Doxygen for DUNE DAQ

Andrew J. Mogan

DUNE DAQ Software Coordination Meeting

June 13, 2024

Doxygen

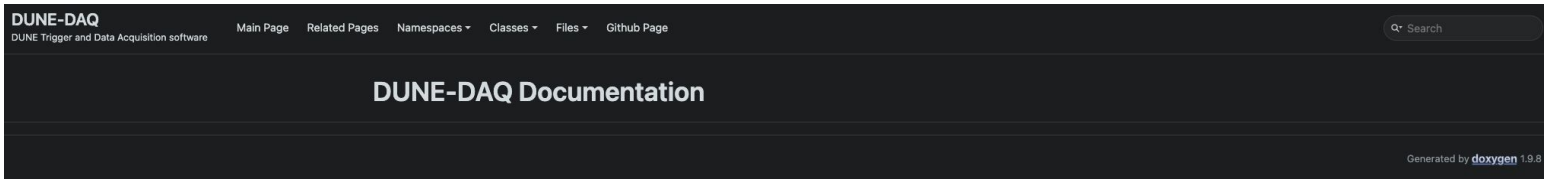
- Automated documentation generator
- Parses many file types, notably C++ and Python source files
- Gathers information on classes, namespaces, etc.
  - Dependency graphs
  - Cross-referencing (automatically generates links)
- Motivation: we now have the potential for many generated classes in the v5/develop line
  - OKS classes spun off of xml files
- We now have a nightly Doxygen workflow to keep up-to-date documentation of these classes etc.
  - Currently only for v5/develop line
- The following slides cover how this works; the result is published to the [DUNE DAQ Github Pages]

How Doxygen Works (in General)

- In general, Doxygen is straightforward to [install and run]

```
INSTALL_DIR=/home/$USER/doxygen-install-dir # Or whatever
wget https://www.doxygen.nl/files/doxygen-1.10.0.src.tar.gz
gunzip doxygen-1.10.0.src.tar.gz
tar xf doxygen-1.10.0.src.tar
cd doxygen-1.10.0/
mkdir build
cd build
cmake -G "Unix Makefiles" \
        -DCMAKE_INSTALL_PREFIX=$INSTALL_DIR .. # Else permission complaints
make && make install
export PATH=$INSTALL_DIR/bin:$PATH
cd /path/to/project
doxygen -g # Generates Doxyfile; modify accordingly, then run
doxygen
```

- Should get an index.html file as output; point a web server to this file

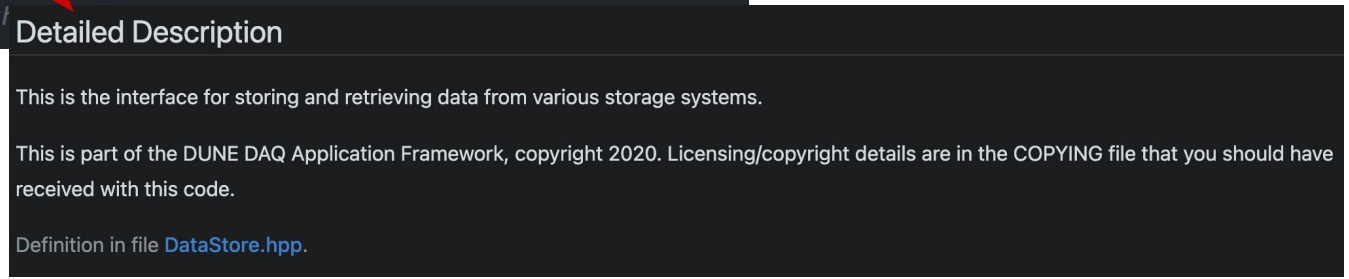


Fermilab

How Doxygen Works – Documenting the Code

- Doxygen parses header and source files for special comment blocks

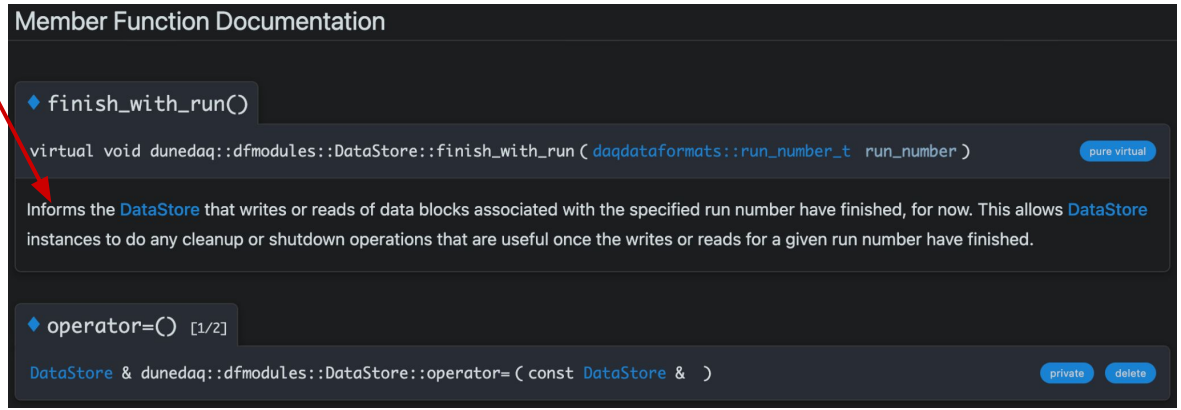
```
/**
 * @file DataStore.hpp
 *
 * This is the interface for storing and retrieving data from
 * various storage systems.
 *
 * This is part of the DUNE DAQ Application Framework, copyright 2020.
 * Licensing/copyright details are in the COPYING file that you should have
 * received with this code.
 */

// 09-Sep-2020, KAB: the initial version of this class was based on the
// Queue interface from th
```



How Doxygen Works – Documenting the Code

- Doxygen parses header and source files for special comment blocks

```
/**
 * @brief Informs the DataStore that writes or reads of data blocks associated
 * with the specified run number have finished, for now.
 * This allows DataStore instances to do any cleanup or shutdown operations
 * that are useful once the writes or reads for a given run number have finished.
 */
virtual void finish_with_run(daqdataformats::run_number_t run_number) = 0;
```

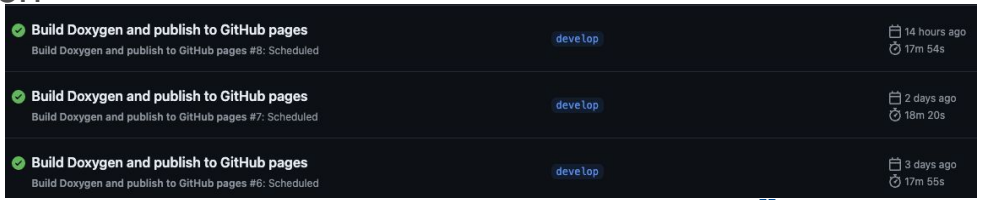


Fermilab

The Workflow

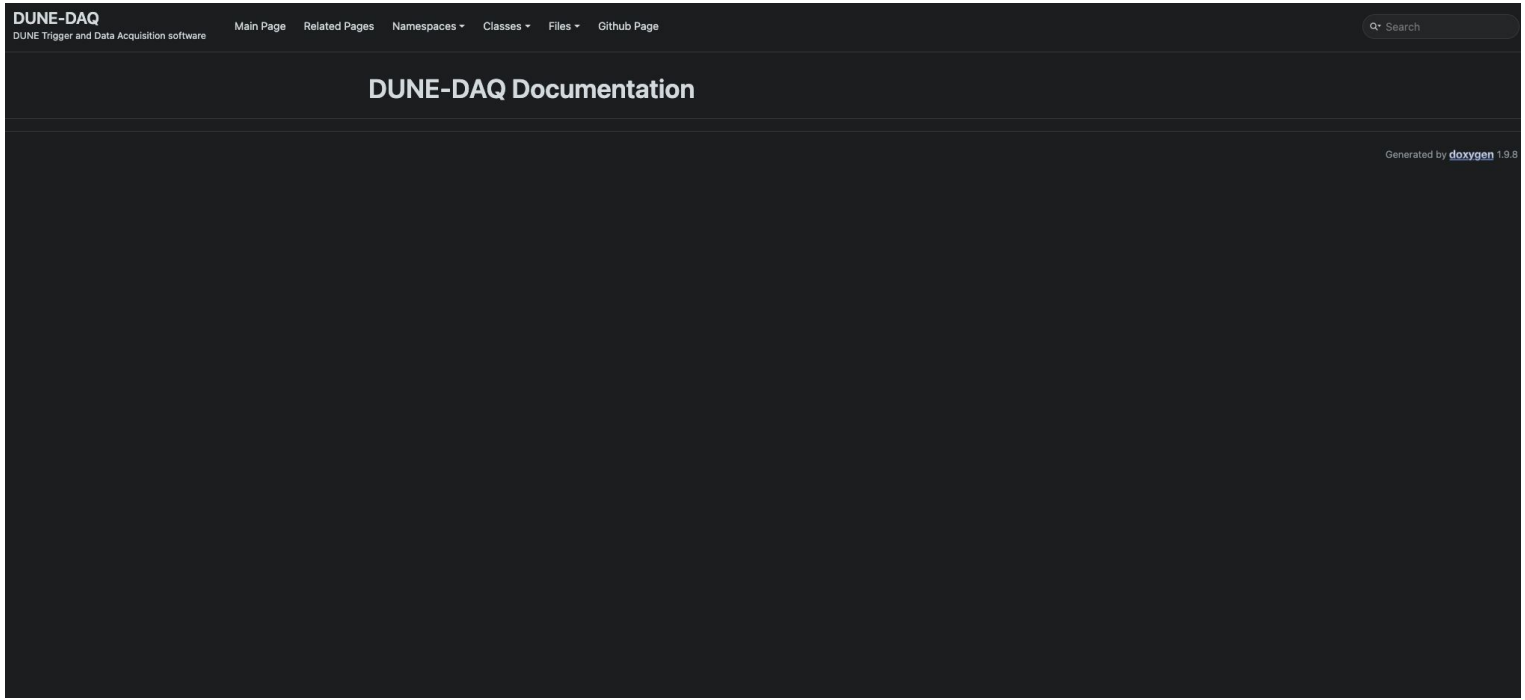
- The workflow lives in the docs repo
- Automatically triggered nightly at 5 a.m. UTC
  - Two hours after the nightly A9 build starts so as to have access to said build
- Runs on a GitHub-hosted runner in the container output by the nightly build (has cvmfs access)
- Alessandro made some handy scripts while testing Doxygen; those are now in daq-release
  - Starts with template Doxyfile and overwrites the input directories based on what's available
  - triggeralgs is currently excluded for technical reasons
- Creates a local dev area, checks out all fddaq and coredaq packages into it, and runs Doxygen
- Publishes result to GitHub Pages
- Takes less than 20 minutes total

```yaml
name: Build Doxygen and publish to GitHub pages

on:
  schedule:
    - cron: "0 5 * * *"

jobs:
  build-doxygen:
    name: build_doxygen
    runs-on: ubuntu-latest
    container:
      image: "ghcr.io/dune-daq/nightly-release-alma9:development_v5"
    env: …
    defaults: …

    steps:
    - name: Checkout daq-release …

    - name: Create dbt area …

    - name: Checkout packages …

    - name: Build dbt area …

    - name: Generate Doxyfile …

    - name: Doxygen Action …

    - name: Deploy …
```



The Result – Homepage

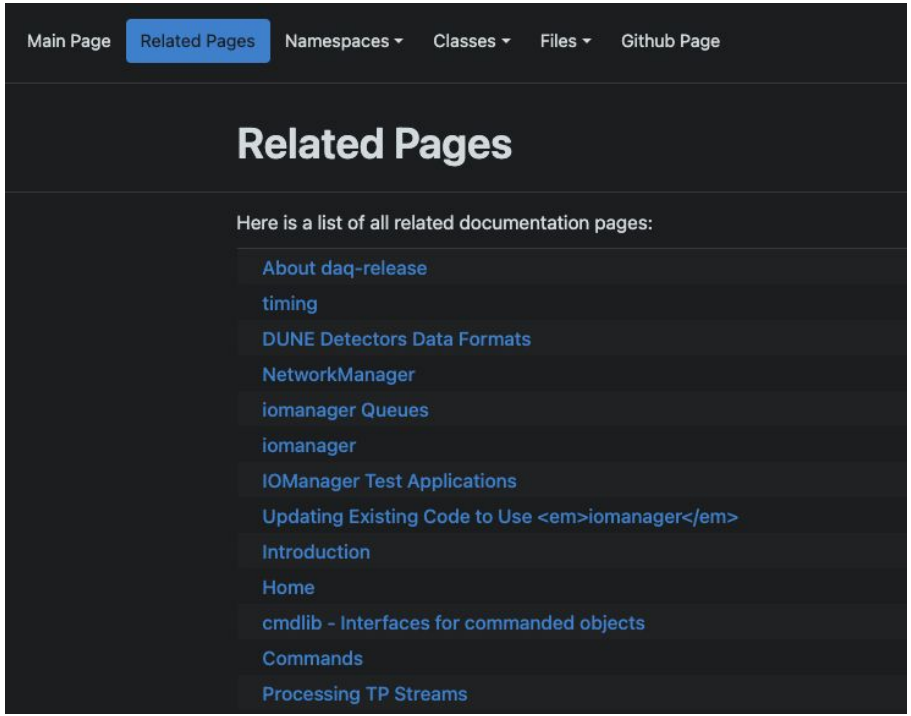
- It works!
  - …but could use some polishing



Fermilab

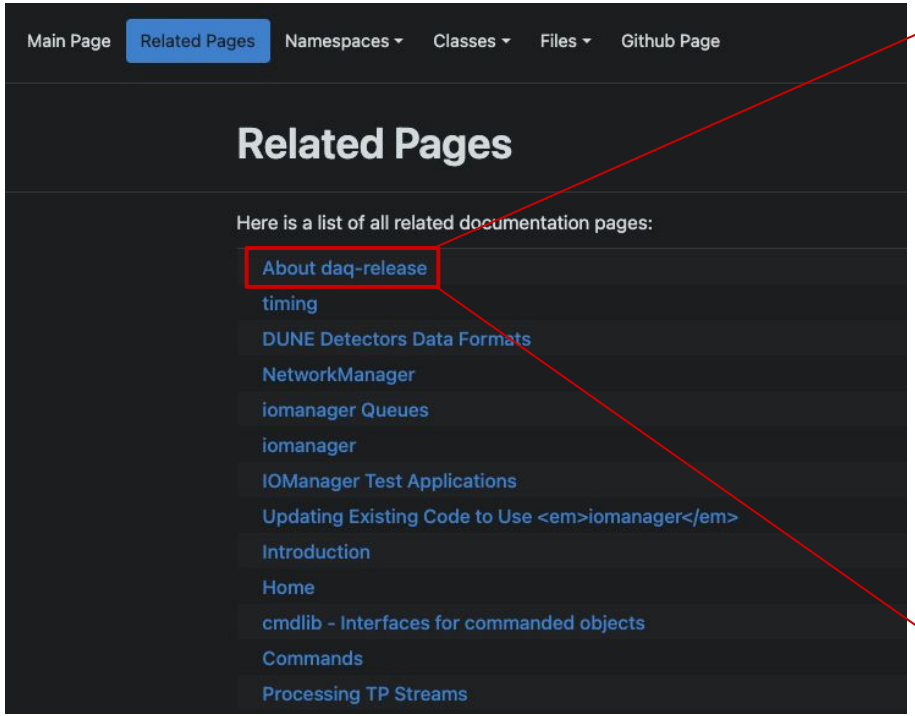
Related Pages

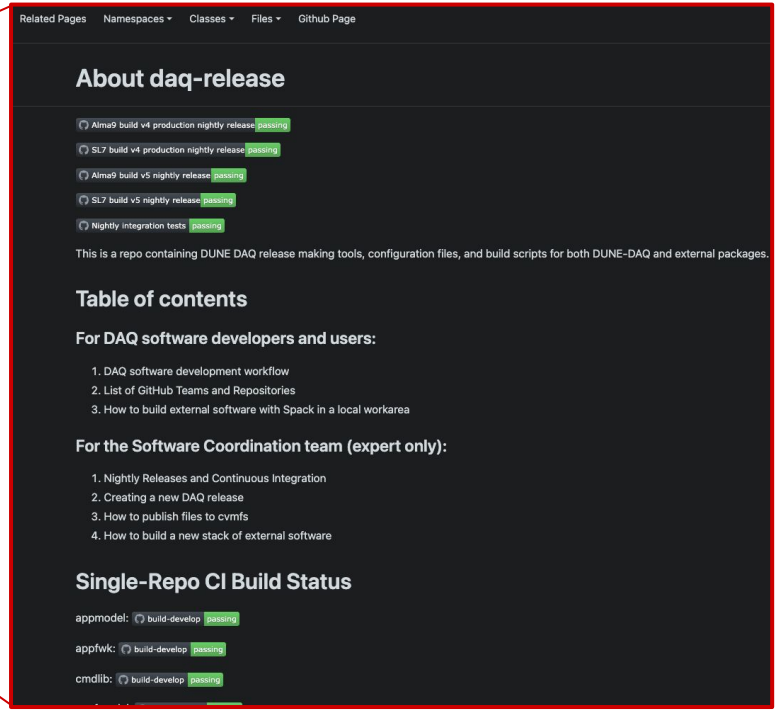
- Contains various links to READMEs and other documentation pages



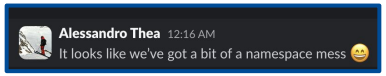
Related Pages

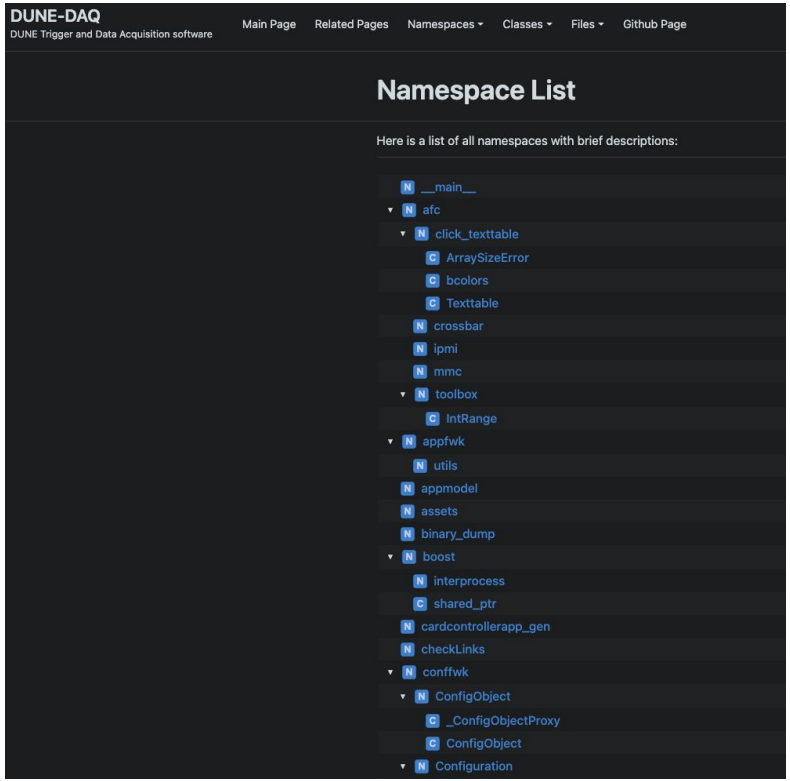
- Contains various links to READMEs and other documentation pages





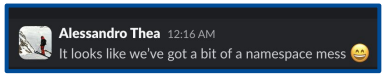
The Result – Namespaces

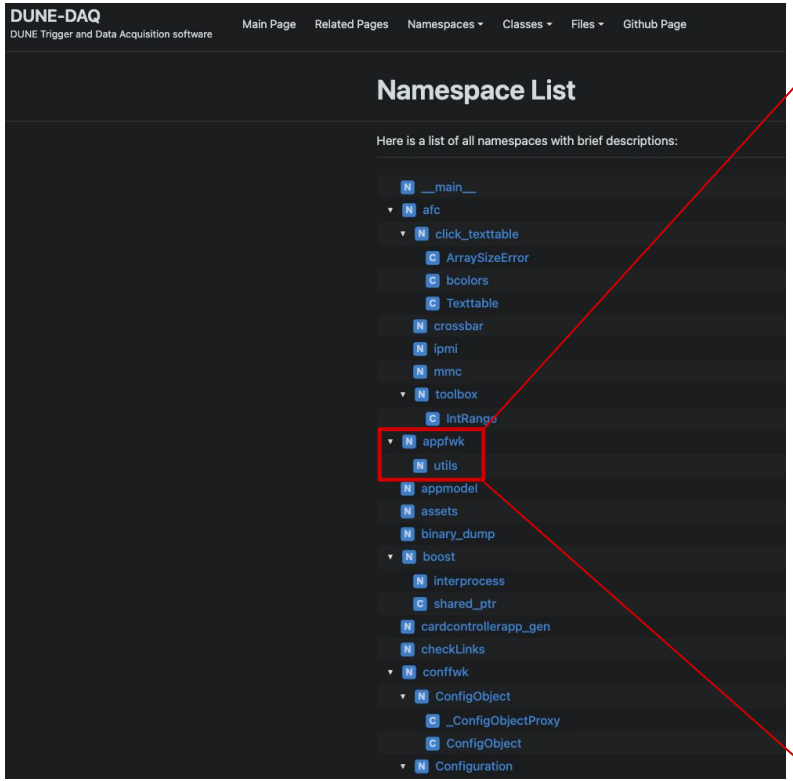


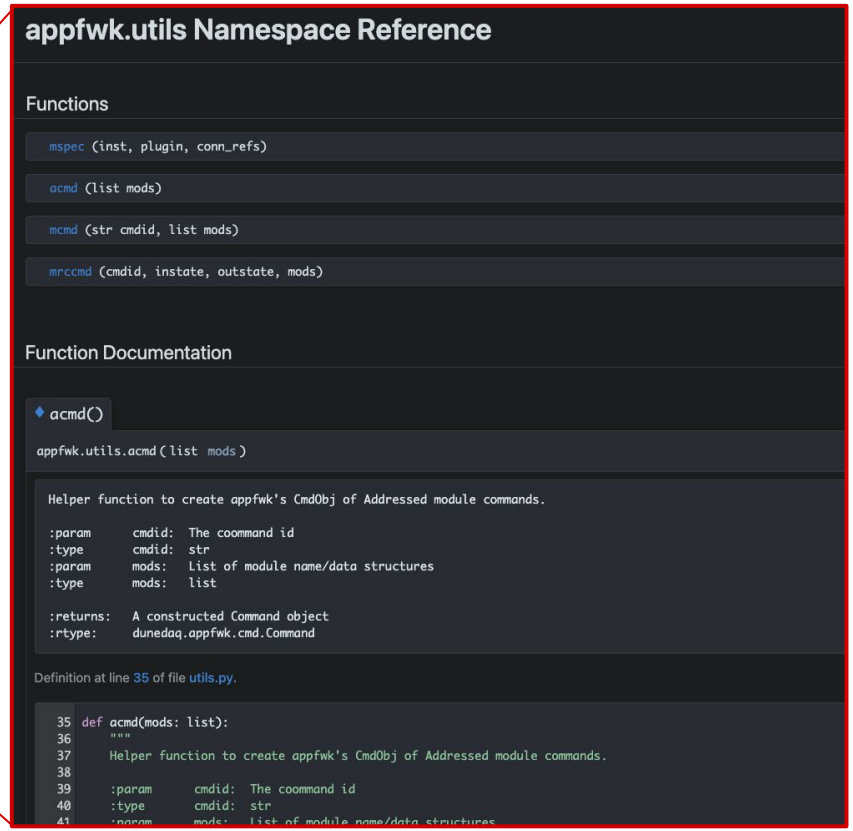


Fermilab

The Result – Namespaces



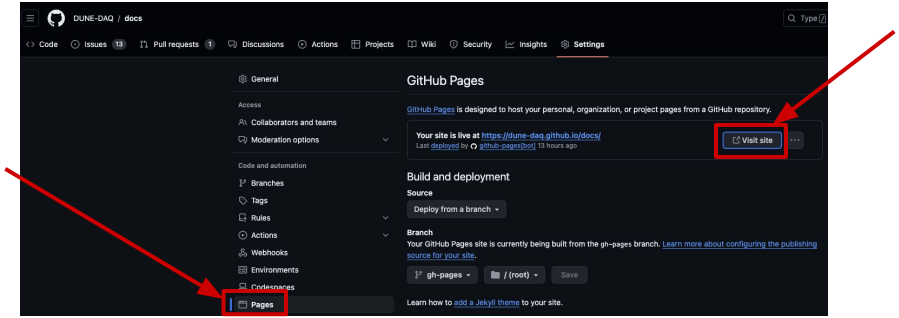




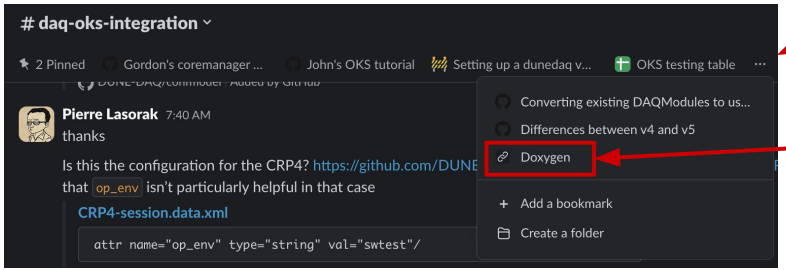
Fermilab

Getting to the Doxygen Page

- Option 1: The Hard Way
  - Go to docs repo in the DUNE DAQ organization GitHub, Settings -> Pages -> Visit Site



- Option 2: Bookmark in the #daq-oks-integration
  - May need to click the three dots to see it



Click the three dots

Fermilab

Summary

- The DUNE DAQ Doxygen page is now live on GitHub Pages and updated nightly
- Discussion points:
  - The page style and layout could be adjusted if people want
  - Where else should we put the link for maximum visibility?
  - Other thoughts?