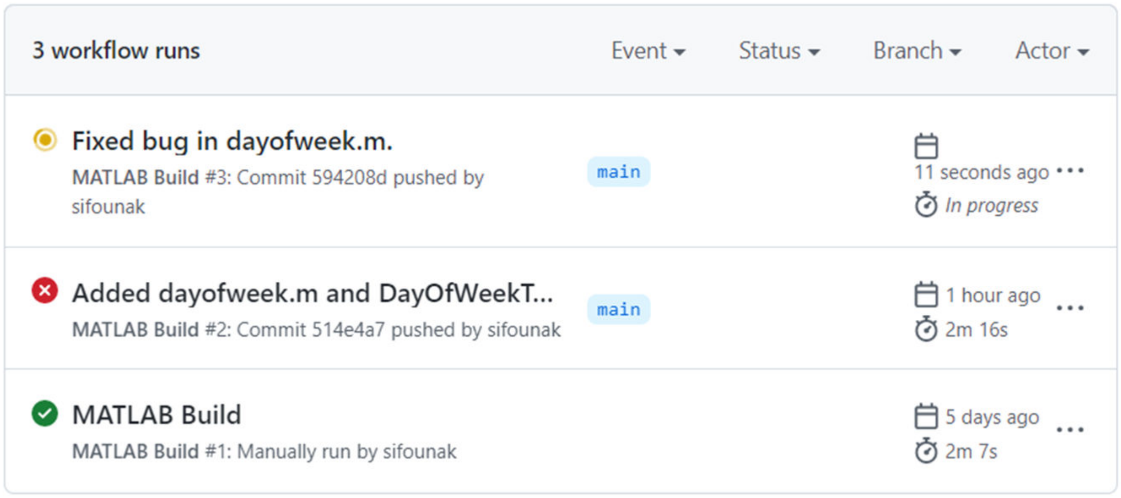
- Right-click in the whitespace of the Current Folder Browser > Source Control > Push
- Press "Push"

Notes:

- You will need to use your personal access token again here
- If you have lost access to your personal access token, you will need to create a new token ***(see Workshop Guide Part 9)***

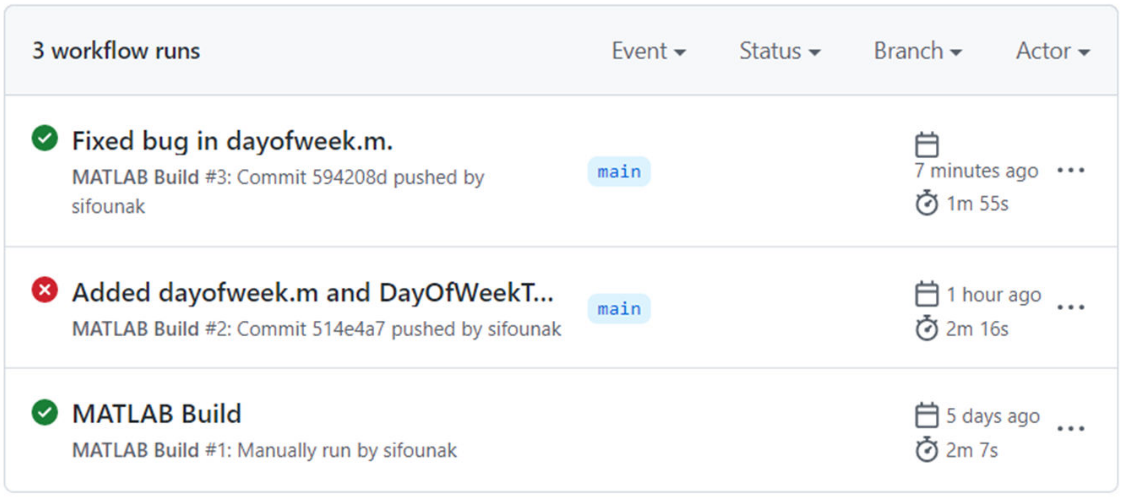
H4C) Switch back to your GitHub repository and verify that the automated build was successful

You can go into the "Actions" area to watch the progress.


3 workflow runs

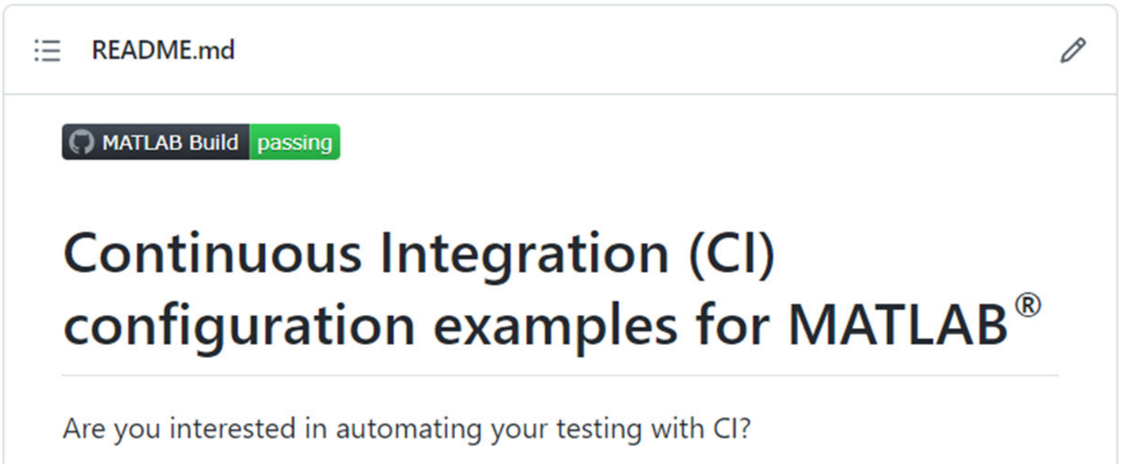
Fixed bug in dayofweek.m. MATLAB Build #3: Commit 594208d pushed by sifounak	main	11 seconds ago *In progress*
Added dayofweek.m and DayOfWeekT... MATLAB Build #2: Commit 514e4a7 pushed by sifounak	main	1 hour ago 2m 16s
MATLAB Build MATLAB Build #1: Manually run by sifounak		5 days ago 2m 7s

The job status will update automatically when the testing is complete.


3 workflow runs

Fixed bug in dayofweek.m. MATLAB Build #3: Commit 594208d pushed by sifounak	main	7 minutes ago 1m 55s
Added dayofweek.m and DayOfWeekT... MATLAB Build #2: Commit 514e4a7 pushed by sifounak	main	1 hour ago 2m 16s
MATLAB Build MATLAB Build #1: Manually run by sifounak		5 days ago 2m 7s

You can also see the updated badge on your repository's main page.



Continuous Integration (CI) configuration examples for MATLAB®

Are you interested in automating your testing with CI?

Note: The badge on your repository's main page may not update right away due to web browser image caching. To force the badge image to be reloaded, you may do either of the following actions:

- Hold the "`Shift`" key as you press the refresh icon
- Press "`Shift + F5`"

Workshop Homework Wrap-Up

Congratulations on completing the Continuous Integration with MATLAB and GitHub Actions Workshop Homework!

Through the workshop and the homework, you have successfully:

- forked a repository to your GitHub account
- cloned a repository to MATLAB Online
- added and updated files in your repository
- automatically run tests via GitHub Actions
- use GitHub Actions to identify bugs in your MATLAB code
- created a repository with a passing GitHub Actions badge

We look forward to seeing all the great things you will continue to do with MATLAB and continuous integration!

Copyright 2022 The MathWorks, Inc.