Article
Evgeny Shvarov · May 28, 2022 3m read

Open Exchange

Something For Nothing or How to Make Github Run Your UnitTests

Hi Developers!

This is yet another short post that is intended to simplify developers' life. Now we'll talk about how to make GitHub run unit tests with every push to the repository by adding just one file to the repo. For free. On Github Cloud. Sounds great, isn't it?

It is possible and very easy to do. Credit goes to @Dmitry Maslennikov (and his repo), ZPM Package Manager, and GitHub Actions. Let's see how this all works!

Let's take a repo with tests first, e.g. the basic IRIS template.

Also, let's assume that you load your dev code to IRIS docker using ZPM. If not, take a look the video on how to do this.

In this particular repo, it goes with the following line and with the presence of module.xml:

```
    zpm "load /home/irisowner/irisbuild/ -v":1:1
```

Next, let's add a GitHub Actions scenario that will perform building the image and running tests:

```
name: unittest

on:
  push:
    branches:
      - master
      - main
  pull_request:
    branches:
      - master
      - main
  release:
    types:
      - released

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Build and Test
        uses: docker/build-push-action@v2
        with:
          context: .
          push: false
          load: true
          tags: ${{ github.repository }}:${{ github.sha }}
          build-args: TESTS=1
```

This scenario builds docker image using Dockerfile of the repository on every **push** or **pull-request** to the **master** or the **main** branch.

There is an additional line that transfers TESTS=1 variable to Dockerfile:

```
          build-args: TESTS=1
```

This argument is evaluated in the Dockerfile and in case TESTS=1 it runs tests of zpm package of the repository:

([$TESTS -eq 0] || iris session iris -U $NAMESPACE "##class(%ZPM.PackageManager).Shell(\"test $MODULE -v -only\",1,1)") && \

This line checks $TESTS parameter and if it doesn't not equal to 0 it opens the iris session in $NAMESPACE (IRISAPP in this case) and runs ZPM command:

```
test $MODULE -v -only
```

-only flag runs only the test without loading module (as it was loaded before in the docker build scenario).

Module name is set via $MODULE="dc-sample-template" in this case:

```
ARG MODULE="dc-sample-template"
```

The command will run all the unittests we mentioned in ZPM module. In this case, we state it with this line:

```
      <UnitTest Name="/tests" Package="dc.sample.unittests" Phase="test"/>
```

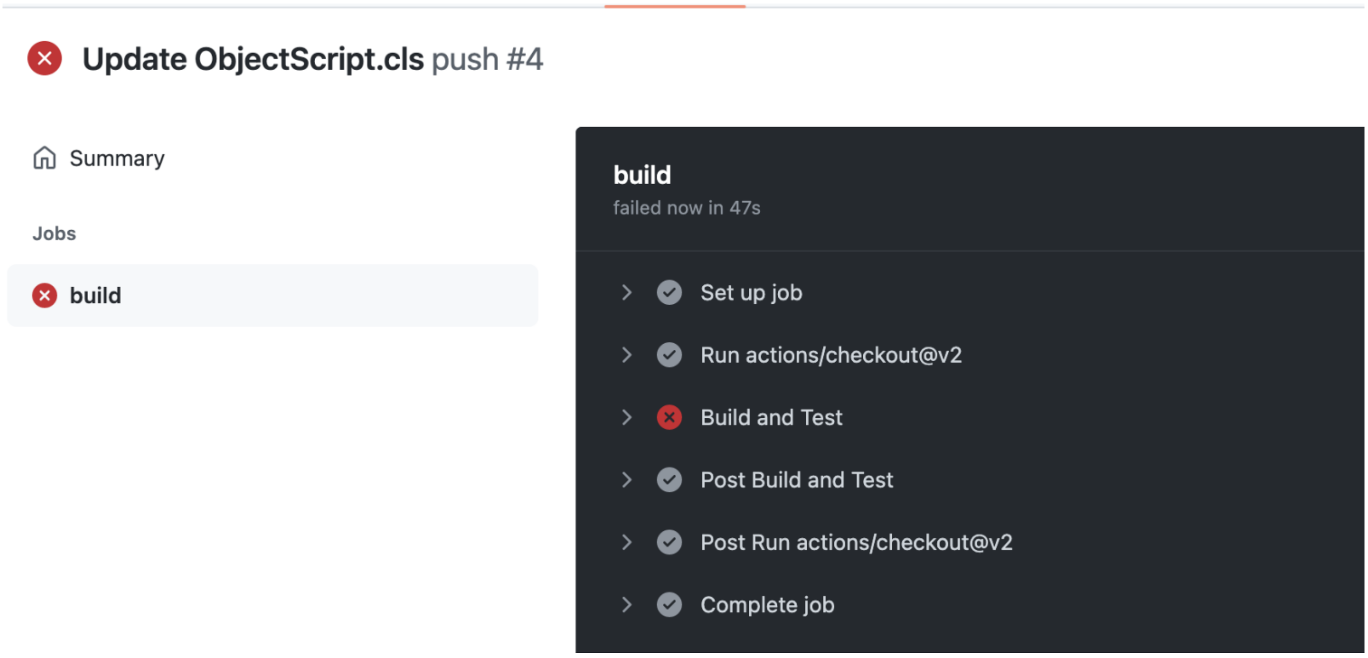
which means that unit tests are situated in /tests folder of the repository and the resource is dc.sample.unittests class package that has two classes.

This scenario will build the image of your repository on the GitHub cloud and run tests for every push or pull request to the master branch!

Let's see how it works!

Test42 method expects the method dc.sample.ObjectScript).Test() to return 42.

Let's change the line to 43 and send the pull-request. We can see (in the Actions section) that the pull request initiated the Github Action Indeed, the build failed:


Update ObjectScript.cls push #4

Summary

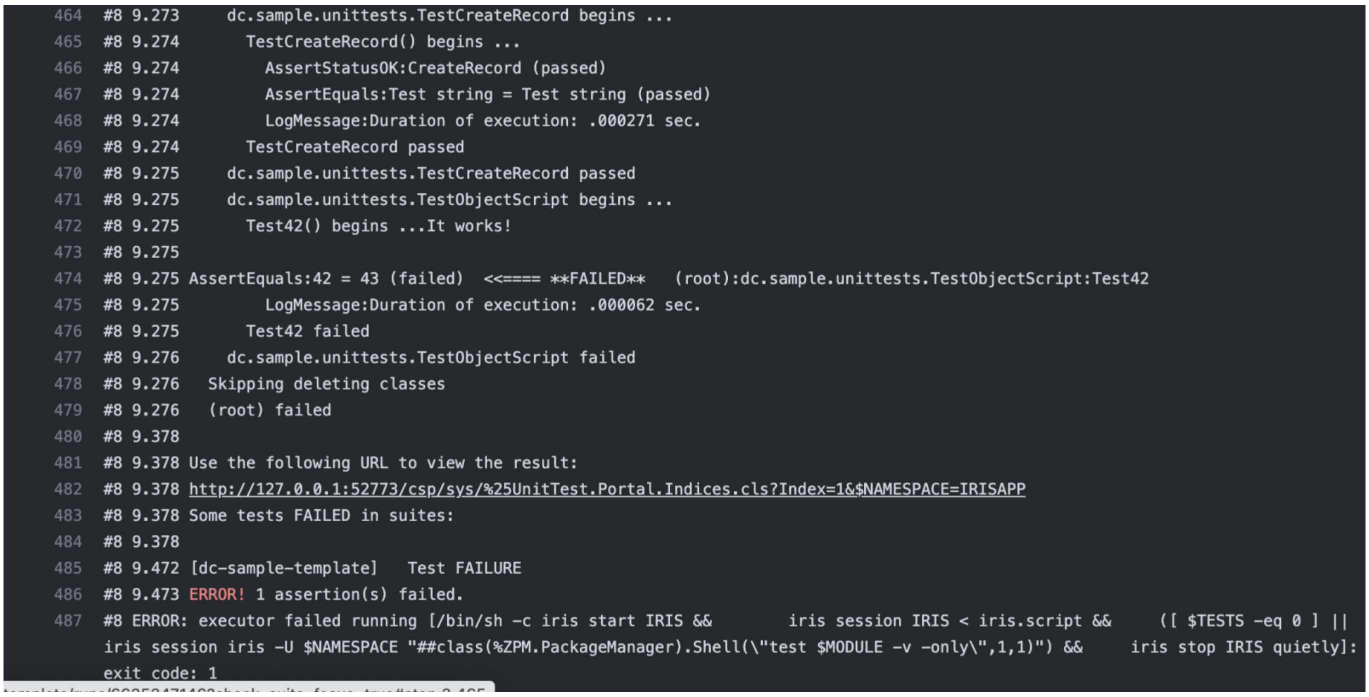
Jobs

build

build
failed now in 47s

- Set up job
- Run actions/checkout@v2
- Build and Test
- Post Build and Test
- Post Run actions/checkout@v2
- Complete job

And the details of the phase say that Test42() failed on AssertEquals 42 =43:



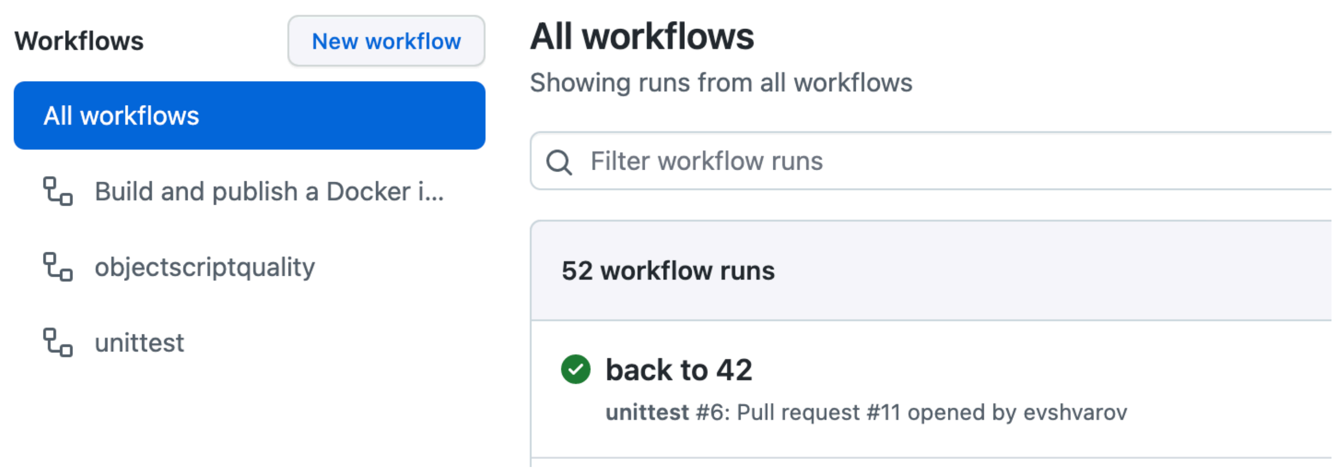
```
#8 9.273     dc.sample.unittests.TestCreateRecord begins ...
#8 9.274       TestCreateRecord() begins ...
#8 9.274         AssertStatusOK:CreateRecord (passed)
#8 9.274         AssertEquals:Test string = Test string (passed)
#8 9.274         LogMessage:Duration of execution: .000271 sec.
#8 9.274       TestCreateRecord passed
#8 9.275     dc.sample.unittests.TestCreateRecord passed
#8 9.275     dc.sample.unittests.TestObjectScript begins ...
#8 9.275       Test42() begins ...It works!
#8 9.275
#8 9.275 AssertEquals:42 = 43 (failed)  <<==== **FAILED**   (root):dc.sample.unittests.TestObjectScript:Test42
#8 9.275         LogMessage:Duration of execution: .000062 sec.
#8 9.275       Test42 failed
#8 9.276     dc.sample.unittests.TestObjectScript failed
#8 9.276   Skipping deleting classes
#8 9.276   (root) failed
#8 9.378
#8 9.378 Use the following URL to view the result:
#8 9.378 http://127.0.0.1:52773/csp/sys/%25UnitTest.Portal.Indices.cls?Index=1&$NAMESPACE=IRISAPP
#8 9.378 Some tests FAILED in suites:
#8 9.378
#8 9.472 [dc-sample-template]   Test FAILURE
#8 9.473 ERROR! 1 assertion(s) failed.
#8 ERROR: executor failed running [/bin/sh -c iris start IRIS &&        iris session IRIS < iris.script &&     ([ $TESTS -eq 0 ] ||
iris session iris -U $NAMESPACE "##class(%ZPM.PackageManager).Shell(\"test $MODULE -v -only\",1,1)") &&     iris stop IRIS quietly]:
exit code: 1
```

Github also sends email notifications of failed CI builds.

Note: to run tests locally just call:

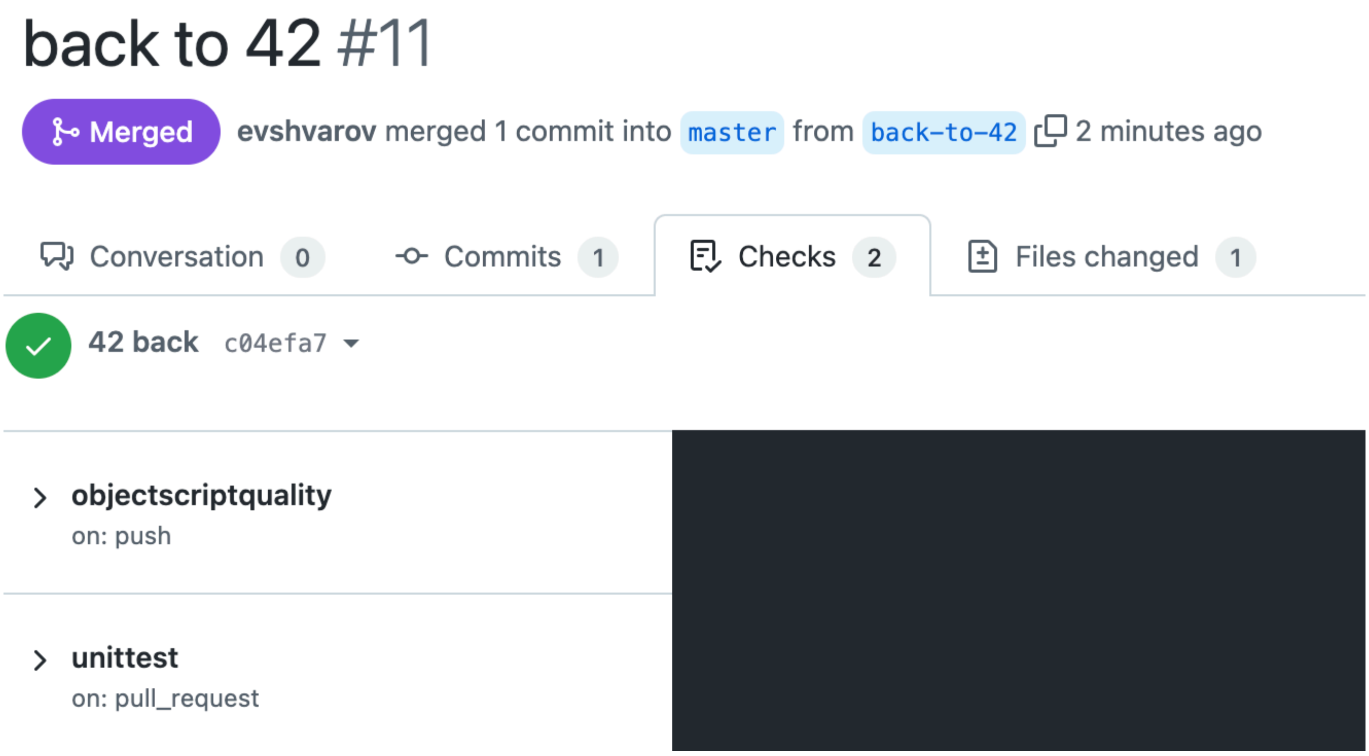
USER>zpm test "module-name"

And if we turn the value of the variable back to 42 the tests will be OK!



back to 42
unittest #6: Pull request #11 opened by evshvarov

And if this is a Pull-Request you can see the result of CI in the special section [Checks](#):


back to 42 #11

Merged evshvarov merged 1 commit into master from back-to-42 2 minutes ago

Conversation 0 | Commits 1 | Checks 2 | Files changed 1

42 back c04efa7

objectscriptquality
on: push

unittest
on: pull_request

That's it!

To sum up making GitHub running (for free!) docker builds and unittests of InterSystems IRIS projects on pushes or pull-requests needs the one CI Github Actions scenario, that will be the same in every project, and three lines in Dockerfile:

ZPM $MODULE name - this needs to be updated with every project,

$TEST parameter,

and the RUN TEST line.

And use the ZPM Package Manager!

Open to comments and feedback, and happy coding!

P.S. the title image is related to one of my favorite SF writers Robert Sheckley's story "Something for Nothing" - here it is in audio too, enjoy!

#Development Environment #DevOps #Docker #GitHub #InterSystems Package Manager (IPM) #Testing #InterSystems IRIS
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/something-nothing-or-how-make-github-run-your-unittests