Visualization in IRSA Services using Firefly

Emmanuel Joliet, and Xiuqin Wu[1]

[1]*IPAC/Caltech, Pasadena, CA, USA;* ejoliet@ipac.caltech.edu , xiuqin@ipac.caltech.edu

Abstract.

NASA/IPAC Infrared Science Archive (IRSA) curates the science products of NASA's infrared and submillimeter missions, including many large-area and all-sky surveys. IRSA offers access to digital archives through powerful query engines (including VO-compliant interfaces) and offers unique data analysis and visualization tools. IRSA exploits a re-useable architecture to deploy cost-effective archives, including 2MASS, Spitzer, WISE, Planck, and a large number of highly-used contributed data products from a diverse set of astrophysics projects.

Firefly is IPAC's Advanced Astronomy WEB UI Framework. It was open sourced in 2015, hosted at GitHub. Firefly is designed for building a web-based front end to access science archives with advanced data visualization capabilities.The visualization provide user with an integrated experience with brushing and linking capabilities among images, catalogs, and plots. Firefly has been used in many IPAC IRSA applications, in LSST Science Platform Portal, and in NED's newly released interface.

In this focus demo, we will show case many data access interfaces and services provided by IRSA based on Firefly. It will demonstrate the reusability of Firefly in query, data display, and its visualization capabilities, including the newly released features of HiPS images display, MOC overlay, and the interactions between all those visualization components.

1. Introduction

Archives are data driven end points which care mostly of curating, maintaning and making avaliable reliably and as much as possible without interuption science data. Nowdays, network and internet is the main access and so UI and GUIs together with friendly APIs are the go-to tools for users

IRSA hosts more than 1PB of data from over 15 projects. It enable data extraction, exploration and visualization which required software development and maintenance (as well as hardware) for long-term persistence and availability. User experience (UX) learning curve is faster when User interfaces (UI) are consistent across websites. Best Software engineering (Scrum) rules and practices are put in place for such endeavor and growing challenges. Backend API are best companions to UI and are build behind for easy direct or internal access (i.e. VO protocols).

A simple view of archive access from internet clients is displayed below.

2. Challenges

Archive data usually can be retrieve and displayed as images, charts and tables. Interactivity and interconnectivity is a must in order to allow useful exploration and extraction. Always pursuing a "user friendly" access is key for enabling data exploration accross different projects. Increasing data volume and complicated use cases are challenges that could be overcome by relying on services running closer to the data, with reusable and derived components across different projects and datasets.

Many projects in IPAC either use Firefly in application or use Firefly API for displays. They are WISE [1], Herschel [2], Finder Chart [3], IRSA Catalog Search [4], IRSA Viewer [5], NED [6].

Using Firefly as a base provided a familiar look and feel to users, and allow all the projects to share the new features added to Firefly, like the most recent new feature of HiPS images display and MOC overlay.

Figure 1 "Firefly, IRSA catalog search" give two images. They are Firefly displaying a catalog entries on an image, color-color plot, and histogram, IRSA catalog search result display using Firefly API.

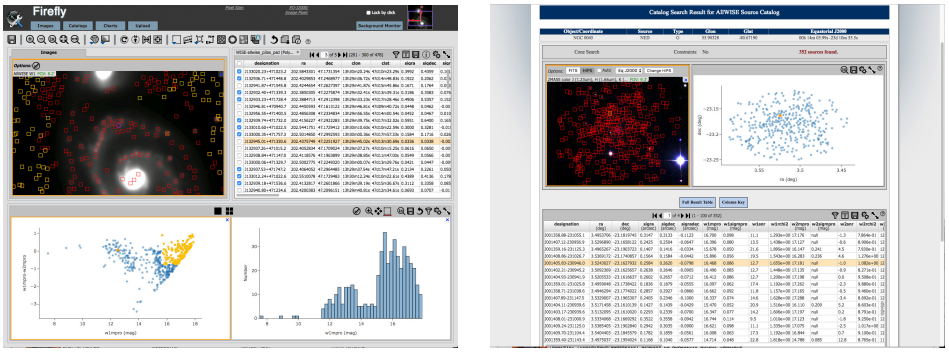

Figure 1. Firefly, IRSA catalog search

Figure 2 "WISE image service, Firefly" give two images. They are WISE image service to display multiple images for WISE, and 4 HiPS images.

3. Technical information

The IPAC Firefly is an open-source library to build UI core components based on ReactJS/Java Through collaborative develompent across IRSA, LSST and NED project, the

[1] `https://irsa.ipac.caltech.edu/applications/wise/`

[2] `https://irsa.ipac.caltech.edu/applications/Herschel/`

[3] `https://irsa.ipac.caltech.edu/applications/finderchart/`

[4] `https://irsa.ipac.caltech.edu/applications/Gator/`

[5] `https://irsa.ipac.caltech.edu/irsaviewer/`

[6] `http://ned.ipac.caltech.edu`

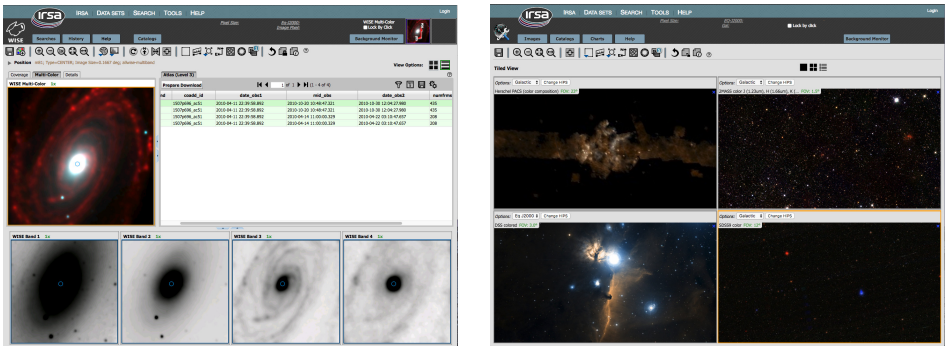

Figure 2. WISE image service, HiPS images

IPAC contributors are maintaining and adding features to the code. The code is hosted in a github repository[7].

The library is used to build Web-based applications. It consists of a Server/client UI architecture sharing common library/stack (apache + tomcat), running HTML/JS client from ES6+(npm+redux+sagas+plotly+,etc.) and a Java layer on the server side, staging searches from DB/APIs/VO. We use Gradle/Jenkins for building and testing follwoing the github pull request workflow with the help of staging builds using docker and kubernetes cluser.

The main widgets available are a FITS image viewer, tables and charts components. The main features are related to explore data using brushing and linking operations.

There are two kind of applications, on one hand there are science data tools such as Time Series tool[8] for light-curve datasets and Finder Chart for cross-comparison of images from various surveys (+API). On the other hand there are project specific applications: i.e. WISE, Spitzer, Planck, Herschel, contributed products.

Recently we have added HIPs capabilities, a periodogram viewer, and more instrument footprints to be overlayed.

The library can be used in two ways. The components and their (re)actions are exposed via a higher level javascript API so HTML pages can import the library and make use of it directly. The Framework can be used to build and extend the existing React classes (low-level) via framework composition. The exposed properties are needed to control project specific requirements. Some of the widgets have particular options to be enabled or disabled depending on the project appearance context/decision. The objects inheritance help developers to maintain and add new features to the Firefly core.

4. Near future

Updates coming soon will include new available image datasets for searching IRSA archives, footprint overlay improved, MOC outline maps.

---

[7] `https://github.com/Caltech-IPAC/firefly`

[8] `https://irsa.ipac.caltech.edu/irsaviewer/timeseries`

We will be improving existing integration with other languages to enable science platform access to run code closer to data for mining and cross exploration with big-data. Python integration within notebooks/JupyterLab exists already and same UI widgets are exposed to allow multiple integration[9]. We will continue the effort to adopt modern web technology to enable richer features and take advantage of 3rd party libraries running in browsers.

5. Demo reference

A demo with step by step tutorial has been put togther in github[10]. The tutorial should cover the following concepts:

- Brushing with histogram/charts: IV catalog search, with error bars, column expression (i.e. WISE)
- Gator -> Light-curve / Period finder with periodogram (WISE/PTF)
- HIPs demo , with URL (+ ivo://), ex: https://irsa.ipac.caltech.edu/data/hips/list
- Footprint overlay (JWST) - layers control
- API html integration: Atlas, Herschel or NED
- IRSA Finder Chart application
- Python integration
- Glimpse of new development: MOC, new Footprint tool

Acknowledgments. The Firefly software is based upon work supported in part by the National Science Foundation through Cooperative Support Agreement (CSA) Award No. AST-1227061 under Governing Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DEAC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.

The applications mentioned were supported by IRSA, the NASA/IPAC Infrared Science Archive. IRSA curates the science products of NASA's infrared and submillimeter missions, including many large-area and all-sky surveys, and NED the NASA/IPAC Extragalactic Database.

[9]see Firefly Python client: `https://github.com/Caltech-IPAC/firefly_client/blob/master/doc/index.rst`

[10]`https://github.com/ejoliet/adass2018`