



**STScI** | SPACE TELESCOPE  
SCIENCE INSTITUTE

EXPANDING THE FRONTIERS OF SPACE ASTRONOMY

# The JWST Quicklook Application (JWQL)

---

Matthew Bourque, Lauren Chambers, and the JWQL team

July 2019

# Team Members



Matthew Bourque  
(Technical Lead)



Francesca Boffi  
(Project Manager)



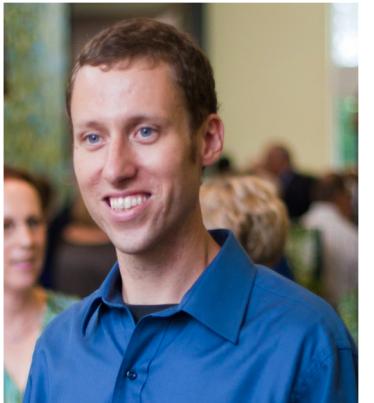
Lauren Chambers  
(Deputy Lead, FGS)



Misty Cracraft  
(MIRI)



Joseph Filippazzo  
(NIRISS)



Bryan Hilbert  
(NIRCam)



Graham Kanarek  
(NIRSpec)



Catherine Martlin  
(WFC3)



Johannes Sahlmann  
(NIRISS)



Ben Sunquist  
(NIRCam)



# Team Members



Matthew Bourque  
(Technical Lead)



Francesca Boffi  
(Project Manager)



Lauren Chambers  
(Deputy Lead, FGS)



Misty Cracraft  
(MIRI)



Joseph Filippazzo  
(NIRISS)



Bryan Hilbert  
(NIRCam)



Graham Kanarek  
(NIRSpec)



Catherine Martlin  
(WFC3)



Johannes Sahlmann  
(NIRISS)



Ben Sunquist  
(NIRCam)



# Acknowledgements

## INS

Tracy Beck

Rosa Diaz

Van Dixon

Scott Friedman

Alex Fullerton

Catherine Gosemeyer

Dean Hines

Sherie Holfeltz

Daniel Kuhbacher

Matt Lallo

Stephanie La Massa

Elena Sabbi

Brian O'Sullivan

Massimo Robberto

Linda Smith

## DMD

Faith Abney

Joshua Alexander

Anastasia Alexov

Sara Anderson

Clara Brasseur

Matthew Burger

Steven Crawford

James Davies

Tom Donaldson

Kim DuPrie

Jonathan Eisenhamer

Lisa Gardner

Joe Hunkeler

Catherine Kaleida

Jenn Kotler

Mark Kyprianou

Karen Levay

Jacob Matuskey

Christian Mesh

Sara Ogaz

Lee Quick

Matt Rendina

Mary Romelfanger

Bernie Shiao

Geoff Wallace

## ITSD

Vera Gibbs

Phil Grant

Greg Masci

Prem Mishra

Don Meuller

Anupinder Rai

Matthew Sienkiewicz

Patrick Taylor

Dave Unger

Thomas Walker

Joe Zahn

## SEITO

Maria Nieto-Santisteban

## SCOPE

Crystal Mannfolk

## OPO

Ann Feild

Lara Wilkinson

## DSMO

Michael Fox

Arfon Smith

## JWSTMO

Margaret Meixner

Joe Polizzi

Jeff Valenti



## Scope and Status

- JWQL is still under heavy development



## Scope and Status

- JWQL is still under heavy development
- Project is primarily for JWST operations
  - No explicit requirements for JWST commissioning
  - That being said, we welcome use of available functionality in our app during commissioning



## Scope and Status

- JWQL is still under heavy development
- Project is primarily for JWST operations
  - No explicit requirements for JWST commissioning
  - That being said, we welcome use of available functionality in our app during commissioning
- On track for 1.0 release in Fall/Winter 2019
- Additional features beyond 1.0 release



*A database-driven web application and automation framework for  
analyzing and monitoring the JWST instruments and their data*



*A database-driven web application and automation framework for  
analyzing and monitoring the JWST instruments and their data*

**Visually inspect new and archived data**



*A database-driven web application and automation framework for analyzing and monitoring the JWST instruments and their data*

**Visually inspect new and archived data**

**Automated calibration & monitoring software**



*A database-driven web application and automation framework for analyzing and monitoring the JWST instruments and their data*

**Visually inspect new and archived data**

**Automated calibration & monitoring software**

**Convenience of a web browser**



*A database-driven web application and automation framework for analyzing and monitoring the JWST instruments and their data*

**Visually inspect new and archived data**

**Automated calibration & monitoring software**

**Convenience of a web browser**

**One solution for five teams**



# System Design



# System Design



MAST Cache



# System Design



MAST Cache



JWQLDB



Astroquery



JWST EDB

# System Design



MAST Cache



JWQLDB

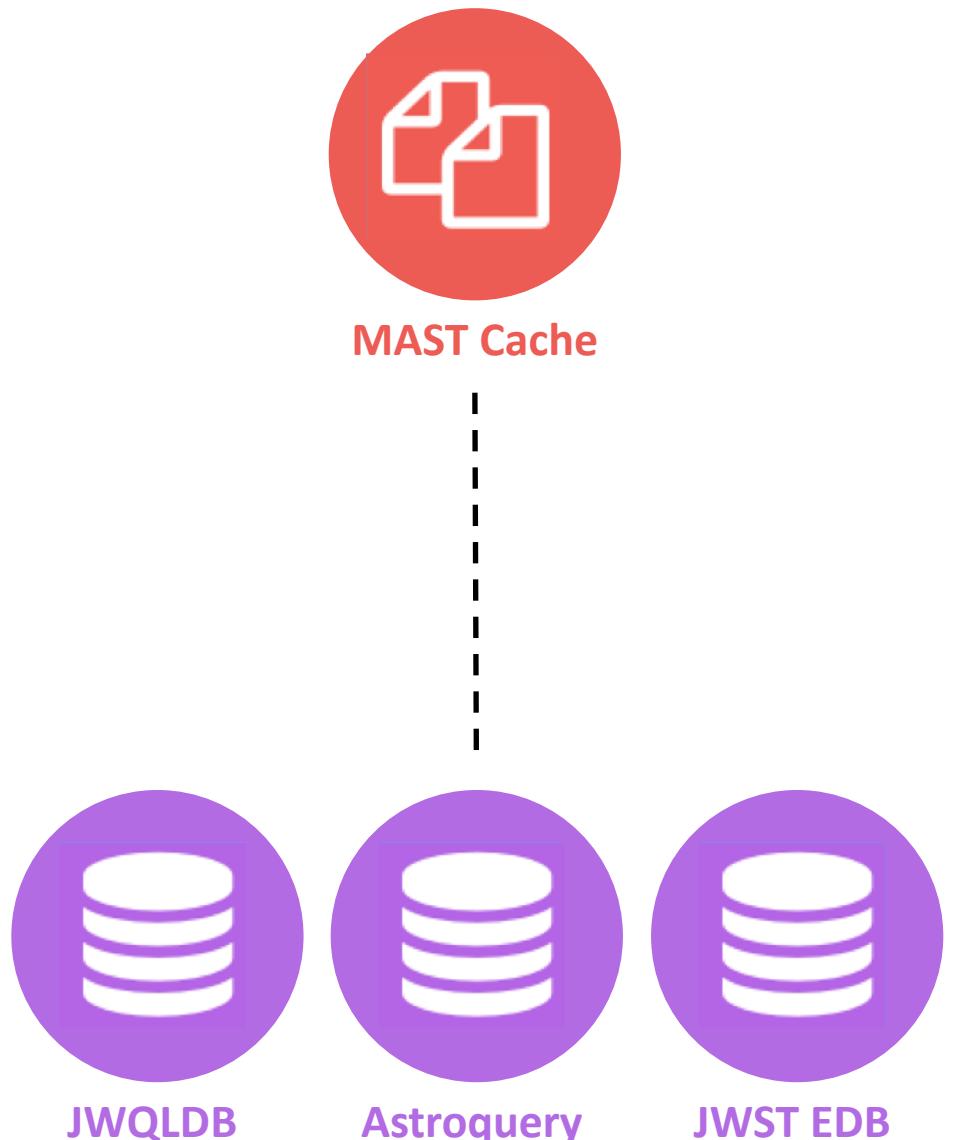


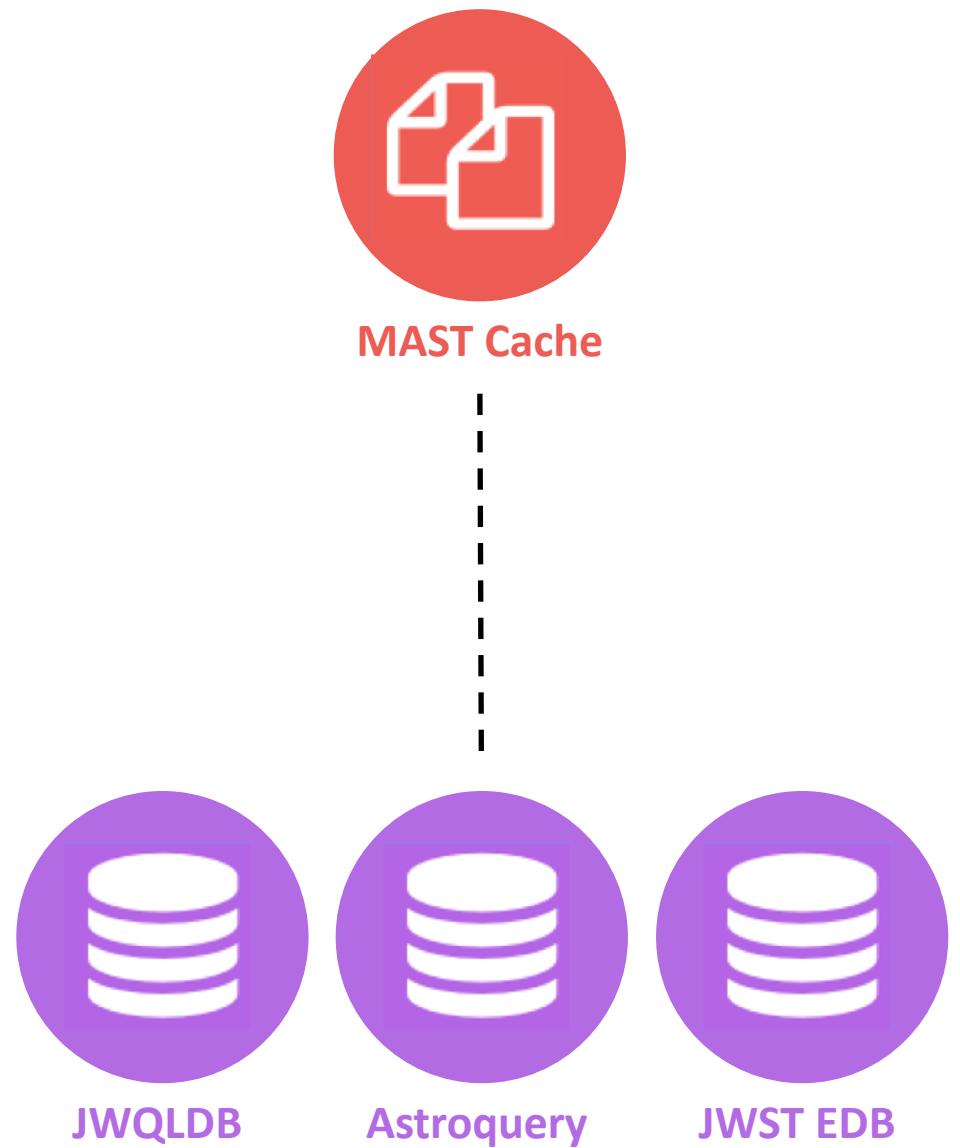
Astroquery



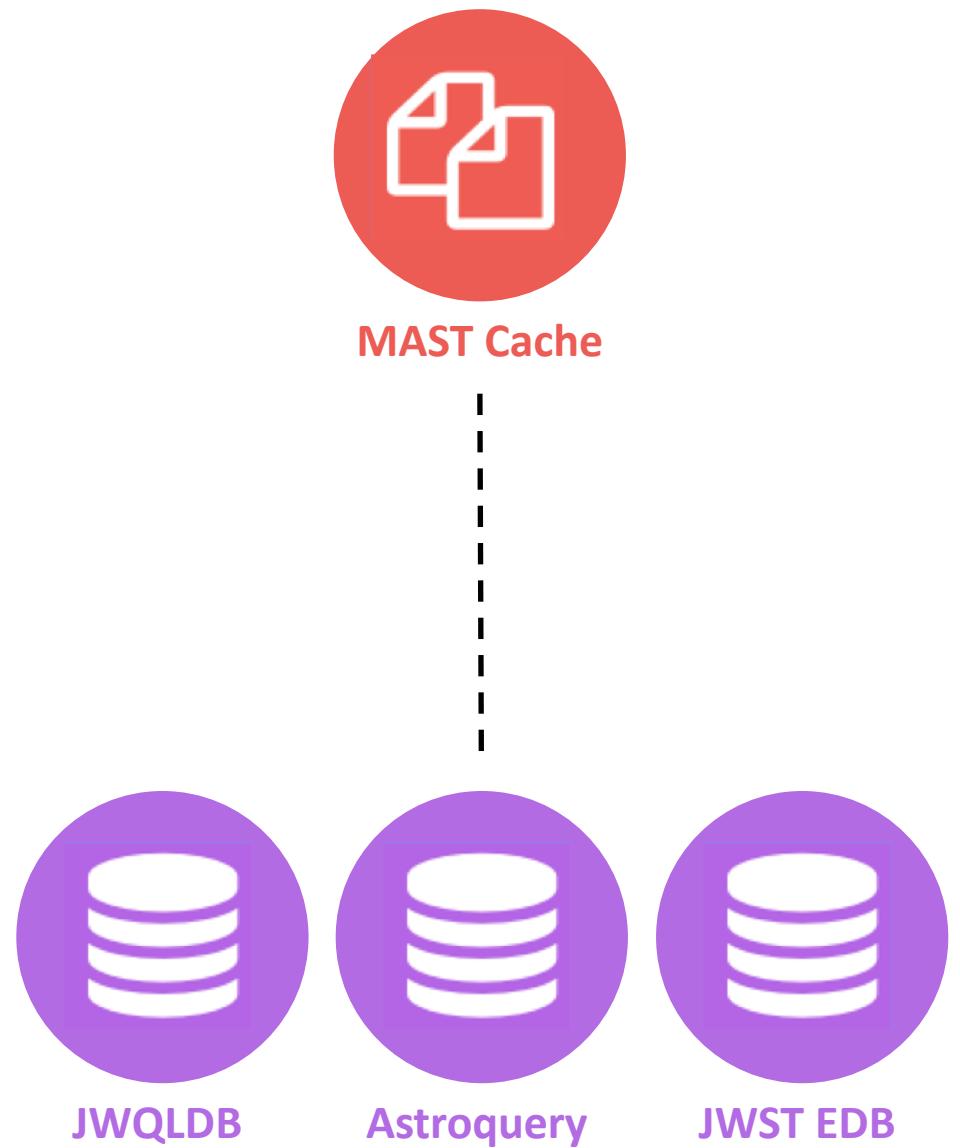
JWST EDB

# System Design

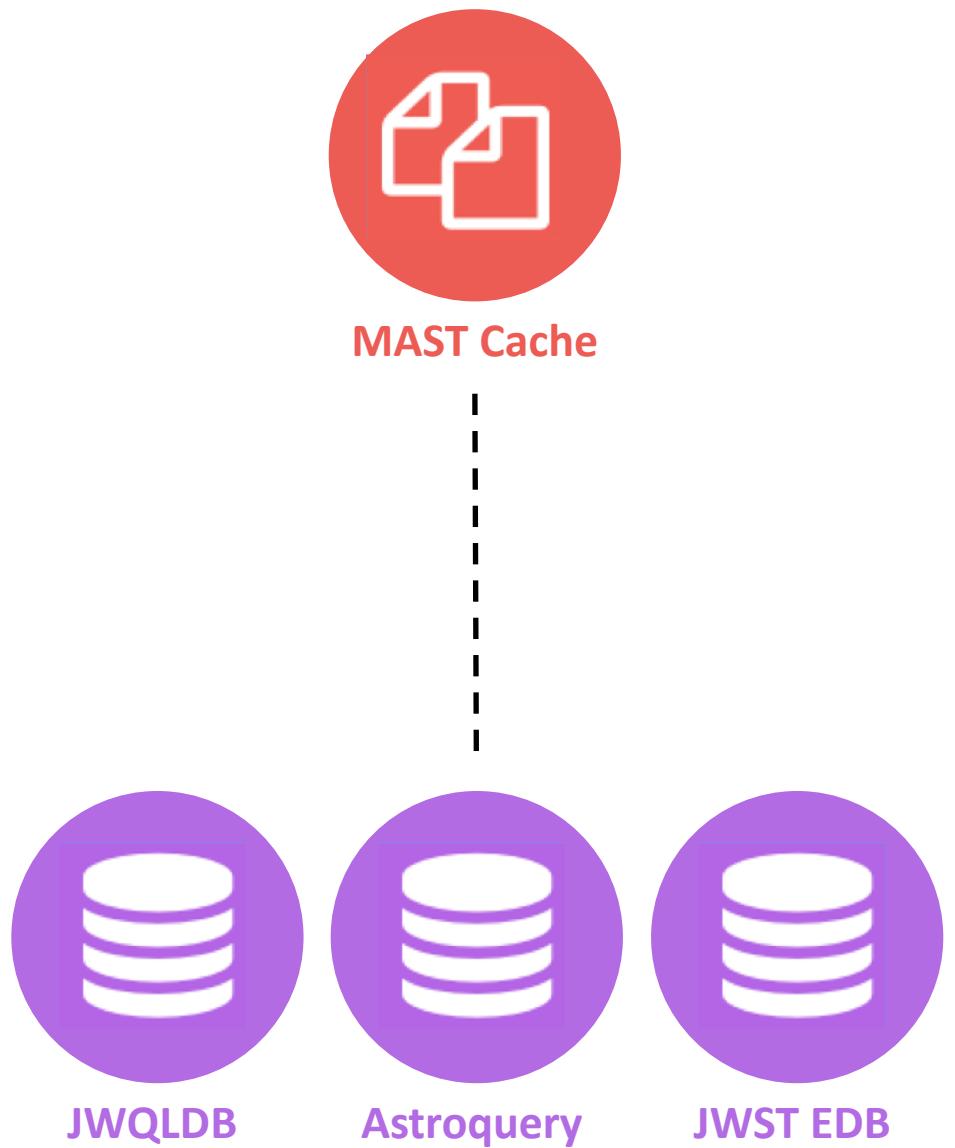




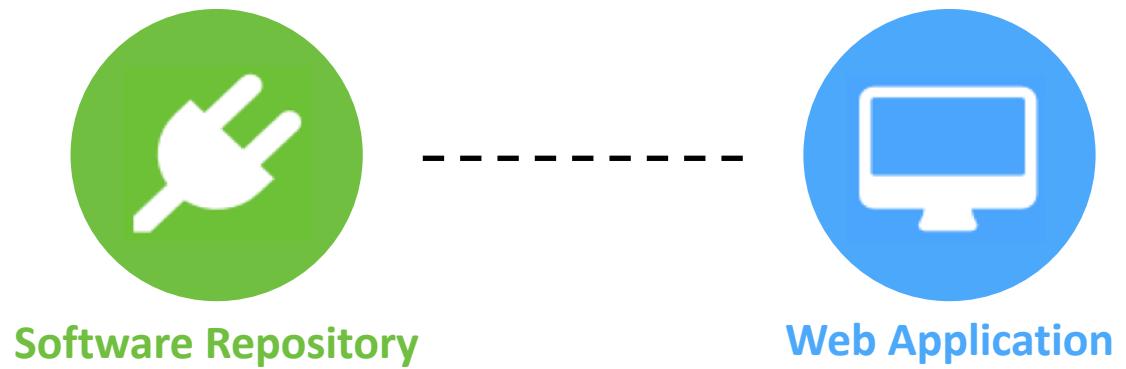
## System Design

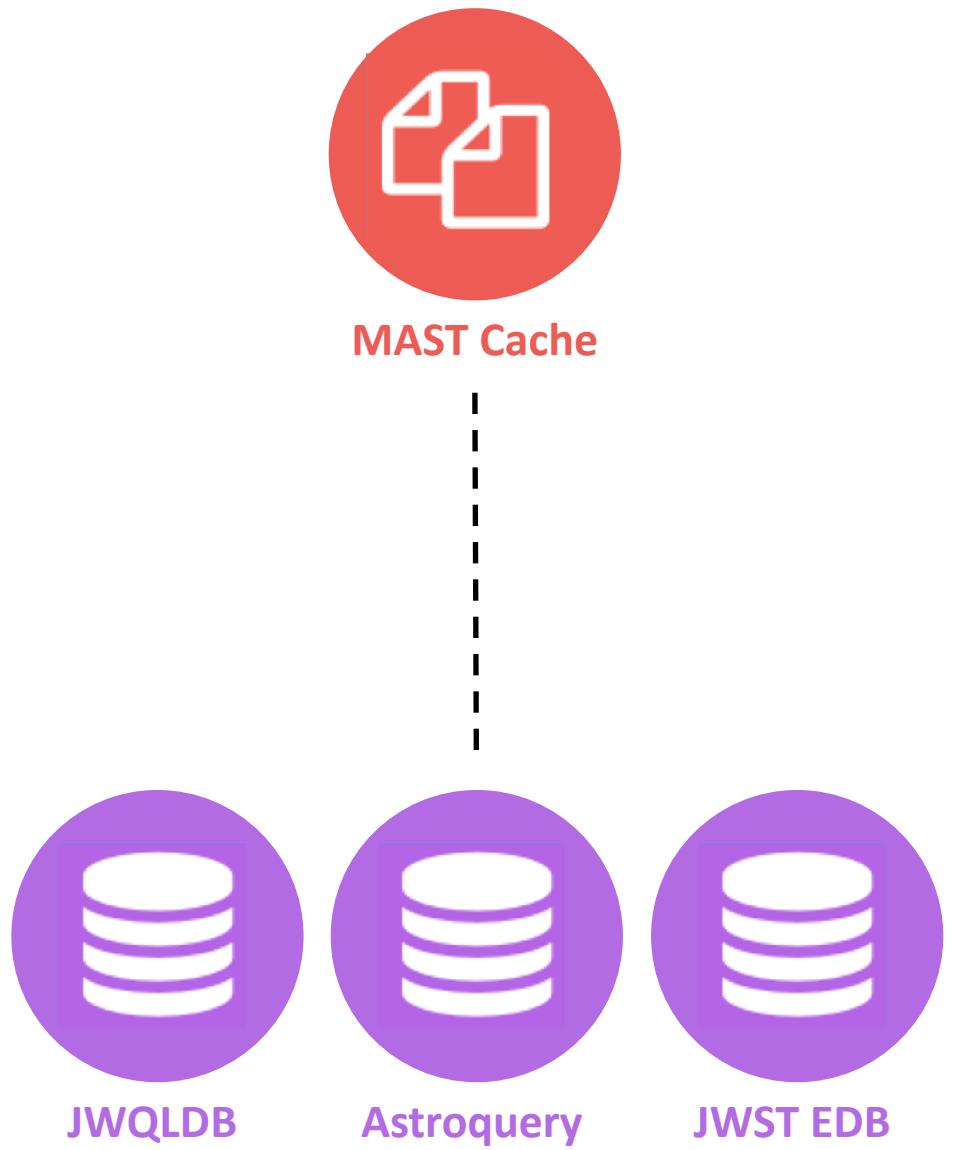


## System Design

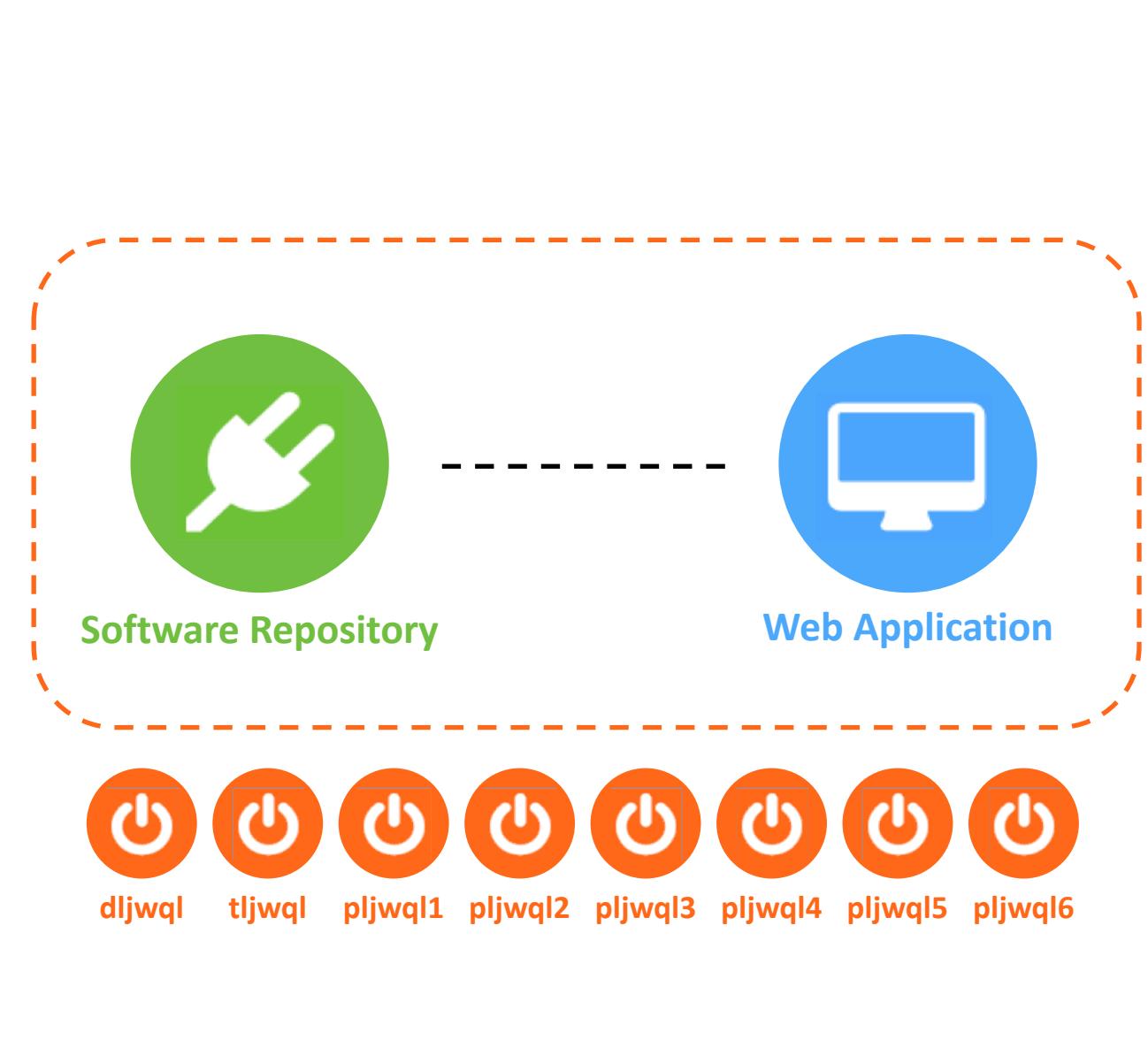


# System Design





## System Design





# Technologies





# Building A Strong Foundation for Software Projects



# Building A Strong Foundation for Software Projects

## Proposal/Requirements Document

### Implementing the JWST Quicklook: A Proposal for the Webb Mission Office

**INS/JWST Quicklook working group:** Francesca B  
Donaldson (consulting), Joe Filippazzo, Michael Fox  
Gosmeyer, Bryan Hilbert, Sherie Holfeltz, Gray Kanare  
(consulting), Catherine Martlin, Ed Nelan, Johannes S  
(consulting)

**INS/JWST Teams Management:** Tracy Beck, Rosa D  
Massimo Robb

**INS Division Office:** Linda Smith, Fran

DATE OF CURRENT VERS

#### Table of Contents

1. Technical Objectives
  - a. General Justification
  - b. NIRCam Justification
  - c. NIRISS Justification
  - d. NIRSpec Justification
  - e. MIRI Justification
  - f. FGS Justification
2. Technical Concept
  - a. Computing Environments
  - b. Filesystem (or access to MAST cache)
  - c. Database Schema (or MAST database view)
  - d. Web Application
  - e. Code Library
  - f. Testing Software Integration
3. Task Labor Categories and Schedules
  - a. Areas of Development
  - b. Schedule
4. Related Research & Development
5. Key Personnel and Consultants
6. Resource Estimates
  - a. FTE Resources
  - b. Computing Resources
7. Beyond the JWST Quicklook



# Building A Strong Foundation for Software Projects

## Proposal/Requirements Document

### Style Guide

The screenshot shows a GitHub repository page for 'jwql/README.md at develop'. The page title is 'jwql Style Guide'. The content includes an introduction about the style guide, sections for 'Prerequisite Reading' (listing PEP8, PEP257, and numpydoc), 'Workflow' (describing CI workflow and flake8), 'Version Numbers and Tags' (explaining the x.y.z convention with a note about API changes), and 'Security' (warning against committing sensitive items). The GitHub interface shows standard navigation and commit history controls.

**jwql Style Guide**

This document serves as a style guide for all `jwql` software development. Any requested contribution to the `jwql` code repository should be checked against this guide, and any violation of the guide should be fixed before the code is committed to the `master` or `develop` branch. Please refer to the accompanying [example.py](#) script for a example code that abides by this style guide.

### Prerequisite Reading

It is assumed that the reader of this style guide has read and is familiar with the following:

- The [PEPB Style Guide for Python Code](#)
- The [PEP257 Docstring Conventions Style Guide](#)
- The [numpydoc docstring](#) convention

### Workflow

All software development for the `jwql` project should follow a continuous integration workflow, described in the [git & GitHub workflow for contributing](#). Before committing any code changes, use `flake8` to check the code against PEP8 standards. Also check that your code is conforming to this style guide.

### Version Numbers and Tags

Any changes pushed to the `master` branch should be tagged with a version number. The version number convention is `x.y.z`, where

`x` = The main version number. Increase when making incompatible API changes.  
`y` = The feature number. Increase when change contains a new feature with or without bug fixes.  
`z` = The hotfix number. Increase when change only contains bug fixes.

### Security

The following items should never be committed in the `jwql` source code or GitHub issues/pull requests:



# Building A Strong Foundation for Software Projects

## Proposal/Requirements Document

## Style Guide

## Code of Conduct

The screenshot shows a GitHub browser interface with the following details:

- Title:** jwql/CODE\_OF\_CONDUCT.md
- Repository:** GitHub, Inc. [US] | https://github.com/spacetelescope/jwql/blob/develop/CODE\_OF\_CONDU...
- File Statistics:** 62 lines (39 sloc) | 5.93 KB
- Actions:** Raw, Blame, History, Print, Edit, Delete

The content of the file is as follows:

### jwql Code of Conduct

---

#### Introduction

This code of conduct applies to all spaces related to the `jwql` project, including all public and private mailing lists, repositories, issue trackers, wikis, and any other communication channel used by our community.

This code of conduct should be honored by everyone who participates in the `jwql` project formally or informally, or claims any affiliation with the project, in any project-related activities and especially when representing the project, in any role.

This code is not exhaustive or complete. It serves to distill our common understanding of a collaborative, shared environment and goals. Please try to follow this code in spirit as much as in letter, to create a friendly and productive environment that enriches the surrounding community.

#### Our Pledge

---

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to:

- Treat all people with respect and provide a harassment and bullying-free environment, regardless of sex, sexual orientation, gender identity, disability, physical appearance, body size, race, nationality, ethnicity, and religion. In particular, sexual language and imagery, sexist, racist, or otherwise exclusionary language are not appropriate.
- Respect the work of others by recognizing acknowledgment/citation requests of original authors. As authors, we pledge to be explicit about how we want our own work to be cited or acknowledged.
- Welcome and invite those interested in joining the community, and realize that including people with a variety of opinions and backgrounds will only serve to enrich our community. In particular, discussions relating to pros/cons of various technologies, programming languages, and so on are welcome, but these should be done with respect, taking proactive measure to ensure that all participants are heard and feel confident that they can freely express their opinions.
- Use public methods of communication for project-related messages, unless discussing sensitive materials. This applies to messages for help or project-related support, too; not only is a public support request much more likely to result in an answer to a question, it also ensures that any inadvertent mistakes in answering are more easily detected and corrected.
- Welcome questions and answer them respectfully, paying particular attention to those new to the community. We



# Building A Strong Foundation for Software Projects

Proposal/Requirements Document

Style Guide

Code of Conduct

Conda Environment

A screenshot of a GitHub browser window displaying the contents of the 'environment\_python\_3.6.yml' file. The window title is 'jwql/environment\_python\_3.6.yml'. The file shows a YAML configuration for a Conda environment, listing various Python packages and their versions. The file has 39 lines (38 sloc) and is 668 Bytes.

```
channels:
- http://ssb.stsci.edu/astroconda-dev
defaults:
dependencies:
- asdf=2.3.1
- astropy>=3.1.2
- astroquery=0.3.9
- bokeh=1.2.0
- crds>=7.2.7
- django=2.2.1
- inflection=0.3.1
- ipython=7.6.0
- jinja2=2.10
- jsonschema=3.0.1
- jwst=0.13.1
- matplotlib=3.0.2
- numpy=1.16.4
- numpydoc=0.9.0
- pandas=0.24.2
- postgresql=9.6.6
- psycopg2=2.7.5
- python=3.6.4
- python-dateutil=2.7.5
- pytest=4.6.2
- pytest-cov=2.6.1
- pytest-html=1.19.0
- sphinx=2.1.0
- sphinx_rtd_theme=0.1.9
- sqlalchemy=1.3.4
- sqlparse=0.3.0
- stsci_rtd_theme=0.0.2
- twine=1.13.0
- pip:
- authlib==0.10
- codecov==2.0.15
- jwedb>=0.0.3
- pysiaf==0.3.0
- sphinx-autodocapi==0.10
```



# Building A Strong Foundation for Software Projects

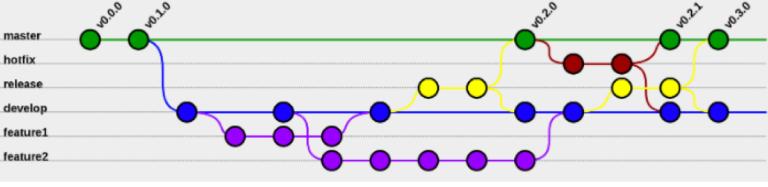
Proposal/Requirements Document  
Style Guide  
Code of Conduct  
Conda Environment  
**git Workflow**

git & GitHub workflow for contributing

Matthew Bourque edited this page on Mar 20 · 12 revisions

The best method for contributing software to the `jwql` project is a workflow that involves forking the `jwql` repository, developing changes on "feature" branches, and opening pull requests through GitHub.

Note that the `jwql` team employs a release-driven `git` workflow, as shown in the following diagram:



As such, all feature branches should be branched off of and merged back into the `develop` branch.

### Contributing new features or bug fixes

The first question you will have to figure out is whether you should open an issue for this update - if you think that this change will be solving a significant problem or add a significant enhancement to the project then it would be advantageous to open an issue ticket [here](#). This will allow both individuals and the team as a whole to keep track of the project and our progress as we go. Any appropriate individuals should be assigned to the issue, and a label(s) should be tagged.

Following that, any changes that you want to eventually make to the `master` or `develop` branch should be done through the workflow where you create a fork and work on your own branch before submitting those changes to be reviewed through a pull request. Instructions on how to do those things can be found below. Note that these instructions are for interacting with git through the command line, however, there are alternatives with graphical user interfaces such as [sourcetree](#).

1. Create a personal fork of the `jwql` repository by visiting the [spacetelescope repository](#) and clicking the `Fork` button. Note that this only has to be done once.
2. Make a local copy of your personal fork by cloning the repository, using the URL found by

Pages 14

Find a Page...

Home

Adding Sphinx API documentation

Checklist for Contributors and Reviewers of Pull Requests

Config file

Environment Installation

git & GitHub workflow for contributing

JWQL Logging Function: The Why's and The How

Running the web app locally

Software Releases

Style Guide

Suggestions and discussion items

Useful Links

Web Application

Web Server

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/spacetelescope/jwql/wiki/git-%26-GitHub-workflow-for-contributing>



# Building A Strong Foundation for Software Projects

Proposal/Requirements Document

Style Guide

Code of Conduct

Conda Environment

git Workflow

## Project Management Workflow

The screenshot shows a GitHub project board for the repository `spacetelescope/jwql`. The board is set up with three columns: **In progress**, **In Review**, and **Done**.

- In progress:** Contains 4 items.
  - Specify Dark Monitor Plots**: High Priority, Monitor, Web Application, enhancement.
  - Expand filesystem monitor to keep track of JWQL central storage area**: Low Priority, Monitor, enhancement.
  - Integrate Bokeh into the Django backend**: Medium Priority, Package, Web Application, enhancement.
  - Build web app view for displaying monitor log files**: Low Priority, bug.
- In Review:** Contains 1 item.
  - Ensure web application is accessible!**: Low Priority, Web Application, enhancement.
- Done:** Contains 15 items.
  - add config.json format verification?**: Low Priority, bug, enhancement.
  - Import environment bokeh version in HTML**: Environment, Low Priority, Web Application, enhancement.
  - Add more inline comments to the Jenkinsfile**: Documentation.
  - Fix Jenkins**: High Priority, Testing, bug.
  - Changes approved**.

At the bottom of the board, it says "Automated as In progress" and "Manage".



# Building A Strong Foundation for Software Projects

Proposal/Requirements Document

Style Guide

Code of Conduct

Conda Environment

git Workflow

Project Management Workflow

**Code Review**

Update web app to use django 2.2 #448

Merged bourque merged 2 commits into [spacetelescope:develop](#) from [bourque:django-2.2](#) 20 days ago

Conversation 4 Commits 2 Checks 0 Files changed 3 +18 -4

bourque commented 24 days ago Member + ...

This PR provides various changes to get the web app working with `django` version 2.2. The new version requires the existence of a define `django.template.backends.django.DjangoTemplates TEMPLATE`. Also, some `not_found` requests now also require additional parameters such as `exception`, which have been wrapped up in a `*args` parameter.

Updates needed to get django version 2.2 working efa92ed ✓

bourque added enhancement Web Application Environment Medium Priority labels 24 days ago

bourque requested a review from laurenmarietta 24 days ago

bourque self-assigned this 24 days ago

Reviewers laurenmarietta ✓

Assignees bourque

Labels Environment Medium Priority Web Application enhancement

Projects None yet

Milestone



# Building A Strong Foundation for Software Projects

Proposal/Requirements Document

Style Guide

Code of Conduct

Conda Environment

git Workflow

Project Management Workflow

**Code Review**

Update web app to use django 2.2 #448

Merged bourque merged 2 commits into spacetelescope:develop from bourque:django-2.2 20 days ago

Conversation 4 Commits 2 Checks 0 Files changed 3 +18 -4

bourque commented 24 days ago Member +  
This PR provides various changes to get the web app working with django version 2.2.  
The new version requires the existence of a define  
django.template.backends.django.DjangoTe  
also require additional parameters such as ex  
parameter.

Updates needed to get django vers  
bourque added enhancement Web Ap  
bourque requested a review from laur  
bourque self-assigned this 24 days ago

Reviewers laurenmarietta Assignees

pep8speaks commented 24 days ago • edited Hello @bourque, Thank you for updating !  
• In the file jwql/website/apps/jwql/views.py :  
Line 374:101: E501 line too long (114 > 100 characters)

Comment last updated at 2019-06-25 18:44:48 UTC

Milestone



# Building A Strong Foundation for Software Projects

Proposal/Requirements Document

Style Guide

Code of Conduct

Conda Environment

git Workflow

Project Management Workflow

**Code Review**

The image shows two screenshots of software development tools. The top screenshot is a GitHub pull request titled "Update web app to use django 2.2 #448". It is marked as "Merged" by "bourque" 20 days ago. The PR has 4 conversations, 2 commits, 0 checks, and 3 files changed. The code review comments include:

- "bourque commented 24 days ago": "This PR provides various changes to get the web app working with django version 2.2. The new version requires the existence of a define django.template.backends.django.DjangoTemplates also require additional parameters such as ex parameter."
- "pep8speaks commented 24 days ago • edited": "Hello @bourque, Thank you for updating !"

The bottom screenshot is a Codecov Report for pull request #448. It shows coverage statistics and impacted files. The coverage diff is +0.14%.

**Codecov Report**

Merging #448 into develop will increase coverage by 0.14%.  
The diff coverage is 0%.

@@	Coverage	Diff	@@
##	develop	#448	+/- ##
=====			
+ Coverage	20.73%	20.88%	+0.14%
=====			
Files	62	62	
Lines	4013	4013	
=====			
+ Hits	832	838	+6
+ Misses	3181	3175	-6

**Impacted Files**

Impacted Files	Coverage Δ
jwql/website/jwql_proj/settings.py	0% <0> (ø)
jwql/website/apps/jwql/views.py	0% <0%> (ø)
jwql/tests/test_api_views.py	92% <0%> (+12%)



# Building A Strong Foundation for Software Projects

Proposal/Requirements Document

Style Guide

Code of Conduct

Conda Environment

git Workflow

Project Management Workflow

Code Review

**API Documentation**

The screenshot shows a web browser displaying the "Welcome to the jwql API documentation!" page. The page features the JWQL logo at the top right. The main content area includes a large "Welcome to the **jwql** API documentation!" heading, a brief description of the JWST Quicklook Application, and links to "API documentation", "Contents", and "Indices and tables". On the left side, there is a sidebar with a "CONTENTS:" list and a sponsored advertisement for learning intermediate Python.

**CONTENTS:**

- database
- edb
- jwql\_monitors
- tests
- utils
- website

**Welcome to the **jwql** API documentation!**

The JWST Quicklook Application (**jwql**) is a database-driven web application and automation framework for JWST instrument monitoring and performance trending. This site serves as the official API documentation for the project. The **jwql** website can currently be found at <https://djwjql.stsci.edu>. The **jwql** source code, installation instructions, and instructions for issue reporting and feature requests can be found in the [jwql GitHub repository](#).

**API documentation**

**Contents:**

- database
- edb
- jwql\_monitors
- tests
- utils
- website

**Indices and tables**

- Index



# Building A Strong Foundation for Software Projects

Proposal/Requirements Document

Style Guide

Code of Conduct

Conda Environment

git Workflow

Project Management Workflow

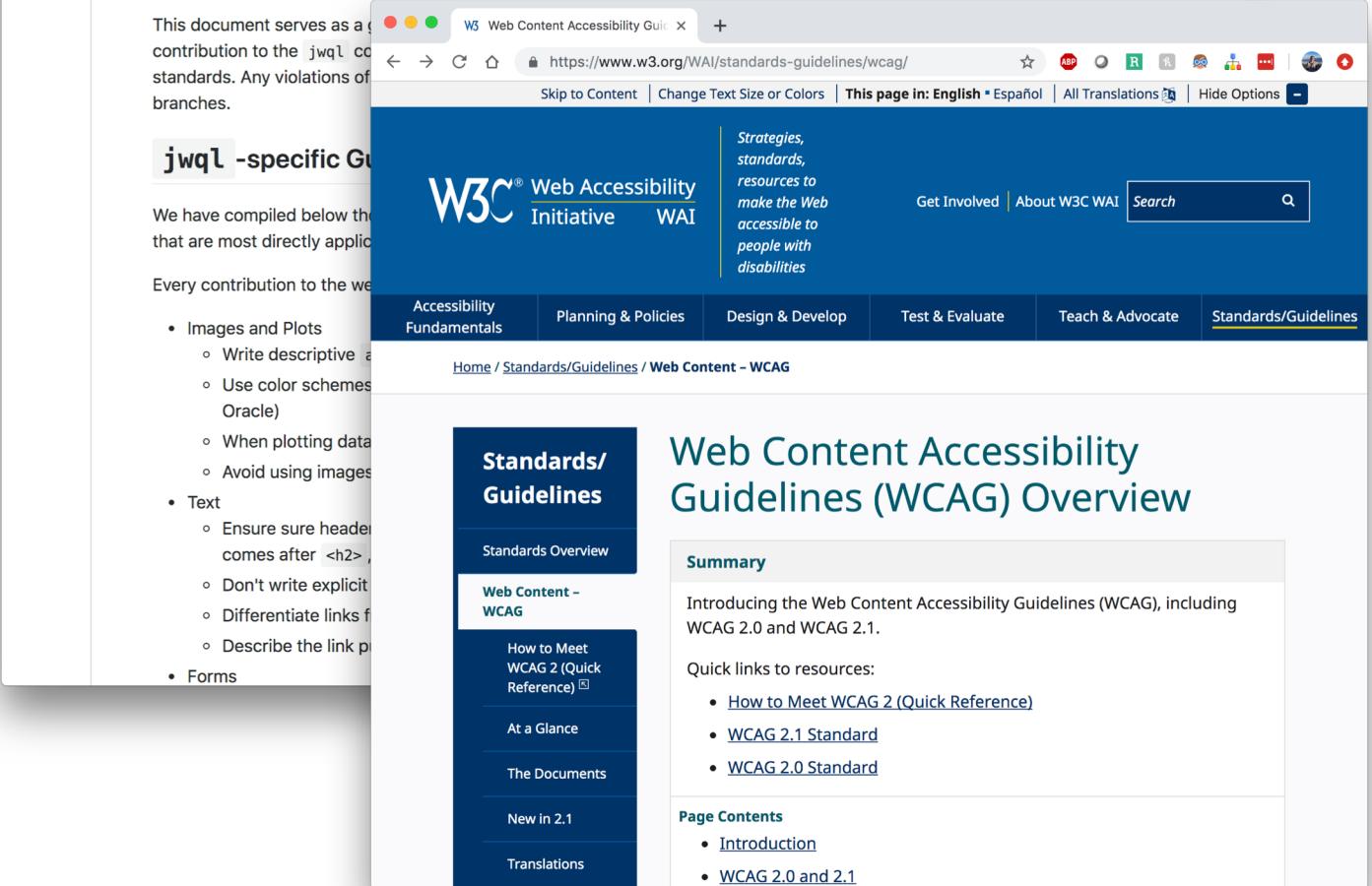
Code Review

API Documentation

**Website Accessibility**



A screenshot of a GitHub repository page titled "jwql/accessibility\_guidelines.md". The page shows 52 lines (37 sloc) and 3.93 KB. The content is a copy of the WCAG guidelines.



The screenshot shows the WCAG homepage with the title "Web Content Accessibility Guidelines (WCAG) Overview". It features a sidebar with links like "Standards/Guidelines", "Web Content - WCAG", "How to Meet WCAG 2 (Quick Reference)", "At a Glance", "The Documents", "New in 2.1", and "Translations". The main content area discusses the introduction to WCAG 2.0 and 2.1, quick links to resources, and page contents.



# Building A Strong Foundation for Software Projects

Proposal/Requirements Document

Style Guide

Code of Conduct

Conda Environment

git Workflow

Project Management Workflow

Code Review

API Documentation

Website Accessibility

**Contribution Guide**

The page contains the following sections:

- Software Contributions**:  
There are two current pages to review before you begin contributing to the `jwql` development. The first is our [style guide](#) and the second is our [suggested git workflow page](#), which contains an in-depth explanation of the workflow.  
Contributors are also encouraged to check out the [Checklist for Contributors Guide](#) to ensure the pull request contains all of the necessary changes.  
The following is a bare-bones example of a best work flow for contributing to the project:
  1. Create a fork off of the `spacetelescope jwql` repository.
  2. Make a local clone of your fork.
  3. Ensure your personal fork is pointing `upstream` properly.
  4. Create a branch on that personal fork.
  5. Make your software changes.
  6. Push that branch to your personal GitHub repository (i.e. `origin`).
  7. On the `spacetelescope jwql` repository, create a pull request that merges the branch into `spacetelescope:develop`.
  8. Assign a reviewer from the team for the pull request.
  9. Iterate with the reviewer over any needed changes until the reviewer accepts and merges your branch.
  10. Delete your local copy of your branch.
- Issue Reporting / Feature Requests**:  
Users who wish to report an issue or request a new feature may do so through the following channels:
  1. Submit a new issue on GitHub (preferred method): <https://github.com/spacetelescope/jwql/issues>
  2. Submit a new ticket on Jira: <https://jira.stsci.edu/projects/JWQL/>
- Code of Conduct**:  
Users and contributors to the `jwql` repository should adhere to the [Code of Conduct](#). Any issues or violations pertaining to the Code of Conduct should be brought to the attention of a `jwql` team member or to `jwql@stsci.edu`.



## Next Steps

- On track to complete four monitors by end of FY19:
  - Engineering Database Mnemonics (Complete!)
  - Dark Monitor (Complete!)
  - Bias Monitor
  - Target Acquisition Monitor
- Production web server (<https://jwql.stsci.edu>)



jwql@stsci.edu



<https://github.com/spacetelescope/jwql>



<https://jwql.readthedocs.io>



JWQLPROJ