GOTO-XBs

Felipe Jiménez-Ibarra

Sep 23, 2023

CONTENTS

GOTO-XBs is a web platform for follow-up of X-ray binaries (XBs) within GOTO data. It provides access to light curves for all known X-ray binaries, constructed from all the observed epochs and across the 4 GOTO bands. The object database can be updated with newly discovered X-ray binaries, triggering the automatic generation of historical light curves for these objects.

The platform also features a range of analysis tools for working with light curves in real-time. These tools include data filtering, phase folding, smoothing within user-defined time windows, periodogram analysis, plot generation, and more.

Users can also download the data, both raw or previously filtered or smoothed light curves.

The platform is built on a Django framework. So far it consists of two applications:

- **dashboard**: a list of the sources being followed. They are organized in a table that makes it easy to find them.
- **source_viewer**: produces an individual webpage for each object with its information. It use `plotly` for creating an interactive plot of the object light curve in GOTO and it allows to download the data.

CHAPTER

ONE

CONTENTS

1.1 Requirements

GOTO-XBs is built on the Django framework, and the requirements necessary to use it are:

- **python 3.8.x**
- **conda** or **virtualenv + virtualenvwrapper**

Check or in case you need it. More details on the dependencies needed and how to install them in the Quick Start Guide (*GOTO-XBs Environment*).

Ready!? Continue to the *Quick Start Guide*!

1.2 Quick Start Guide

1.2.1 Downloading GOTO-XBs Repository

The source code for GOTO-XBS is hosted on the GOTO GitHub page. To developing GOTO-XBs locally, you can clone the repository in : https://github.com/GOTO-OBS/goto-xbs.git

1.2.2 GOTO-XBs Environment

GOTO-XBs requires a specific environment with the necessary dependencies to run smoothly. We maintain two options for creating this environment: using Conda or virtualenv

Using Conda

1. **Install Conda**:

   If you haven't already, download and install Conda. We suggest to use Miniconda, a minimal installer for Conda, from the official Miniconda website.

2. **Create a Conda Environment**:

   Open your terminal and create a new Conda environment named 'goto-xbs' using `goto-xbs-env.yml` file (provided in the root directory):

   ```
   conda env create -f goto-xbs-env.yml
   ```

3. **Activate the Environment**:

   To activate the newly created environment:

   ```
   conda activate goto-xbs
   ```

and you should now be in the 'goto-xbs' environment.

Using virtualenv

1. **Install Python**:

   You will need Python 3.8.x installed on your system (check official Python website. in case you need it)

2. **Create a virtualenv**:

   Open your terminal and navigate to the root directory of your GOTO-XBs project. Create a virtual environment named 'goto-xbs' using the following command:

   ```
   python -m venv goto-xbs
   ```

3. **Activate the Environment**:

   ```
   source goto-xbs/bin/activate
   ```

   You should now be in the 'goto-xbs' virtual environment.

3. **Installing Dependencies**:

You can now install the project's dependencies using the `requirements.txt` file:

```
pip install -r requirements.txt
```

Your environment is now set up with all the necessary dependencies for GOTO-XBs. You can proceed to run and use the platform.

1.2.3 Settings Files

Settings are organized into three separate files: `dev_settings.py`, `production_settings.py`, and `base_settings.py`.

- `dev_settings.py`: is the app settings used when running in a development environment.
- `production_settings.py`: is employed when deploying the system in a production environment.
- `base_settings.py`: contains settings common to both the development and production environments. It is loaded at the start of both *dev_settings.py* and *production_settings.py*, reducing redundancy and ensuring consistency.

To switch between development and production settings, you need to define the `DJANGO_SETTINGS_MODULE` system variable.

To activate development settings, use the following command in your terminal:

```
$ export DJANGO_SETTINGS_MODULE=goto_xbs.dev_settings
```

To activate production settings, use this command:

```
$ export DJANGO_SETTINGS_MODULE=goto_xbs.production_settings
```

You can verify the current state of the `DJANGO_SETTINGS_MODULE` variable echoing in the terminal by:

```
$ echo $DJANGO_SETTINGS_MODULE
```

This will display the currently active settings module.

1.3 Databases Description

GOTO-XBs utilizes an SQLite database with two primary tables:

1.3.1 Source Table

The *Source* table contains fundamental information about the X-ray binaries (XBs) being tracked. It is structured with the following columns:

- `source_id`: a unique identifier for each source.
- `name`: the name of the source.
- `pretty_name`: a human-readable name for the source.
- `ra`: Right Ascension (RA) coordinate of the source.
- `dec`: Declination (Dec) coordinate of the source.

1.3.2 Photometry Table

The *Photometry* table stores photometric data obtained from the GOTO database. It includes the following columns:

- `source`: a unique identifier linking the data to a specific source. This field serves as a foreign key referencing the *source_id* in the *Source* table.
- `date`: date of the observation.
- `mjd`: Modified Julian Date (MJD) of the observation.
- `mag`: magnitude.
- `mag_err`: error associated with the magnitude.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search