

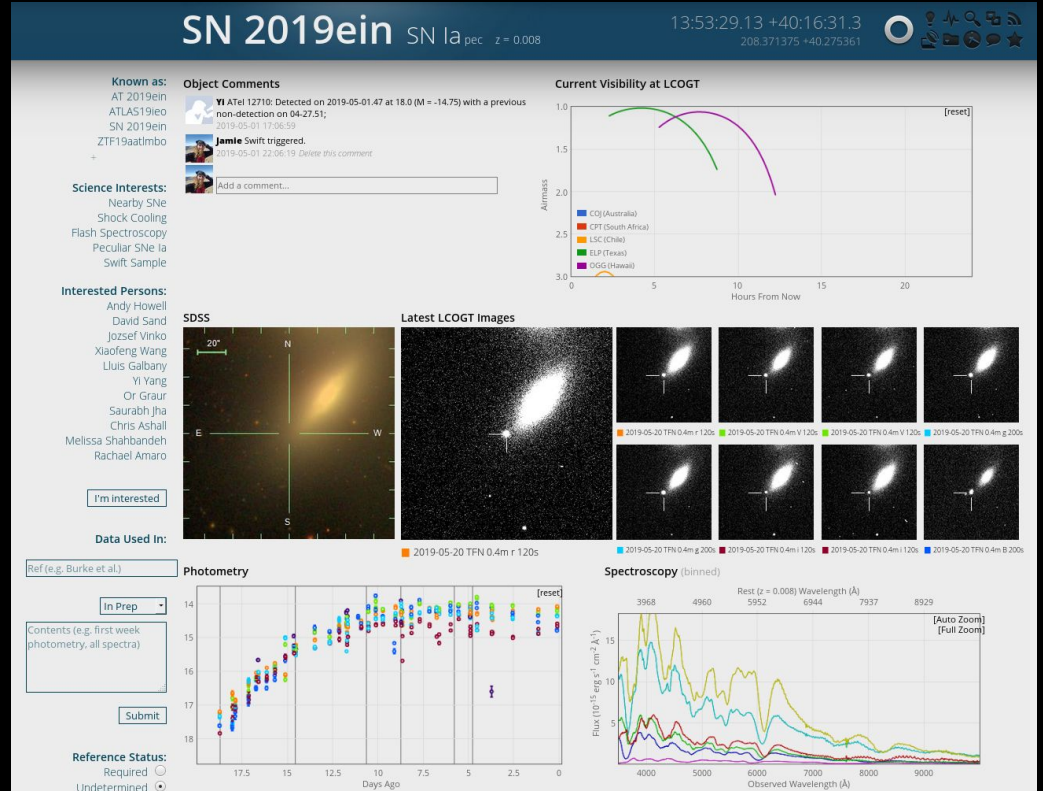
# Using the TOM Toolkit to Upgrade the Supernova Exchange

-Jamison Burke



# Background: the Supernova Exchange (SNEx)

- Proto-TOM used by the SN group
  - Written by former post-doc
  - Web interface (<https://supernova.exchange>)
  - Manages targets and observations



# Background: the Supernova Exchange (SNEEx)

## Pros:

- SN group could not function without it
- Front-end ease-of-use facilitates international collaboration
- Battle-tested and working

# Background: the Supernova Exchange (SNEEx)

## Pros:

- SN group could not function without it
- Front-end ease-of-use facilitates international collaboration
- Battle-tested and working

## Cons:

- Lead developer left and (almost) all development stopped
- Intuitive front-end but un-intuitive back-end with no documentation
  - 868,310 lines of code in 2,256 files 🤖
  - From-scratch mixture of Python, Javascript, and HTML
  - I still don't know how it actually works

# Background: the Supernova Exchange (SNEEx)

## Pros:

- SN group could not function without it
- Front-end ease-of-use facilitates international collaboration
- Battle-tested and working

## Cons:

- Lead developer left and (almost) all development stopped
- Intuitive front-end but un-intuitive back-end with no documentation
  - 868,310 lines of code in 2,256 files 🤖
  - From-scratch mixture of Python, Javascript, and HTML
  - I still don't know how it actually works

Upgrade SNEEx  
using the TOM  
Toolkit!

# Present: SNEEx 2 Video Walkthrough

[https://www.youtube.com/watch?time\\_continue=1&v=scNkmQBjsgE](https://www.youtube.com/watch?time_continue=1&v=scNkmQBjsgE)

# Future: Lines of Code Comparison

Old SNEEx:

-2,256 files

-868,130 lines of code

# Future: Lines of Code Comparison

Old SNEEx:

-2,256 files

-868,130 lines of code

SNEEx 2.0:

-45 files

-2,421 lines of code



[github.com/jfrostburke/snex2](https://github.com/jfrostburke/snex2)



# Future: Assessment of TOM Toolkit/SNEx 2.0

## Pros:

- Can start a new project extremely quickly
- Documentation (TOM + Django) helps a lot
- 

## Cons:

- Tough to interface with existing databases
- Missing some key features (but still in development)
  - No support for cadenced observations (yet)
  - Can't plot spectral lines to ID supernovae (yet)

### Old SNEx:

- 2,256 files
- 868,130 lines of code

### SNEx 2.0:

- 45 files
- 2,421 lines of code



Interested in making your own TOM?  
David is leading a hack session **Thursday, 2:00 PM.**

# Thanks! Questions?

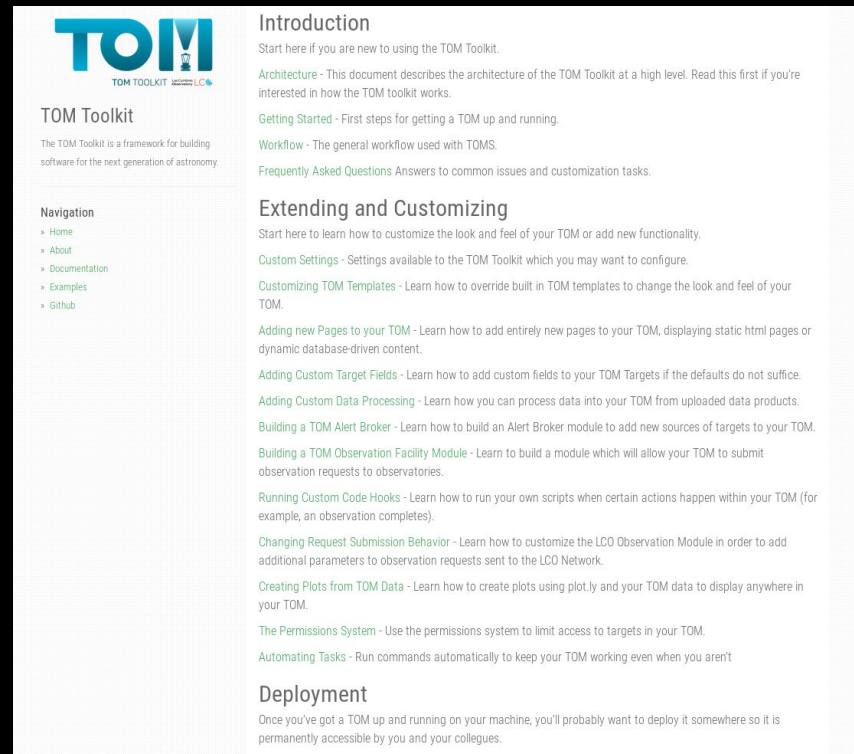
Contact: [jburke@lco.global](mailto:jburke@lco.global)





# Present: Development Journey

- Start with TOM documentation
- Weekly meetings with developers (David + Austin)
- Resources
  - TOM documentation is extremely useful and time-saving
  - Should've read the Django tutorial better before starting
  - Googlable problems



The screenshot shows the TOM Toolkit documentation website. The header features the TOM Toolkit logo and the text "The TOM Toolkit is a framework for building software for the next generation of astronomy". A navigation menu on the left lists: Home, About, Documentation, Examples, and Github. The main content area is titled "Introduction" and includes sections for "Architecture", "Getting Started", "Workflow", and "Frequently Asked Questions". Below this is the "Extending and Customizing" section, which contains numerous links to guides such as "Custom Settings", "Customizing TOM Templates", "Adding new Pages to your TOM", "Adding Custom Target Fields", "Adding Custom Data Processing", "Building a TOM Alert Broker", "Building a TOM Observation Facility Module", "Running Custom Code Hooks", "Changing Request Submission Behavior", "Creating Plots from TOM Data", "The Permissions System", and "Automating Tasks". The "Deployment" section is also visible at the bottom.

## TOM Toolkit

The TOM Toolkit is a framework for building software for the next generation of astronomy

### Navigation

- Home
- About
- Documentation
- Examples
- Github

## Introduction

Start here if you are new to using the TOM Toolkit.

**Architecture** - This document describes the architecture of the TOM Toolkit at a high level. Read this first if you're interested in how the TOM toolkit works.

**Getting Started** - First steps for getting a TOM up and running.

**Workflow** - The general workflow used with TOMS.

**Frequently Asked Questions** Answers to common issues and customization tasks.

## Extending and Customizing

Start here to learn how to customize the look and feel of your TOM or add new functionality.

**Custom Settings** - Settings available to the TOM Toolkit which you may want to configure.

**Customizing TOM Templates** - Learn how to override built in TOM templates to change the look and feel of your TOM.

**Adding new Pages to your TOM** - Learn how to add entirely new pages to your TOM, displaying static html pages or dynamic database-driven content.

**Adding Custom Target Fields** - Learn how to add custom fields to your TOM Targets if the defaults do not suffice.

**Adding Custom Data Processing** - Learn how you can process data into your TOM from uploaded data products.

**Building a TOM Alert Broker** - Learn how to build an Alert Broker module to add new sources of targets to your TOM.

**Building a TOM Observation Facility Module** - Learn to build a module which will allow your TOM to submit observation requests to observatories.

**Running Custom Code Hooks** - Learn how to run your own scripts when certain actions happen within your TOM (for example, an observation completes).

**Changing Request Submission Behavior** - Learn how to customize the LCO Observation Module in order to add additional parameters to observation requests sent to the LCO Network.

**Creating Plots from TOM Data** - Learn how to create plots using plot.ly and your TOM data to display anywhere in your TOM.

**The Permissions System** - Use the permissions system to limit access to targets in your TOM.

**Automating Tasks** - Run commands automatically to keep your TOM working even when you aren't

## Deployment

Once you've got a TOM up and running on your machine, you'll probably want to deploy it somewhere so it is permanently accessible by you and your colleagues.

# Present: Customization (TNS Targets)

- Asked for Austin's help to add new page

File modified:	Purpose:
<code>custom_code/urls.py</code>	Specify URL for new page
<code>snex2/urls.py</code>	Include custom urls
<code>base.html</code>	Add page tab to main navbar
<code>custom_code/views.py</code>	Filler text for new page

# Present: Customization (TNS Targets)

- Asked for the internet's help to get page to do what I want

File modified:	Purpose:
<code>models.py</code>	Generate table in database to store TNS Targets
<code>views.py</code>	Mostly helper functions to output table entries cleanly
<code>filters.py</code>	Specify what and how to filter, also format the form
<code>templates/tns_targets.html</code>	Django magic python/html to fill in form and table
<code>+sqlalchemy ingestion cronjob</code>	Ingest targets into database

# Present: Customization (TNS Targets)

Github: <https://github.com/jfrostburke/snex2>

filters.py:

```
class TNSTargetForm(forms.Form):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.helper = FormHelper()
        self.helper.add_input(Submit('submit', 'Filter'))
        self.helper.layout = Layout(
            Div(
                Div(PrependedText('name', 'Name like'), css_class='col-md-4'),
                Div(PrependedText('source_group', 'Discovered by'), css_class='col-md-4'),
                Div('in_tess', css_class='col-md-4'),
                css_class='form-row'
            ),
            Div(
                Div(PrependedText('disc_mag', 'Discovery mag brighter than',
                                placeholder='19'), css_class='col-md-6'),
                Div(PrependedAppendedText('lnd_jd', 'Last non-detection within the last',
                                'days', placeholder='5'), css_class='col-md-6'),
                css_class='form-row'
            ),
        )
```

tns\_targets.html:

```
<form action="" method="get" class="form">
  <h4 style="margin-top: 0">Filter</h4>
  {% crispy filter.form %}
</form>
```

Webpage output:

Filter

Name like  Discovered by  In TESS?

Discovery mag brighter than 19  Last non-detection within the last 5  days

LCO spectroscopy limit: 18.5