Meetup2xibo Documentation

Release 3.3.1

Joel Shprentz

Jun 09, 2021

Introduction

Meetup2xibo is a command line program that retrieves event descriptions from Meetup.com and loads them into a Xibo digital signage dataset. Xibo can display events in sign layouts such as agendas, calendars, room activity signs, and maps.

Meetup2xibo was developed for Nova Labs, a makerspace in Reston, Virginia. The busy Nova Labs calendar has up to a dozen Meetup.com events per day. Xibo displays event agendas for today and tommorrow on several large wall-mounted monitors. Meetup2xibo keeps those agendas up-to-date as events are added, cancelled, or updated.

Overview

Meetup2xibo is a command line program that retrieves event descriptions from Meetup.com and loads them into a Xibo digital signage dataset. Xibo can display events in sign layouts such as agendas, calendars, room activity signs, and maps.

Meetup2xibo was developed for Nova Labs, a makerspace in Reston, Virginia. The busy Nova Labs calendar has up to a dozen Meetup.com events per day. Xibo displays event agendas for today and tommorrow on several large wall-mounted monitors. Meetup2xibo keeps those agendas up-to-date as events are added, cancelled, or updated.

1.1 Resources

- Documentation: https://meetup2xibo.readthedocs.io
- Source code: https://github.com/jshprentz/meetup2xibo
- Python package: https://pypi.org/project/meetup2xibo/
- Free software: MIT license

1.2 Features

Meetup2xibo . . .

- Retrieves events from the Meetup.com API.
- Inserts, updates, and deletes events via the Xibo CMS API.
- Extracts event locations from Meetup.com venue name and find-us fields.
- Maps abbreviated and misspelled event locations to preferred forms.
- Allows per-event overrides of location mapping rules.
- Computes event end times from Meetup.com start times and durations.
- Formats event start and end times for Xibo.
- Removes accounting codes from event names.
- Detects and logs schedule conflicts.
- Logs changes to Xibo events.
- Logs location mappings.

Supporting programs . . .

- Summarize logs in a daily email message.
- Report yesterday's date for use in shell scripts.

1.3 Credits

Meetup2xibo was developed by Joel Shprentz (@jshprentz).

Background

Nova Labs announces events and registers participants with Meetup.com. Nova Labs display event schedules and other information with Xibo. Meetup2xibo replaces a manual process for downloading Meetup.com event information into Xibo.

2.1 The Players

Nova Labs is a makerspace in Reston, Virginia with over 300 members (as of April 2019). Nova Labs members can access a woodshop, a metal shop, laser cutters, 3D printers, an electronics bench, sewing and embroidery machines, vinyl cutters, a blacksmithing forge, and many other maker tools. To build communities of makers, Nova Labs offers a busy schedule of classes, meetings, workshops, and other events to its members and guests.

Meetup.com is a web-based service that helps organize physical events for people with similar interests. Meetup.com maintains a database of events, provides event web pages, and handles attendee registration details. Equally important, Meetup.com suggests other related events to participants, helping event organizers to reach potential new attendees.

Nova Labs maintains its full calendar of events on Meetup.com. Figure 2.1 shows the Meetup.com calendar for one week of Nova Labs events.

Xibo is an open source, web based, digital signage system. The Xibo CMS (Content Management System) allows administrators to design sign layouts and schedule them on sign displays. The Xibo CMS is written in PHP and stores data in a MySQL database. Xibo client software runs on Windows- or Android-based compters connected to display monitors.

Nova Labs installed Xibo in August 2017 to control sign display monitors at the entrance door, the front desk, and elsewhere. Nova Labs has created dozens of sign layouts including a daily agenda (Figure 2.2), event announcements, and photos of past activities.

2.2 The Manual Process

For 18 months, Nova Labs administrators copied event data from Meetup.com to Xibo using a manual process.
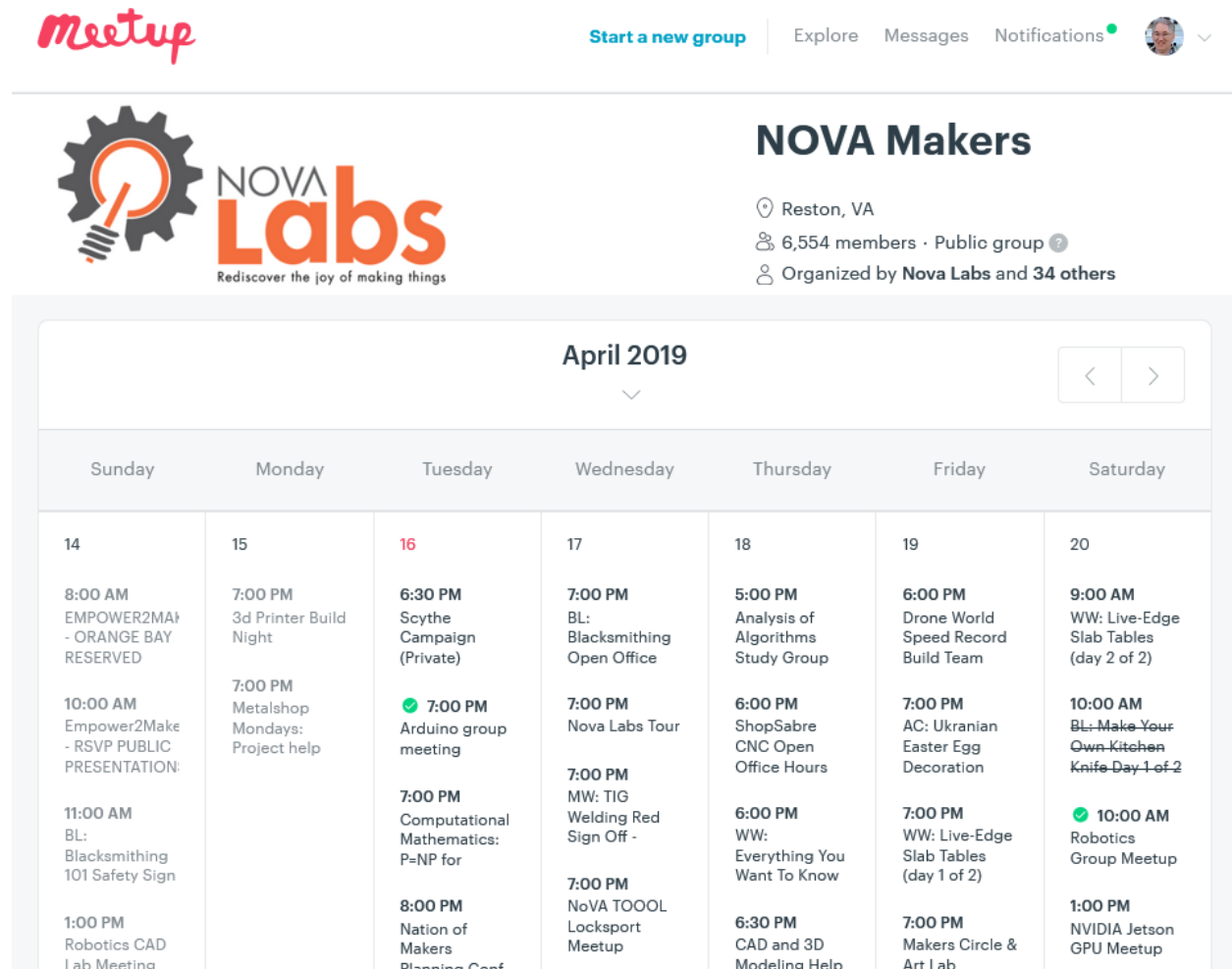
Figure 2.1.: An example of a Meetup.com calendar of events at Nova Labs.
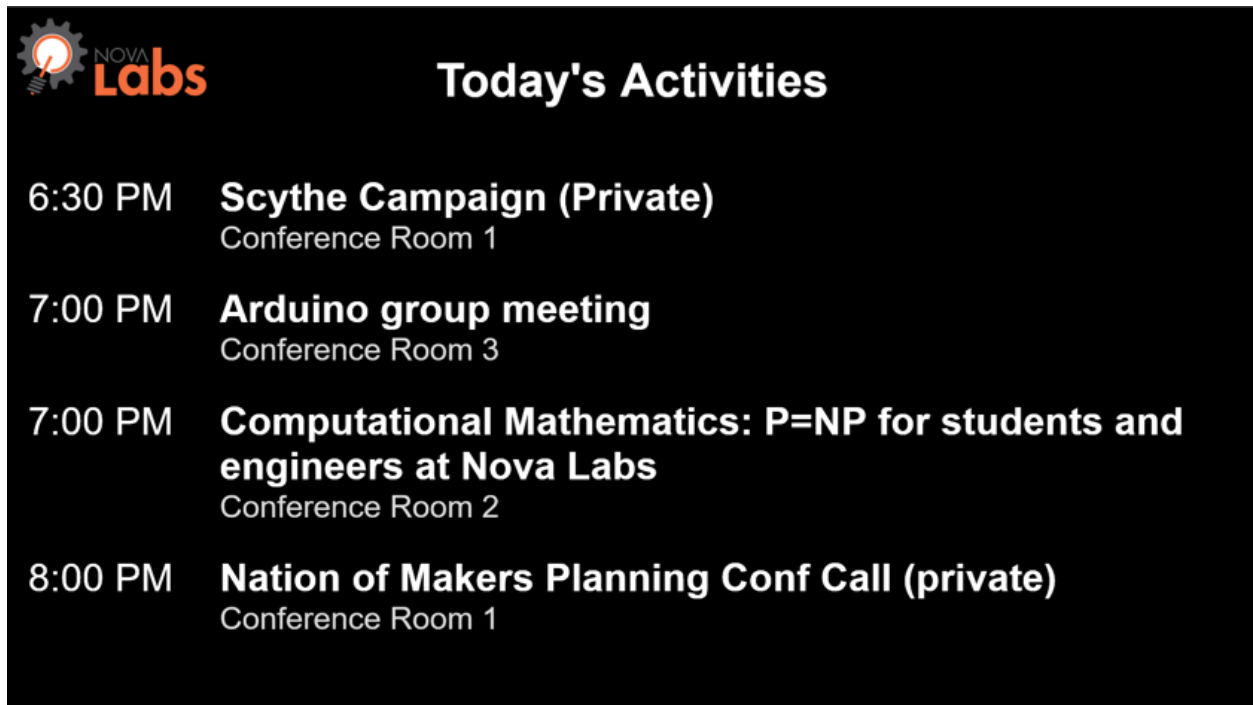
Figure 2.2.: Xibo displays a daily agenda of Nova Labs events.

Once or twice a week, a Xibo administrator used a web browser to request event data from Meetup.com's API (application programming interface). Meetup.com provided for 200 current and upcoming events. The JSON (JavaScript Object Notation) formatted data was saved to a file. Figure 2.3 shows an example of the JSON data.

Figure 2.3.: A small part of the JSON event data from Meetup.com. Left: Raw JSON data from Meetup.com. Right: Pretty printed JSON data showing fields and values.

The Xibo administrator then ran a 70-line Python script to extract and convert pertinent information from the JSON file. The script saved the extracted information in a CSV (comma separated value) formatted file. The administrator could open the file with Microcoft Excel to review the conversion and correct errors. Figure 2.4 shows an example of the converted data.

The Xibo admininstator used a web browser to import the CSV formatted event data into a Xibo CMS dataset, replacing all previous data. Xibo displayed the import dialog box shown in Figure 2.5. The administrator had to check certain boxes and correct the column order before uploading the file.

The Xibo admininstrator completed the process by reviewing the event data in the Xibo dataset. Figure 2.6 shows how Xibo displays event data for review. Xibo computes rightmost three field (start time, days till start, and minutes past end). The administrator can click on any row to edit the data.

start_time	end_time	meetup_id	name	location
11/14/2017 19:00	11/14/2017 21:00	242650704	Laser 103: Exotic Materials (Glass/Mirrors & Ceram	Conference Rm 3
11/15/2017 18:00	11/15/2017 20:00	rjwmvnywpbtb	Blacksmithing for supervised practice and fun	Out back in the parking lot
11/15/2017 18:00	11/15/2017 21:00	244890075	Woodshop Roundup and Community Gathering	Woodshop
11/15/2017 18:30	11/15/2017 21:30	zwdswmywpbcb	Threat Hunting w/ Splunk (Advanced)	Classroom A and B
11/15/2017 19:00	11/15/2017 22:00	znmlrnywpbtb	TechComm Team Meeting	Conference Rm 2
11/15/2017 19:00	11/15/2017 21:00	vtskzmywpbtb	FPGA Study and Experimentation Group	Conference Rm 1
11/16/2017 18:00	11/16/2017 21:00	nbtgxmywpbvb	ShopSabre CNC Office Hours	In the woodshop
11/16/2017 18:00	11/16/2017 20:30	244741422	Startups-Ignite Startup Social	Classroom A and B
11/16/2017 19:00	11/16/2017 21:00	244770816	Laser Cutter 101 Red Safety Training	Conference Rm 3
11/16/2017 19:00	11/16/2017 21:00	qdxwwmywpbvb	Woodworking Office Hours: Projects, Help, and Tc	woodshop
11/17/2017 19:00	11/17/2017 22:00	244754640	Cosplay Modding Workshop	Classroom A
11/17/2017 19:00	11/17/2017 22:00	zfbzmlywpbwb	Makers Circle & Art Lab	Nova Labs
11/17/2017 19:00	11/17/2017 22:00	244873262	Intro to Woodworking - Cutting Boards (Day 1 of 2)	Woodshop
11/18/2017 10:00	11/18/2017 12:00	vxslmmywpbxb	Robotics Group Meetup	Classroom A
11/18/2017 12:30	11/18/2017 15:30	245115754	Metal Shop Orientation - Yellow Tools Safety Trair	Metal shop

Figure 2.4.: The first several lines of event data converted to CSV format and viewed as a spreadsheet in Microsoft Excel.

2.3 Lessons Learned

Nova Labs members were pleased with the Xibo displays of event agendas.

Members were disappointed that the displayed agendas often were out-of-date compared with Meetup.com. The twice-weekly manual process could not keep up with the dozen or more Meetup.com changes entered by event administrators each day. For example, when an event was assigned to a different room shortly before the start time, Meetup.com would show the new room assignment, but the Xibo still displayed the old location.

Xibo administrators had to precisely follow the manual process. Occasional mistakes led to outdated events getting loaded into Xibo.

Xibo administrators had to repeatedly correct some faulty data entered into Meetup.com. Some events included variations on room names. Occasionally an event showed no location at all. The Python script grew to accomodate frequently needed event edits and provide default values.

Eventually the timeliness concerns and the tediousness of the manual process motivated the development of an automated solution, meetup2xibo.

Figure 2.5.: The Xibo "CSV Import" dialog box ready to upload a file.

ID	Name	Location	Meetup ID	ISO Start Time	ISO End Time	Start Time	Days Till Start	Minutes Past End	
1	Laser 103: Exotic Materials (Glass/Mirrors & Ceramic/Stones)	Conference Rm 3	242650704	2017-11-14 19:00:00	2017-11-14 21:00:00	7:00 PM	-530	762849	✖
2	Blacksmithing for supervised practice and fun	Out back in the parking lot	rjwmvnywpbtb	2017-11-15 18:00:00	2017-11-15 20:00:00	6:00 PM	-529	761469	✖
3	Woodshop Roundup and Community Gathering	Woodshop	244890075	2017-11-15 18:00:00	2017-11-15 21:00:00	6:00 PM	-529	761409	✖
4	Threat Hunting w/ Splunk (Advanced)	Classroom A and B	zwdswmywpbcb	2017-11-15 18:30:00	2017-11-15 21:30:00	6:30 PM	-529	761379	✖
5	TechComm Team Meeting	Conference Rm 2	znmlrnywpbtb	2017-11-15 19:00:00	2017-11-15 22:00:00	7:00 PM	-529	761349	✖
6	FPGA Study and Experimentation Group	Conference Rm 1	vtskzmywpbtb	2017-11-15 19:00:00	2017-11-15 21:00:00	7:00 PM	-529	761409	✖

Figure 2.6.: The Xibo dataset review screen lets administrators reivew and edit event data.

Overview
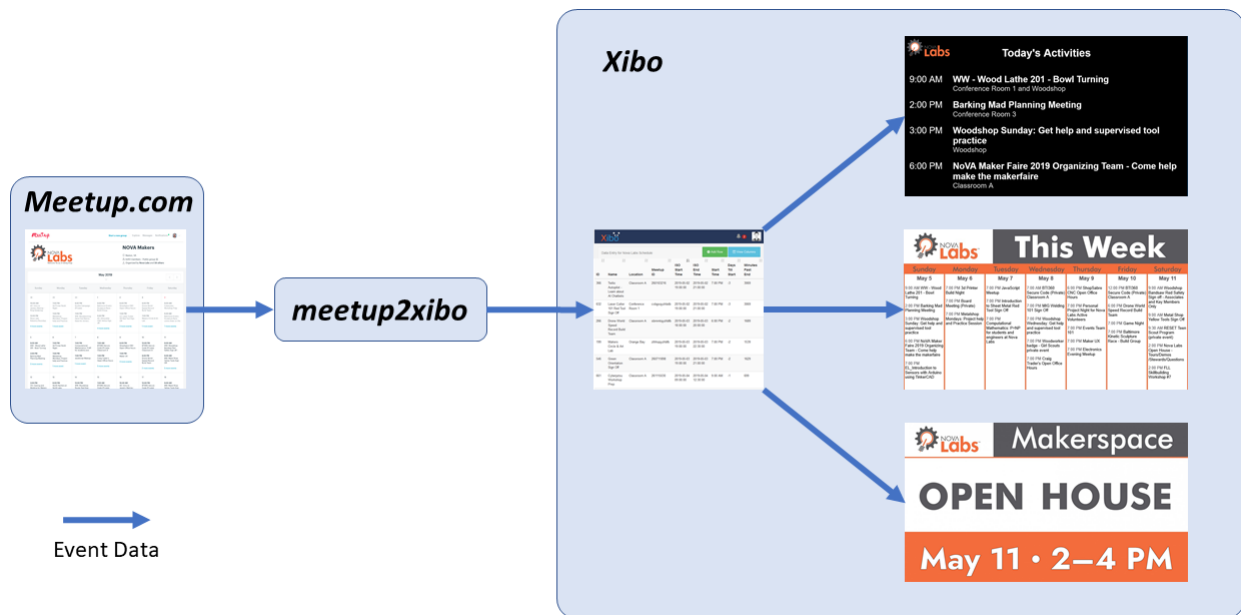
Figure 3.1.: Meetup2xibo retrieves event data from Meetup.com, transforms the data, and updates the event data stored in a Xibo dataset. Ticker widgets in Xibo layouts display the event data retrieved from the Xibo dataset.

Meetup2xibo is a command line program that retrieves event descriptions from Meetup.com and loads them into a Xibo digital signage dataset. Xibo layouts can contain ticker widgets that filter, sort, and display event data. Figure 3.1 shows event data flow from Meetup.com to Xibo layouts.

3.1 Preparation

Meetup2xibo run on any computer running Linux or a similar operating system. An administrator will need some basic computer skills to install and configure meetup2xibo. See *Prerequisites* for a description of the computing and skills requirements.

Meetup2xibo stores event data in a Xibo dataset. See *Xibo Quick Start* for instructions about about importing the dataset structure and an agenda layout. Alternatively, see *Xibo Events Dataset* for instructions about creating the dataset manually.

Meetup2xibo accesses the Xibo API as a distinct authorized user. See *Xibo Client User* for instructions about adding and authorizing this user.

3.2 Installation

Meetup2xibo can be installed with standard Python package tools. See *Installation*.

3.3 Configuration

Meetup2xibo gets its configuration from environment variables. The configuration includes:

- Meetup.com API settings
- Xibo CMS API credentials
- Xibo dataset column names
- Location corrections
- Places for schedule conflict analysis
- Date/time thresholds for event insertion, deletion, and cancellation

The *meetup2xibo man page* defines the configuration environment variables. *Configuration* explains the environment variables in more detail. An annotated example configuration script, /data/meetup2xibo.env, provides a starting point for customization.

3.4 Logging

Meetup2xibo can log its activity to a named file, standard output, or both. *Command line options* control the message levels to log and where to send the logs.

A supporting program, **summarize-m2x-logs**, analyzes logs and summarizes changes to events. See the *man page*.

3.5 Cron Jobs

The Linux cron daemon (or the Windows Task Scheduler) can run Cron runs meetup2xibo every 10 minutes so the Xibo layouts show the latest event changes, particularly last minute room changes. Cron runs summarize-m2x-logs early every morning to email a summary of the previous days logs. See *Cron Jobs* for instructions about setting up the cron jobs.

Prerequisites

4.1 Computing Environment

- Linux or maybe Windows

- Internet

- Network to Xibo CMS via HTTPS

- Python 3.5, 3.6, or 3.7

- Python virtual environment recommended

Meetup2xibo was developed, tested, and run on computers with Ubuntu, a Linux operating system. Meetup2xibo has no dependencies on Ubuntu or Linux; it should run on Windows, macOS, Raspberry Pi, or anywhere Python is available.

Meetup2xibo requires an internet connection to access Meetup.com.

Meetup2xibo requires network connection to your Xibo CMS. If the Xibo CMS runs on a private network, meetup2xibo must also run on that network.

Meetup2xibo connects securely to the Xibo CMS via HTTPS. The Xibo CMS must be configured for secure HTTPS access as described by "Xibo for Docker on Linux."

Meetup2xibo was coded with the Python programming language. Python must be installed to run meetup2xibo. Meetup2xibo has been tested successfully with Python 3.5, 3.6, and 3.7

Meetup2xibo is distributed as a Python package. It has dependencies on several other Python packages. The Python Packaging Authority recommends creating a Python virtual environment to keep meetup2xibo and its dependencies from conflicting with other Python packages. See "Installing packages using pip and virtual environments."

4.2 Skills Needed

- Run commands from Linux shell

- Schedule cron jobs

- Edit configuration files

- Access Xibo CMS as administrator

You must be able to run commands on your computer. On Linux, you can use a terminal window to enter commands. On Windows, you can use a command or PowerShell window. From a remote computer, you can connect via ssh to run commands.

You must be able to schedule commands on your computer to run meetup2xibo periodically. On Linux, setup cron jobs. On Windows, use Windows Task Scheduler.

You must be able to use a text editor to edit configuration files.

You must have administrative access to your Xibo CMS to create dataset, layouts, schedules, and users.

Xibo Quick Start

Meetup2xibo stores event data in a Xibo dataset. The steps below describe a quick alternative to manually creating the dataset as described in *Xibo Events Dataset*.

5.1 Download Agenda Layout and Dataset Structure

Xibo can import or export dataset structures only when associated with a layout. Download the agenda layout and event dataset structure, which was exported from Xibo as a zip file: /data/export_quick-start-agenda-for-xibo-1-8.zip.

5.2 Import Agenda Layout

Import the downloaded agenda layout and associated event dataset structure. Click the *Import* button above the layouts list, as shown in Figure 5.1.

Select and upload the agenda layout zip file as shown in Figure 5.2.

Observe that the layout list contains the imported layout as shown in Figure 5.3.

5.3 Review and Rename Imported Dataset

Observe that the dataset list contains the imported dataset as shown in Figure 5.4.

Edit the dataset's name, description, and code, as shown in Figure 5.5. Choose any meaningful dataset name. The description is optional, but helpful. Choose any meaningful single-word code to identify the dataset for Xibo API clients. The meetup2xibo configuration must contain this code as described in *Xibo Dataset Code and Column Names*.
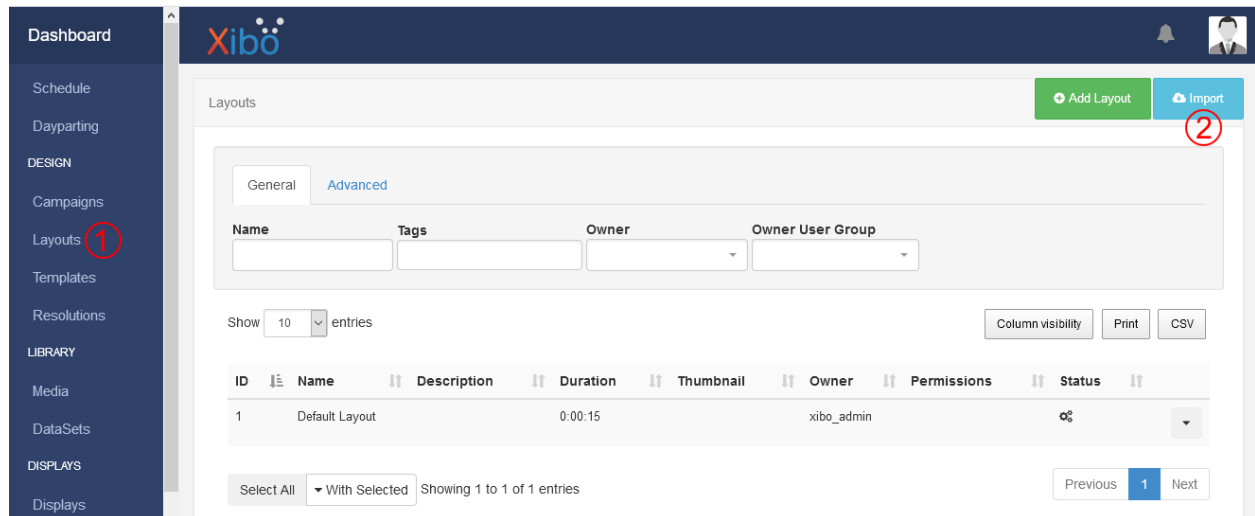
Figure 5.1.: Import layout. Click *Layouts* (1) in the Xibo CMS menu to display the layouts list. Click the *Import* button (2) to open the *Upload Layout* dialog box.

5.4 The Agenda Layout

Figure 5.6 shows the imported quick start agenda layout. Xibo administrators may customize the layout with Xibo's layout design tools.
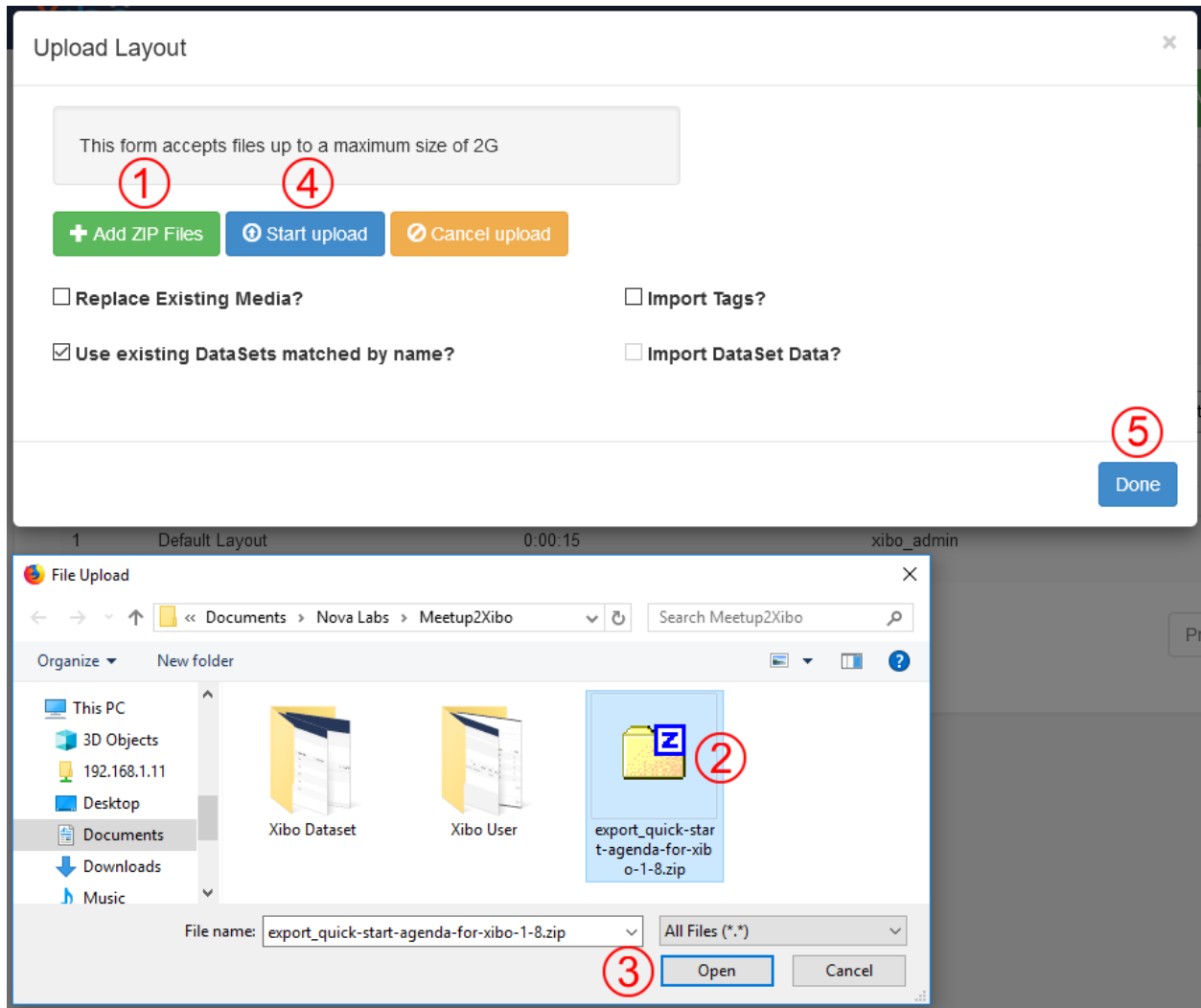
Figure 5.2.: Upload layout. Click the *Add ZIP Files* button (1) to open the *File Upload* dialog box. Navigate to folder containing the downloaded zip file. Select the zip file (2). Click the *Open* button (3) to add the zip file to the *File Upload* dialog box's list of files. Click the *Start upload* button (4) to upload the zip file. When the upload completes, click the *Done* button (5) to close the dialog box.
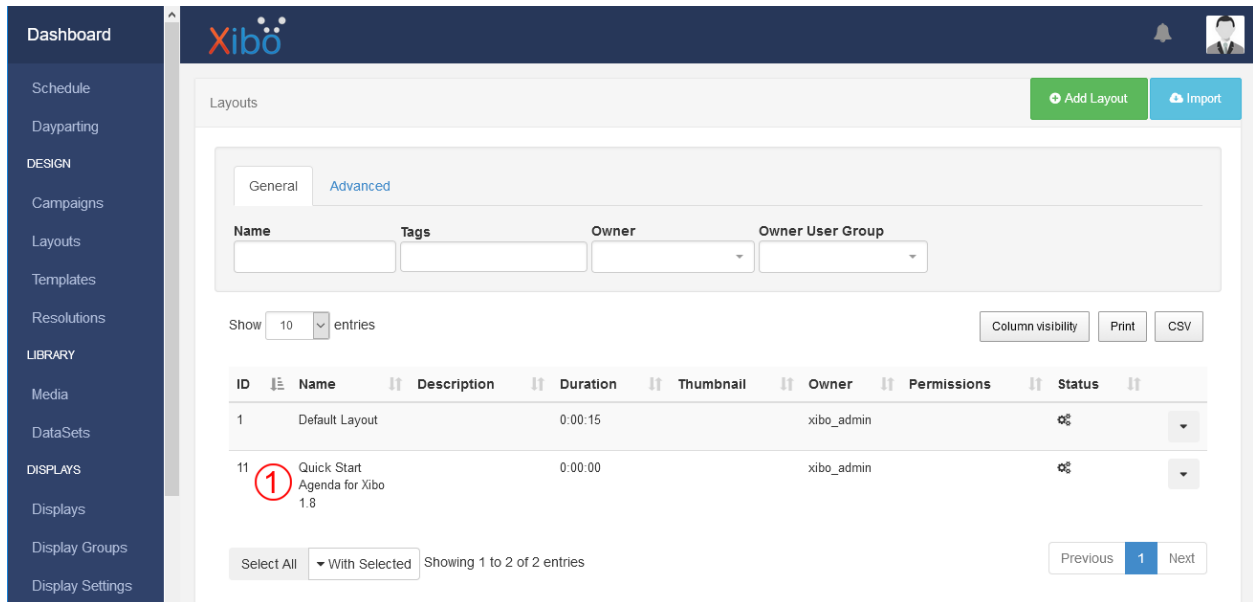
Figure 5.3.: Xibo's layout list includes the imported layout (1).
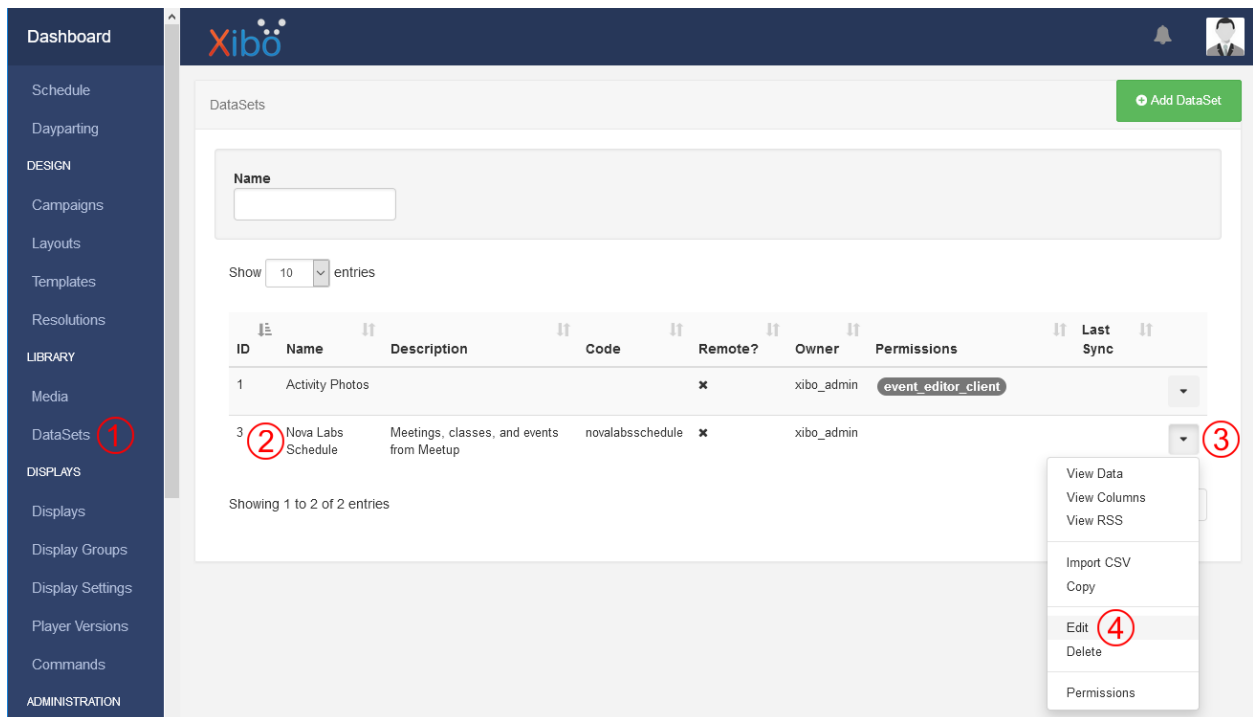
Figure 5.4.: Review dataset. Click *Datasets* (1) in the Xibo CMS menu to display the datasets list. Observe that the list includes the imported dataset (2). In the imported dataset's row, click the down arrow (3) to open the popup menu. Select *Edit* (4) from the menu to open the *Edit Dataset* dialog box.

Figure 5.5.: Edit dataset. Edit the dataset name (1), description (2), and code (3). Click *Save* (4) to save the changes.

Figure 5.6.: The quick start agenda displays a schedule of today's events.

Xibo Events Dataset

Xibo stores event data in a dataset. Meetup2Xibo manages five data value columns in that dataset. Xibo administrators may add additional formula columns to aid filtering, sorting and formating data for layouts.

The steps below describe how to setup the dataset in Xibo. *Xibo Quick Start* provides an alternative method using an imported dataset structure.

6.1 Create an Events Dataset

Meetup2xibo saves, updates, and deletes event data stored in a Xibo dataset. Click the *Add DataSet* button above the datasets list, as shown in Figure 6.1.
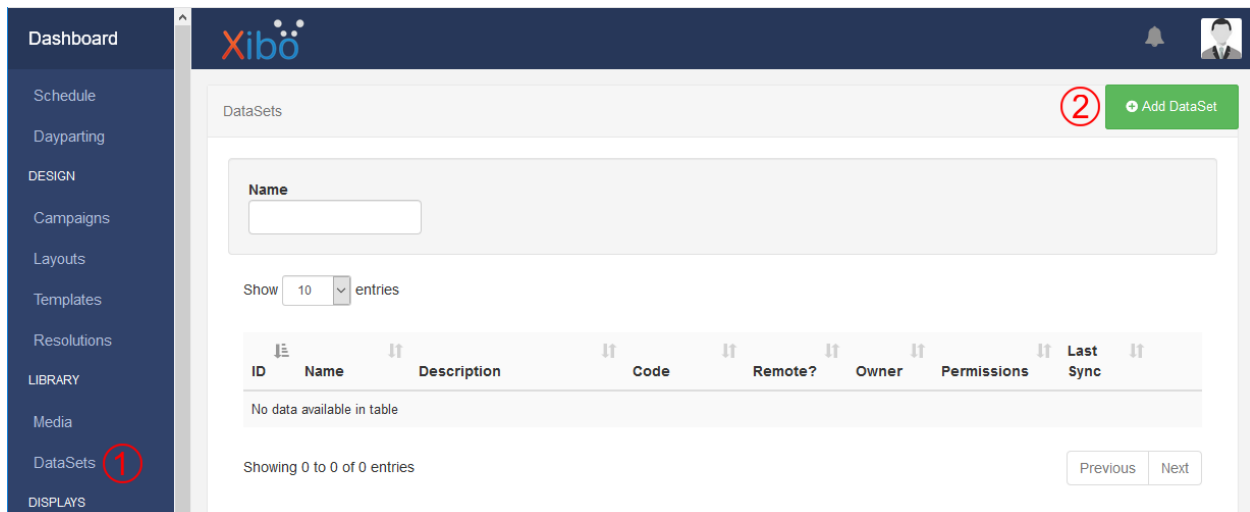
Figure 6.1.: Click *Datasets* (1) in the Xibo CMS menu to display the datasets list. Click the *Add Dataset* button (2) to open the *Add Dataset* dialog box.

Enter the new dataset's name, description, and code, as shown in Figure 6.2. Choose any meaningful dataset name. The description is optional, but helpful. Choose any meaningful single-word code to identify the dataset for Xibo API clients. The meetup2xibo configuration must contain this code as described in *Xibo Dataset Code and Column Names*.


Figure 6.2.: Type the dataset name (1), description (2), and code (3). Click *Save* (4) to add the new dataset.

6.2 Dataset Columns

Meetup2xibo expects the dataset to contain five string value columns. Table 6.1 shows the column headings used at Nova Labs. Xibo administrators may choose any column headings and order.

Table 6.1.: Events Dataset Value Columns

Heading	Example Value	Recommended	
		Filter	Sort
Name	3D Printing 101	✓	
Location	Conference Room 3	✓	
Meetup ID	fkcslpyzhbrb	✓	
ISO Start Time	2019-05-13 18:00:00		✓
ISO End Time	2019-05-13 19:30:00		

Xibo automatically adds one column to new datasets. Select *Edit* from the events dataset's popup menu, as shown in Figure 6.3.

Correct the column heading, as shown in Figure 6.4. All value columns have the *String* data type and the *Value* column type. The list content should be left blank. The default column order will suffice.
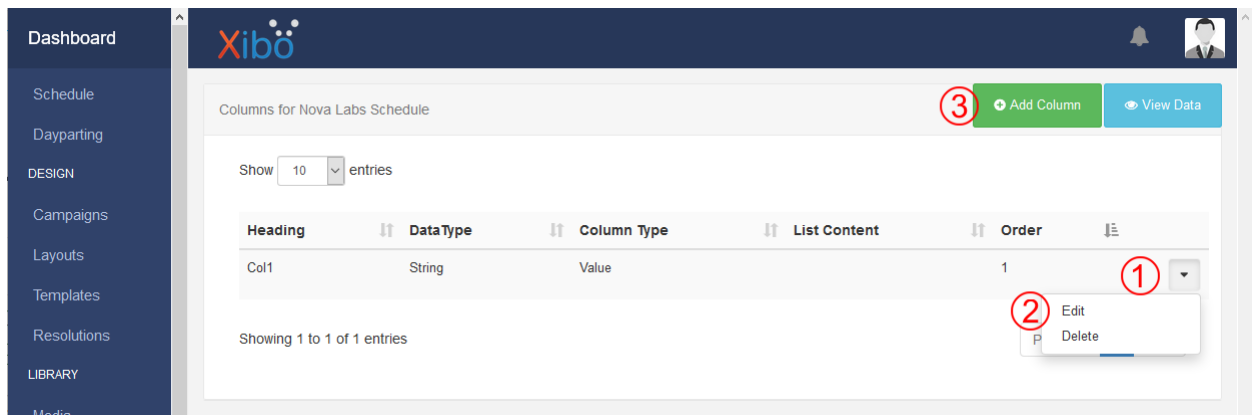
Figure 6.3.: In the Xibo-supplied first row, click the down arrow (1) to open the popup menu. Select *Edit* (2) from the menu to open the *Edit Column* dialog box. Click *Add Column* (3) to add each additional dataset column.

Xibo administrators may choose which columns to filter and sort. Filtering searches for events containing desired values. For example, a Xibo administrator can filter events to show only those with a name containing "Electronics 101." Sorting changes the order of events listed when viewing data. For example, a Xibo administrator can sort events by starting time. Nova Labs filters and sorts the columns shown in Table 6.1.

Click *Add Column* (Figure 6.3) to add each of the remaining columns to the dataset. The *Add Column* form is similar to the *Edit Column* form shown in Figure 6.4. When complete, the events dataset columns should appear in a list similar to Figure 6.5.

6.3 Formula Columns

Xibo administrators may add formula columns for filtering, data formatting, or CSS styling. Table 6.2 shows examples of formula columns used at Nova Labs. The example values were calculated at 6:06 PM for the example event in Table 6.1.

Figure 6.4.: Edit the heading (1) to "Name" or the heading chosen for the first column. Optionally check the boxes for *Filter?* (2) and/or *Sort?* (3). Click the *Save* button (4) to save the changes.
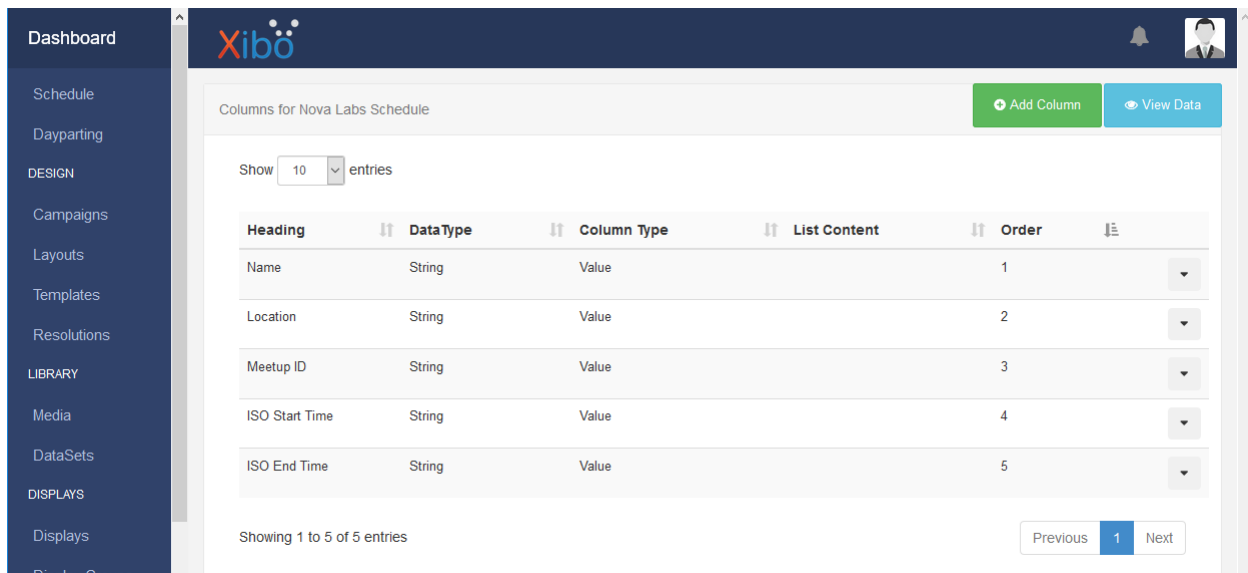
Figure 6.5.: The events dataset columns after the five value columns have been created.

Table 6.2.: Events Dataset Formulas

Heading	Example Value	Purpose	Formula
Cancelled		CSS class	`IF(`Location` = 'Cancelled', 'cancelled', '')`
Compact End Time	7:30 PM	Formatting	`IF(TIME(`ISO End Time`) = '12:00:00.000000', 'Noon', IF(TIME(`ISO End Time`) = '00:00:00.000000', 'Midnight', IF(MINUTE(`ISO Start Time`) = 0 AND MINUTE(`ISO End Time`) = 0, DATE_FORMAT(`ISO End Time`, '%l %p'), DATE_FORMAT(`ISO End Time`, '%l:%i %p'))))`
Compact Start Time	6:00	Formatting	`IF(TIME(`ISO Start Time`) = '12:00:00.000000', 'Noon', IF(TIME(`ISO Start Time`) = '00:00:00.000000', 'Midnight', IF(MINUTE(`ISO Start Time`) = 0 AND MINUTE(`ISO End Time`) = 0, IF((HOUR(`ISO Start Time`) < 12) XOR (HOUR(`ISO End Time`) < 12), DATE_FORMAT(`ISO Start Time`, '%l %p'), DATE_FORMAT(`ISO Start Time`, '%l')), IF((HOUR(`ISO Start Time`) < 12) XOR (HOUR(`ISO End Time`) < 12), DATE_FORMAT(`ISO Start Time`, '%l:%i %p'), DATE_FORMAT(`ISO Start Time`, '%l:%i')))))`
Day Name or Date	Today	Formatting	`IF(DATEDIFF(`ISO Start Time`, NOW()) = 0, 'Today', IF(DATEDIFF(`ISO Start Time`, NOW()) BETWEEN 1 AND 5, DATE_FORMAT(`ISO Start Time`, '%W'), DATE_FORMAT(`ISO Start Time`, '%M %e')))`
Days Till Start	0	Filtering	`DATEDIFF(`ISO Start Time`, NOW())`
Minutes Past End	-84	Filtering	`TIMESTAMPDIFF(MINUTE, `ISO End Time`, NOW())`
Month and Day	May 13	Formatting	`DATE_FORMAT(`ISO Start Time`, '%M %e')`
Past Day		CSS class	`IF(DATEDIFF(`ISO Start Time`, NOW()) < 0, 'pastday', '')`
Start	6:00	For-	`DATE_FORMAT(`ISO Start Time`, "%l:%i %p")`

All columns have the *String* data type and the *Formula* column type. The List Content should be left blank. Xibo administrators may choose any column headings and order.

Click *Add Column* (Figure 6.3) to add each of the formula columns to the dataset. The *Add Column* form is shown in Figure 6.6.

Figure 6.6.: Edit the heading (1) to formula column heading. Select *Formula* (2) from the *Column Type* menu. Enter the formula into the *Formula* text field (3). Click the *Save* button (4) to save the changes.

6.3.1 CSS Class Formulas

The CSS class formulas check a condition and compute either a blank value or a CSS class name. Use these formulas in the source view of a Xibo layout ticker widget appearance tab.

Cancelled The CSS class name *cancelled* when an event is cancelled; blank otherwise. For example, the Agenda layout (Figure 2.2) uses the *cancelled* CSS class to strike through cancelled events names.

```
<div class="event-name [Cancelled|65]">[Name|1]</div>
```

The widget's optional stylesheet contains the CSS class definition.

```
.cancelled {
    text-decoration: line-through;
    }
```

Past Day The CSS class name *pastday* when an event occured before today; blank otherwise. For example, the Nova Labs weekly calendar grid layout uses the *pastday* CSS class to dim events on days before today.

```
<p class="[Past Day|64] [Cancelled|65]"><span class="time">[Start Time|6]</span>␣
↪[Name|1]</p>
```

The widget's optional stylesheet contains the CSS class definition.

```
.pastday {
    color: #B1B0B5;
    }
```

6.3.2 Date and Time Formatting Formulas

The date and time formatting formulas render start and end dates and times in formats appropriate for various layouts. Use these formulas in a Xibo layout ticker widget appearance tab.

Month and Day The spelled out month and start date of an event.

Day Name or Date The word "Today" for today's events. The spelled out day name for events one to five days ahead. The month and day of other events.

Start Time The start time in 12-hour AM/PM format.

Compact Start Time and Compact End Time Used together to render event time ranges such as 2–4 PM.

Shows minutes when either the start or end time do not occur on the hour, such as 1:30–4:00 PM.

Shows the start time AM/PM indicator when the start and end AM/PM time values differ, such as 10 AM–4 PM.

Shows the words "midnight" and "noon" as needed, such as Noon–4 PM.

6.3.3 Filtering Formulas

The filtering formulas compute date and time differences useful for selecting events. Use these formulas in a Xibo layout ticker widget filter tab.

For example, the Agenda layout (Figure 2.2) displays today's events until 30 minutes after their end time. The layout's "Use advanced filter clause?" checkbox is checked and this filter clause selects the desired events:

```
`Days Till Start` = 0 AND `Minutes Past End` < 30
```

Days Till Start The number of days until the event starts. This value is 0 for today's events, 1 for tomorrow's events, and so on. Past events have negative values.

Minutes Past End The number of minutes since the event ended.

Xibo Client User

Setup a separate Xibo user for the client application meetup2xibo. That user will have authorization only to update data in the event dataset. Experienced Xibo administrators can follow this outline:

1. Add a user to Xibo.

2. Authorize that user to access the dataset API.

3. Grant the new user the view and edit permissions for the event dataset.

4. Add a new application to Xibo, noting the application credentials.

5. Set the application owner to the new user.

The step-by-step instructions below demonstrate how to setup the client user, dataset permissions, and event editor application.

7.1 Add a User for Meetup2xibo

A distinct meetup2xibo client user will provide secure, limited access to the Xibo API. Click the *Add User* button above the users list, as shown in Figure 7.1.

Enter the new user's user name and password, as shown in Figure 7.2. Choose any meaningful user name. The password will never be used, so choose a long, random sequence of characters.

7.2 Authorize the User to Access Event Data

The meetup2xibo user requires access to the datasets API, but should be blocked from other API components. Select *Page Security* from the meetup2xibo user's popup menu as shown in Figure 7.3.

Update the ACL (Access Control List) for the meetup2xibo user, as shown in Figure 7.4
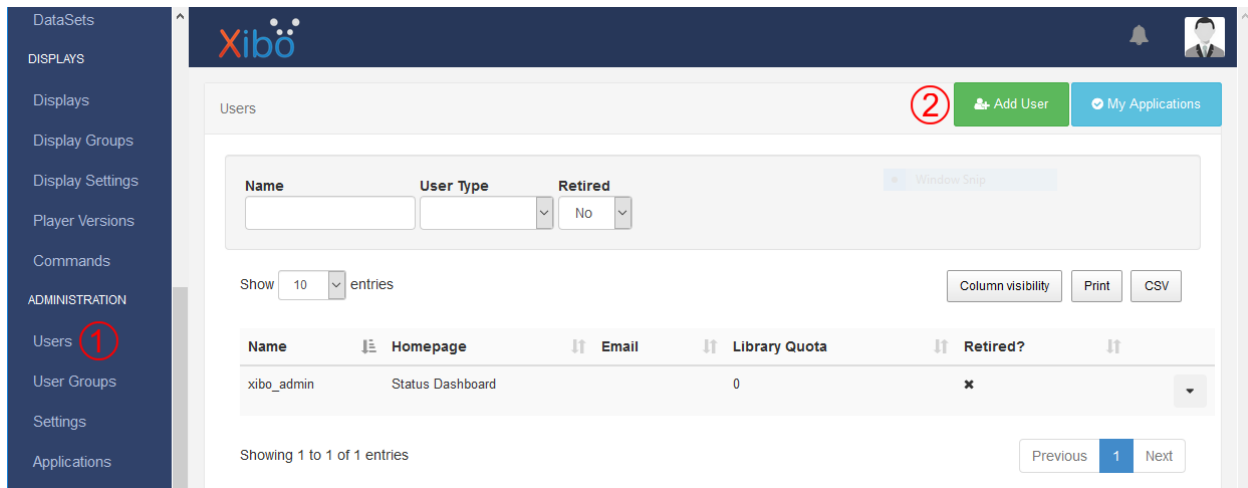
Figure 7.1.: Click *Users* (1) in the Xibo CMS menu to display the users list. Click the *Add User* button (2) to open the *Add User* dialog box.

7.3 Grant Dataset Permissions

The meetup2xibo user requires permission to view and edit event data. Select *Permissions* from the event dataset's popup menu, as shown in Figure 7.5.

Authorize the meetup2xibo user to view and edit the dataset, as shown in Figure 7.6.

7.4 Add an Application for Meetup2xibo

Xibo limits API access to a limited list of client applications. Click the *Add Application* button above the applications list as shown in Figure 7.7.

Enter the new application's name, as shown in Figure 7.8. Choose any meaningful application name.

7.5 Authorize the Application

The new application must allow API access from the unattended application meetup2xibo. Select *Edit* from the meetup2xibo application's popup menu as shown in Figure 7.9.

The *Xibo CMS API credentials configuration* requires the client ID and secret shown in the *Edit Application* dialog box. Authorize the client credentials, as shown in Figure 7.10.

The meetup2xibo application require access to the entire Xibo scope. Grant access and associate the application with the meetup2xibo user, as shown in Figure 7.11.

Figure 7.2.: Type the username (1) and password (2). Click *Save* (3) to add the new user.
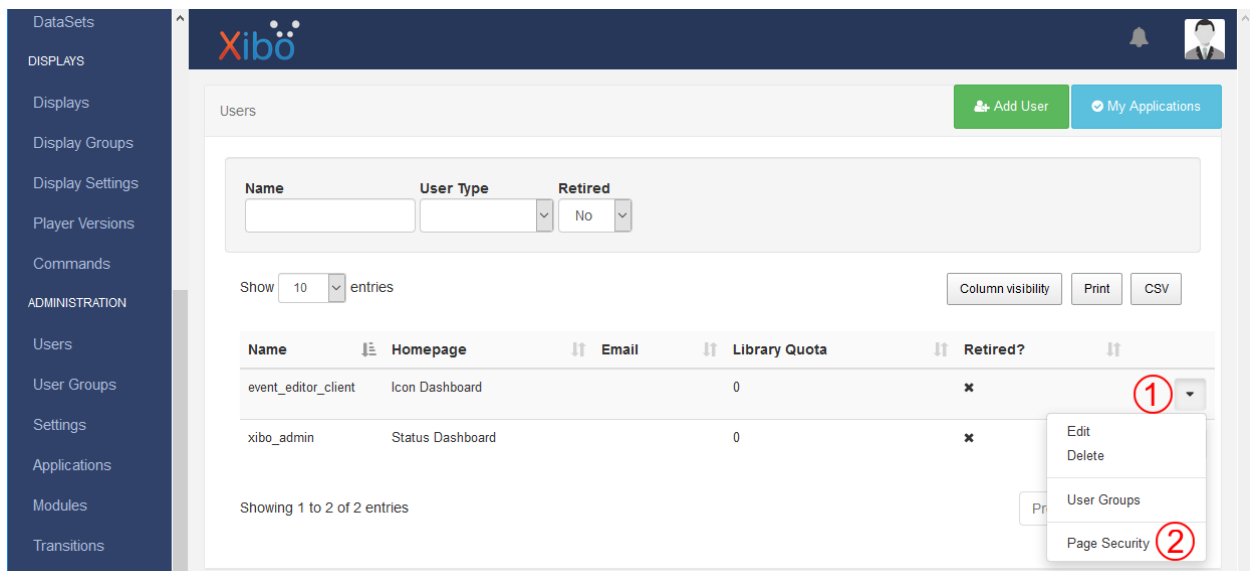
Figure 7.3.: In the meetup2xibo user's row, click the down arrow (1) to open the popup menu. Select *Page Security* (2) from the menu to open the access control list dialog box.

Figure 7.4.: Click the *DataSets* checkbox (1) to authorize use of the datasets API. Scroll down and click *Save* (2) to save the changes.
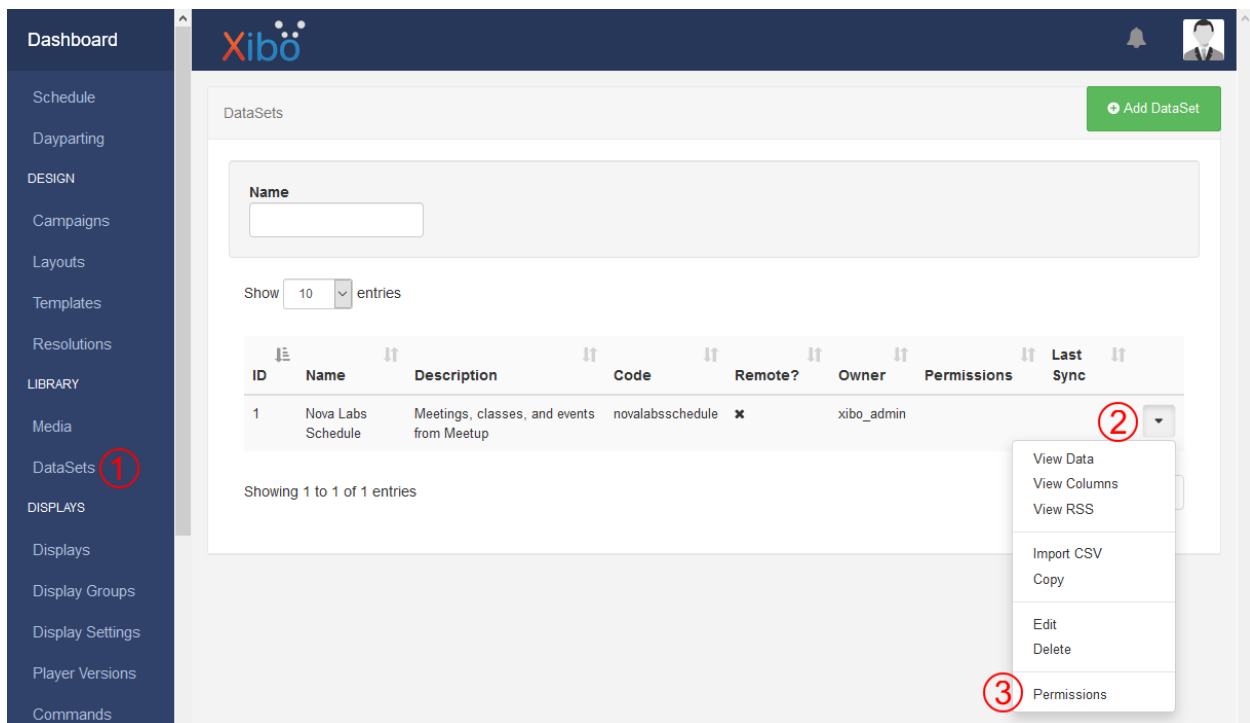
Figure 7.5.: Click *DataSets* (1) in the Xibo CMS menu to display the datasets list. In the event dataset's row, click the down arrow (2) to open the popup menu. Select *Permissions* (3) from the menu to open the permissions dialog box.
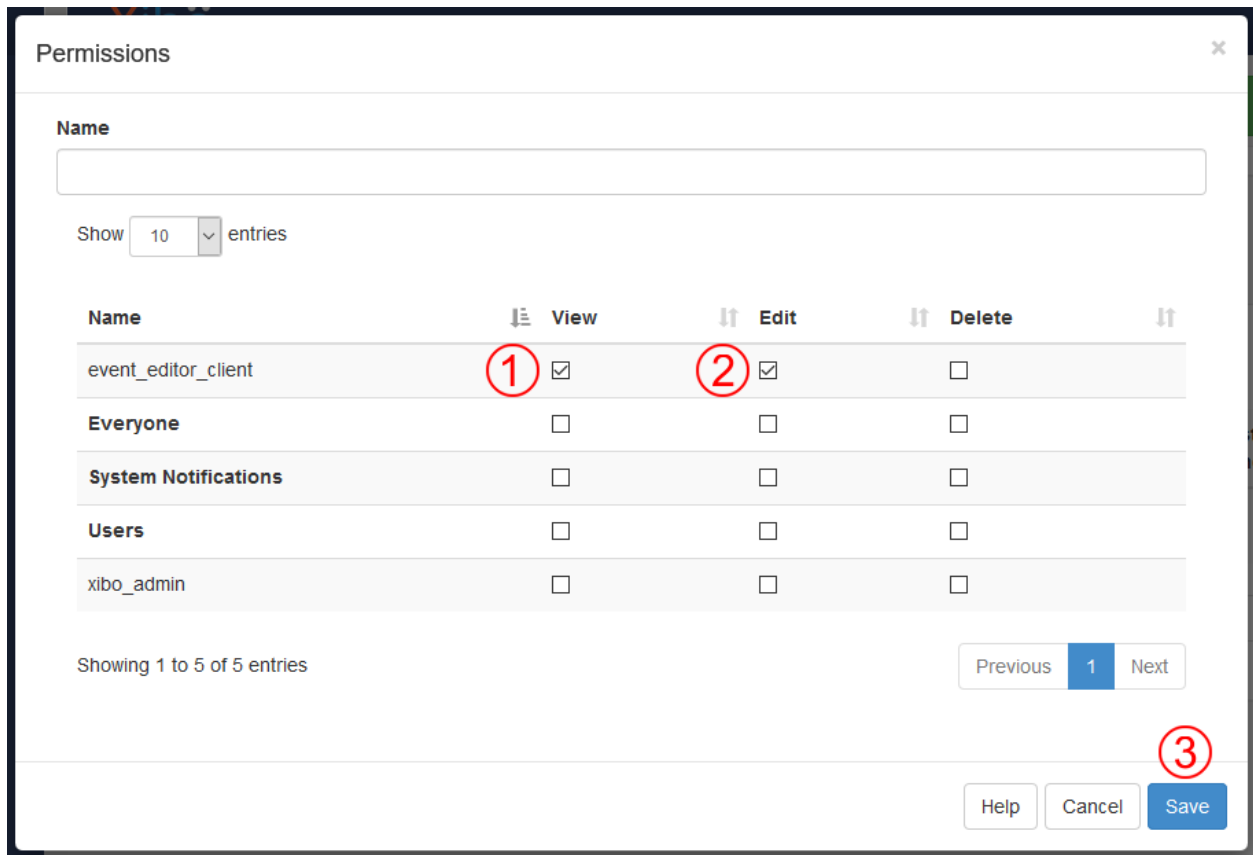
Figure 7.6.: In the row for the meetup2xibo user, click the *View* (1) and *Edit* (2) checkboxes. Click *Save* (3) to save the changes.
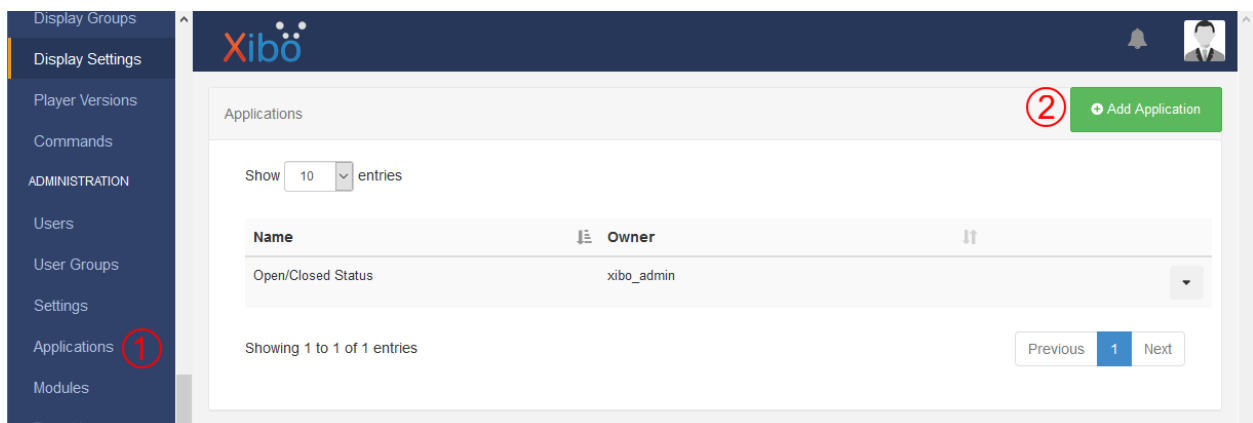
Figure 7.7.: Click *Applications* (1) in the Xibo CMS menu to display the applications list. Click the *Add Application* button (2) to open the *Add User* dialog box.
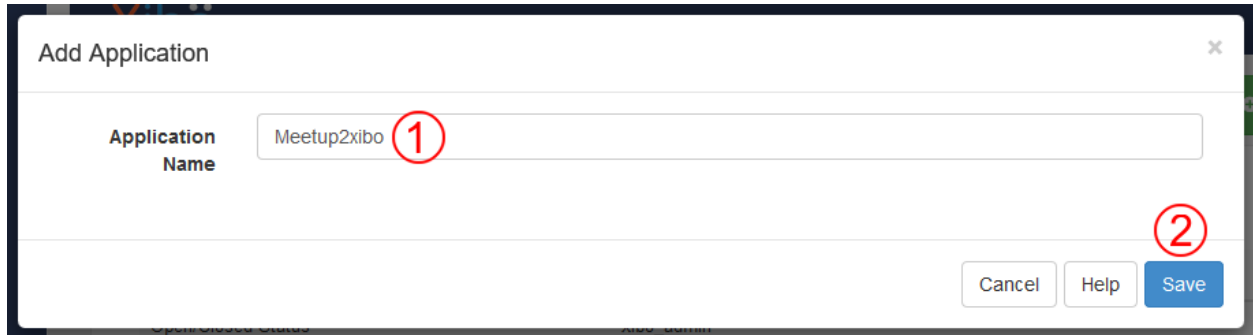
Figure 7.8.: Enter the application name (1). Click *Save* (2) to add the new application.
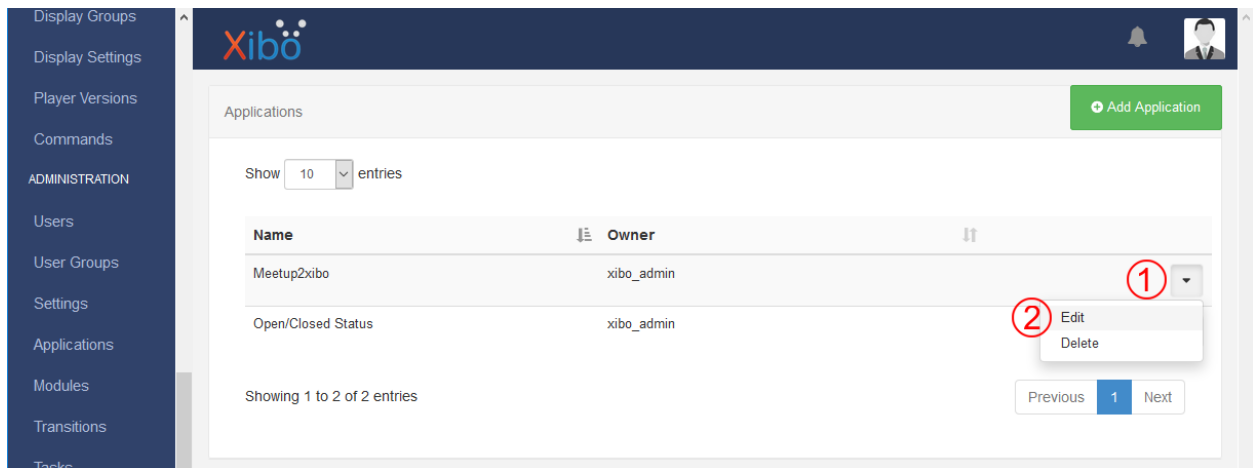
Figure 7.9.: In the meetup2xibo application's row, click the down arrow (1) to open the popup menu. Select *Edit* (2) from the menu to open the *Edit Application* dialog box.

Figure 7.10.: Note the client ID (1) and client secret (2) for the meetup2xibo configuration. Click the *Client Credentials?* checkbox (3) to authorize OAuth2 client credentials. Do not save yet.
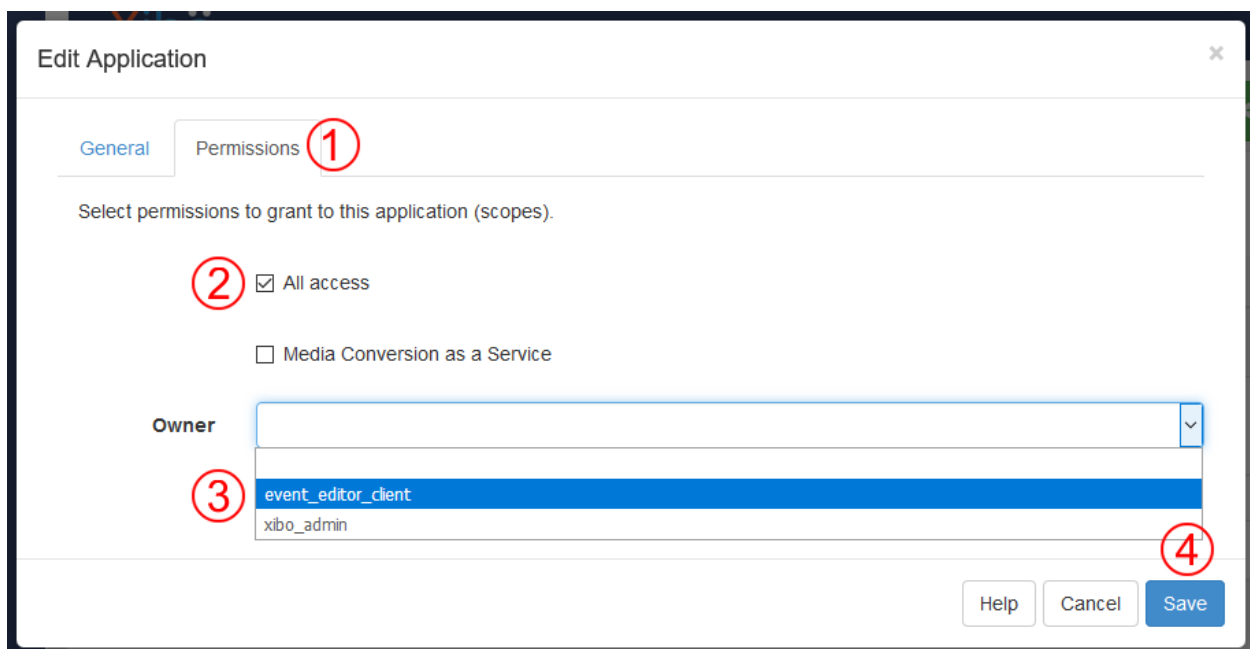
Figure 7.11.: Click the *Permissions* tab (1) to display the application permissions. Click the *All access* checkbox (2) to grant the application access to the entire scope. Select the meetup2xibo user from the pulldown menu (3) to assign an application owner. Click *Save* (4) to save the changes.

Installation

8.1 Stable release

To install meetup2xibo, run this command in your terminal:

```
$ pip install meetup2xibo
```

This is the preferred method to install meetup2xibo, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

8.2 From sources

The sources for meetup2xibo can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/jshprentz/meetup2xibo
```

Or download the tarball:

```
$ curl  -OL https://github.com/jshprentz/meetup2xibo/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

Configuration

As recommended in The Twelve-Factor App, meetup2xibo gets its configuration from environment variables. The *meetup2xibo man page* defines the configuration environment variables. The configuration includes:

- Meetup.com API settings
- Xibo CMS API credentials
- Xibo dataset code and column names
- Xibo web server certificate
- Location corrections
- Date/time thresholds for event insertion, deletion, and cancellation

On Linux, a configuration shell script is a common way to provide environment variables. An annotated example configuration script, /data/meetup2xibo.env, provides a starting point for customization.

The configuration script includes sensitive authorization data, so limit access to the user running meetup2xibo. For example, this chmod command gives the user read and write permissions, and removes permissions from group members and other users:

```
$ chmod u=rw,go-rwx meetup2xibo.env
```

9.1 Meetup.com API Settings

Meetup2xibo requests events for a specific Meetup.com group. The Meetup.com API use a computer-friendly group URL (Uniform Resource Locator) name instead of the actual group name. Locate the group URL name in a web browser address line, as shown in Figure 9.1. Environment variable *MEETUP_GROUP_URL_NAME* provides the URL group name.

By default, the Meetup.com API returns data about the next 200 upcoming events. During 2018 at Nova Labs, this provided about 60 days of future events. By mid-2019, 200 upcoming events provided only 45 days of future events.
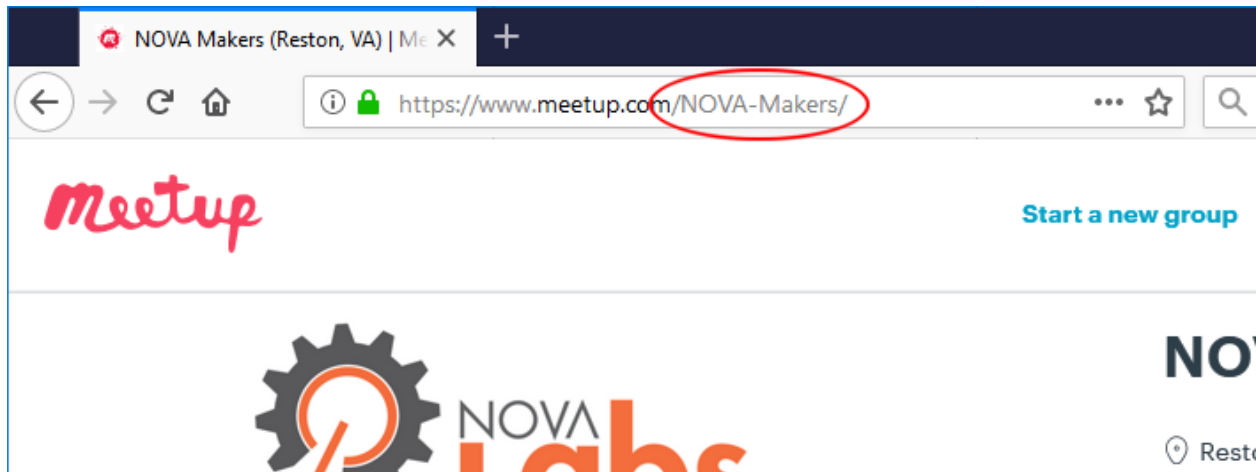
Figure 9.1.: The Meetup.com group URL name, "NOVA-Makers" circled in red, is the last part of the group's Meetup.com URL.

Organizations with more or fewer events may want data about more or fewer than 200 upcoming events. Rather than rely on the Meetup.com default, environment variable `MEETUP_EVENTS_WANTED` specifies the number of upcoming events wanted from Meetup.com.

9.2 Xibo CMS API Credentials

Meetup2xibo connects to the Xibo CMS running on the host specified by environment variable `XIBO_HOST`. This value may be an IP address, a hostname, or a domain name. Meetup2xibo does not need to run on the same computer as the Xibo CMS.

Meetup2xibo connects to the port number specified by environment variable `XIBO_PORT`. The standard SSL/HTTPS port is number 443.

The Xibo CMS authenticates client applications with OAuth2. The client application, meetup2xibo, needs a client ID and a client secret. When a Xibo administrator adds or edits an application, Xibo reveals the client ID and client secret as described in *Authorize the Application* and shown in Figure 7.10. The environment variables `XIBO_CLIENT_ID` and `XIBO_CLIENT_SECRET` provide the necessary credentials to meetup2xibo.

9.3 Xibo Dataset Code and Column Names

The Xibo administrator creates an events dataset through the Xibo CMS web interface as described in *Create an Events Dataset*. The administrator chooses a code to identify the dataset for Xibo API clients, as shown in Figure 6.2. Configure the code in environment variable `EVENT_DATASET_CODE`.

The Xibo administrator creates dataset columns and assigns column headings as described in *Dataset Columns*. Those column headings must be configured in environment variables for meetup2xibo. Table 9.1 lists the environment variables and their corresponding column headings at Nova Labs. Use the column headings chosen in *Dataset Columns*. Environment variable `XIBO_ID_COLUMN_NAME` must have the value *id*, the heading assigned internally by the Xibo CMS.

Table 9.1.: Environment Variables for Dataset Columns

Environment Variable	Column Heading
MEETUP_ID_COLUMN_NAME	Meetup ID
NAME_COLUMN_NAME	Name
LOCATION_COLUMN_NAME	Location
START_TIME_COLUMN_NAME	ISO Start Time
END_TIME_COLUMN_NAME	ISO End Time
XIBO_ID_COLUMN_NAME	id

9.4 Xibo Web Server Certificate

An SSL (Secure Sockets Layer) certificate secures Xibo's web servers. A certificate may be purchased from a certificate authority, a web hosting company, or another provider. Let's Encrypt provides free certificates through its automated web site.

Meetup2xibo's implementation language, Python, recognizes SSL certificates issued by major certificate authorities, including Let's Encrypt.. Meetup2xibo needs no configuration to support these SSL certificates.

Some Xibo installations must rely on self-signed SSL certificates to provide secure access. Digital Ocean has a helpful guide for setting this up: How To Create a Self-Signed SSL Certificate for Apache in Ubuntu 16.04.

When a self-signed certificate protects Xibo's web server, meetup2xibo needs the path to the certificate's public key configured by environment variable *SITE_CA_PATH*.

9.5 Location Corrections

Meetup.com provides many details about meeting venues: name, street address, city, state, country, latitude, longitude, and more. Meetup.com also provides a "how to find us" data field. Most Nova Labs events take place inside Nova Labs, so the agenda displays only the name of the classroom, conference room, or workshop. The Meetup.com venue name and how-to-find-us fields usually contain the room assignment(s), among other information.

Meetup2xibo scans the venue name and "how to find us" fields, searching for known places, such as rooms, workshops, studios, classrooms, fields, etc. The places are collected in the order found. Meetup2xibo corrects place spelling variations and renders the list as an English phrase such as "Room 1, Room 2, and Workshop 3." The resulting phrase is stored in the Xibo events dataset location field.

For example, a woodworking class has the venue name "*Nova Labs (Conference Rm 2)" and the how-to-find-us "[woodshop]". Meetup.com sets the location in Xibo to "Conference Room 2 and Woodshop".

9.5.1 Place Phrases

When scanning the venue name and "how to find us" fields, Meetup2xibo searches for phrases provided in a curated list of phrases and preferred place names. For example, Table 9.2 shows the phrases that match the preferred place name "Conference Room 1."

Table 9.2.: Example Phrases and Places

Phrase	Place
Conf Rm 1	Conference Room 1
Conference rm 1	Conference Room 1
Conference room 1	Conference Room 1

The list of phrases and corresponding places is configured in environment variable *PLACE_PHRASES* as a JSON list of objects, as shown in Listing 9.1.

Listing 9.1: Place Phrases JSON Configuration Example

```
export PLACE_PHRASES='[
    {"phrase": "Conf Rm 1",        "place": "Conference Room 1"},
    {"phrase": "Conference rm 1",   "place": "Conference Room 1"},
    {"phrase": "Conference room 1", "place": "Conference Room 1"}
]'
```

Meetup2xibo ignores spacing and upper/lower case distinctions when searching for phrases. There is no need, for example, to list both "Conf Rm 1" and "Conf rm 1".

Meetup2xibo matches the longest possible phrases first, regardless of their order in the configuration file. For example, phrase "Room 123" would be matched ahead of phrases "Room 1" and "Room 12".

If none of the phrases configured in environment variable *PLACE_PHRASES* match, Meetup2Xibo tries matching the phrases in environment variable *MORE_PLACE_PHRASES*. The format is the same as shown in Listing 9.1. At Nova Labs, *PLACE_PHRASES* lists specific rooms within Nova Labs. *MORE_PLACE_PHRASES* lists more general event places, such as Nova Labs; nearby event places, such as George Mason University; and the uncertain TBD (To Be Determined).

9.5.2 Default Location

If none of the place phrases match, Meetup2xibo uses the location specified by environment variable *DEFAULT_LOCATION*. Meetup2xibo logs a warning message whenever the default location is needed.

To support *conflict detection*, environment variable *DEFAULT_PLACES* lists places associated with the default location. Listing 9.2 shows an example of a default locaion associated with specific places.

Listing 9.2: Default Location with Specific Places Example

```
export DEFAULT_LOCATION="Dance Studios"
export DEFAULT_PLACES='["Studio 1", "Studio 2"]'
```

Listing 9.3 shows default location and places at Nova Labs. The default location does not identify any specific room or workshop, so the default places list is empty.

Listing 9.3: Default Location with No Specific Places Example

```
export DEFAULT_LOCATION="Nova Labs"
export DEFAULT_PLACES='[]'
```

9.5.3 Special Locations

Some Meetup.com events have missing, uncommon, or incorrect locations. Environment variable *SPECIAL_LOCATIONS* contains a list of JSON objects that control the matching process for specific events. Listing

9.4 shows examples of special locations.

Listing 9.4: Special Locations JSON Configuration Example

```
1  export SPECIAL_LOCATIONS='
2  [
3      {
4          "meetup_id": "259083135",
5          "location": "",
6          "places": [],
7          "override": false,
8          "comment": "Electronics 101: no room yet"
9      },
10     {
11         "meetup_id": "gqpyzfbhb",
12         "location": "Classroom A",
13         "places": ["Classroom A"],
14         "override": false,
15         "comment": "Location in event name"
16     },
17     {
18         "meetup_id": "269568127",
19         "location": "Baltimore Museum of Industry",
20         "places": [],
21         "override": false,
22         "comment": "Field trip"
23     },
24     {
25         "meetup_id": "259565142",
26         "location": "Parking Lot, Classroom A, and Conference Room 2",
27         "places": ["Classroom A", "Conference Room 2"],
28         "override": true,
29         "comment": "Picnic"
30     }
31 ]'
```

Within each special location object, the `meetup_id` contains the event ID from Meetup.com. The `location` contains the event location for Xibo. The `location` may be an empty string if the meetup2xibo default location is acceptable. The `places` contains a list of places to check for conflicts. The `override` flag must have a `true` or `false` value as explained below. The `comment` helps administrators remember why the special location was created; it may be an empty string. Missing location warnings are suppressed for events with special location objects.

After the location matching process, meetup2xibo applies these rules to the computed location:

- For events with no special location object, prefer any computed location over the default.

- For events with a special location object and no computed location, prefer any special location over the default.

- For events with both a special location object and a computed location, if the `override` flag is `true`, prefer any special location over the computed location over the default. Otherwise, if the `override` flag is `false`, prefer the computed location over any special location over the default.

The following examples demonstrate the use of special locations:

No Meetup.com venue name or how-to-find-us Meetup2xibo will use the default location and log a warning. When the administrator adds the special location object shown in Listing 9.4 lines 3–9, meetup2xibo will continue to use the default location and places without logging a warning. If the event host later adds a venue name or how-to-find-us in Meetup.com, meetup2xibo will use known locations found there instead of the default location.

Known location only in Meetup.com event name or description Meetup2xibo will use the default location and

---

log a warning. When the administrator adds the special location object shown in Listing 9.4 lines 10–16, meetup2xibo will use the special location instead of the default location. The place "Classroom A" will be checked for conflicts. If the event host later adds a venue name or how-to-find-us in Meetup.com, meetup2xibo will use known places found there instead of the special location or the default location.

Unknown locations from Meetup.com Meetup2xibo will use the default location and log a warning. When the administrator adds the special location object shown in Listing 9.4 lines 17–23, meetup2xibo will use the special location and places instead of the default location and places.

Known and unknown locations from Meetup.com Meetup2xibo will derive a location from the known places found in the venue name or how-to-find-us, "Classroom A" and "Conference Room 2" in this example. Meetup2xibo will not recognize "Parking Lot," an uncommon and unconfigured place. When the administrator adds the special location object shown in Listing 9.4 lines 24–30, meetup2xibo will use the special location and places, overriding the location derived from the known places.

9.6 Detecting Scheduling Conflicts

Meetup2xibo and summarize-m2x-logs can detect and report possible scheduling conflicts. A conflict occurs when two or more events are scheduled in the same place at the same time.

Meetup2xibo checks for conflicts only in places listed by environment variable *CONFLICT_PLACES*. Other places, such as an entire facility (Nova Labs) or a common gathering area (Lobby), can accomodate multiple simultaneous activities. Listing 9.5 shows an example of a list of places with possible scheduling conflicts.

Listing 9.5: Conflict Places Example

```
export CONFLICT_PLACES='[
    "Classroom A",
    "Classroom A/B",
    "Classroom B",
    "Computer Lab",
    "Conference Room 1",
    "Conference Room 2",
    "Conference Room 3"
]'
```

Some places may be subdivided; others may be combined. For example, Classroom A/B may be partitioned into Classroom A and Classroom B. Conferences Rooms 1, 2, and 3 may be combined to form the Conference Center. Meetup2xibo must know which places contain other places to detect scheduling conflicts such as simultaneous meetings in Classroom A and Classroom A/B.

Environment variable *CONTAINED_PLACES* lists JSON objects describing places containing other places. Listing 9.6 shows the example above of Classroom A/B and the Conference Center.

Listing 9.6: Contained Places Example

```
export CONTAINING_PLACES='[
    {"place": "Classroom A/B",
     "contains": ["Classroom A", "Classroom B"]},
    {"place": "Conference Center",
     "contains": ["Conference Room 1", "Conference Room 2", "Conference Room 3"]}
]'
```

Meetup2xibo supports typical facilities with few restrictions.

Places may be listed in any order. Places are analyzed by name, not by their position in lists.

Contained places may be nested. For example, the Convention Center may contain the Conference Center among other places.

Contained places may overlap. A Seminar Area may contain Classroom A and Conference 1, overlapping the example places Classroom A/B and Conference Center.

Contained and containing places need not be checked for conflicts. For example, the Conference Center is not on the list of `CONFLICT_PLACES`.

Loops are forbidden. The Conference Center may not contain the Convention Center if the Convention Center already contains the Conference Center.

9.7 Suppressed Events

Xibo should not display some placeholder and private events. Environment variable *SUPPRESS_MEETUP_IDS* contains a list of Meetup IDs of events to suppress from Xibo displays. Suppressed events will not be analyzed for locations or checked for conflicts.

9.8 Timezone

Meetup2xibo converts event start and end times from UTC (Coordinated Universal Time), the internal format of Meetup.com, to the timezone configured in the Xibo CMS. Figure 9.2 shows how to review or change the Xibo timezone.
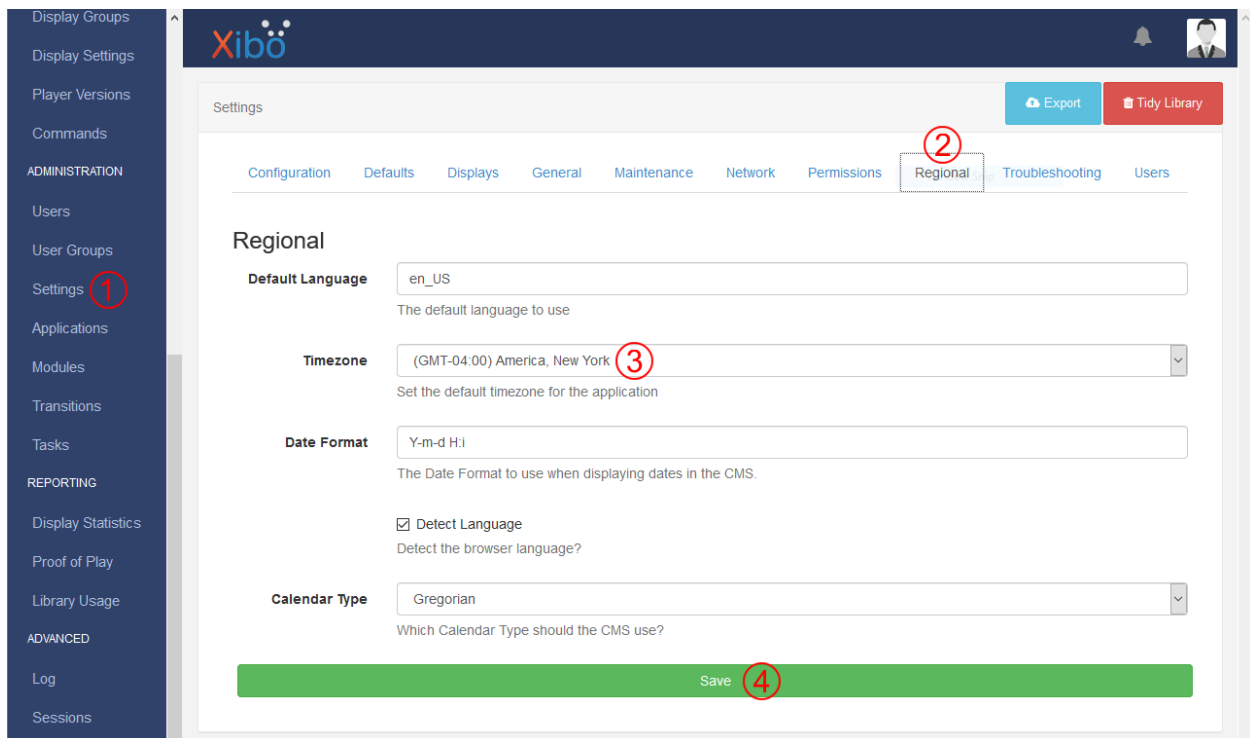
Figure 9.2.: Click *Settings* (1) in the Xibo CMS menu to display some of Xibo's settings. Click the *Regional* tab (2) to display settins that vary by region. Review or change the timezone (3) using the dropdown menu. Click the *Save* button (4) to save changes, if necessary.

The timezone specified by environment variable *TIMEZONE* must correspond to the setting in Xibo, but the format is slightly different. Meetup2xibo follows the naming rules for the tz database. Wikipedia provides a convenient List of tz database time zones. For example, Nova Labs, near Washington, DC, set the Xibo timezone to "(GMT-04:00) America, New York" in August 2017, a daylight savings time month.. The corresponding tz database name is "America/New_York".

9.9 Date/time Thresholds

The Meetup.com API sometimes fails to list current and very recent events while Xibo still should display them. Events in the Xibo dataset and missing from Meetup.com should not be deleted unless they start some number of hours in the future, configured by environment variable *DELETE_BEFORE_START_HOURS*.

Some Xibo displays, such as daily agendas and weekly calendars, should continue to display events after their conclusion. Past events should not be delegated from the Xibo dataset until some number of hours after they end, as configured by environment variable *DELETE_AFTER_END_HOURS*.

The Meetup.com API lists some fixed number (hundreds) of future events. As new near-term events are added to the Meetup.com calendar, some previously reported far future events are bumped from the listing. To avoid flapping, meetup2xibo does not delete events from the Xibo dataset more than some number of days in the future, as configured by environment variable *DELETE_UNTIL_FUTURE_DAYS*.

When Meetup.com events are cancelled far in the future, guests have ample time to receive notifications and alter thier plans. Meetup2xibo ignores cancelled events after some number of days in the future, configured by environment variable *IGNORE_CANCELLED_AFTER_DAYS*, treating them as deleted.

Logging

Meetup2xibo logs its activity to a log file and optionally to the console *(stderr)*. Command line flags control which log messages should be directed to which destinations. Supporting program **summarize-m2x-logs** summarizes log files, reporting on meetup2xibo runs and event changes.

10.1 Log Destinations

By default, meetup2xibo writes log messages to file meetup2xibo.log in the current directory. The `-t` option can specify an alternate path to the log file. The `-v` option adds console output. The `-w` option adds console output and limits the console log level to warnings and errors.

10.2 Log Levels

By default, meetup2xibo logs only info, warning, and error messages. The `-d` option adds debug level messages, which help software developers. As described above, the `-w` option adds console output and limits the console log level to warnings and errors. Table 10.1 summarizes the log levels and the impact of the `-d` and `-w` flags.

Table 10.1.: Meetup2xibo Log Levels

Log Level	Reports	Default	-d Flag	-w Flag (console)
Error	Failures	✓	✓	✓
Warning	Abnormal conditions	✓	✓	✓
Info	Routine operations	✓	✓	
Debug	Developer details		✓	

10.3 Log Summaries

Summarize-m2x-logs summarizes one or more meetup2xibo log files. The summary includes:

- Counts of meetup2xibo runs, by version

- Lists of schedule conflicts

- Lists of events inserted, updated, deleted, and retired

- Field-by-field details of updated events

- Warnings (in red)

  - About today's schedule conflicts

  - About unknown event locations

  - About special event locations no longer needed

  - About suppressed Meetup.com event IDs no longer needed

- A spreadsheet showing conversions from Meetup.com locations to Xibo locations

Summarize-m2x-logs Cron Job describes how to run summarize-m2x-logs daily and send the HTML-formatted results via email. Figure 10.1 shows an example of the daily email message.

An email message attachment contains a CSV formatted file listing the conversions from Meetup.com venue name and how-to-find-us fields to Xibo locations. An administrator can review the CSV file in Microsoft Excel or another spreadsheet program. The administrator can resolve location conversion anomalies by editing the Meetup.com event or by adding a new location phrase or special location to the meetup2xibo configuration. Figure 10.2 shows an example of the location conversion spreadsheet.

Meetup to Xibo Summary for 2019-07-25 🖨 ☑

Joel Shprentz   ✇ Jul 26, 2019, 1:44 AM (1 day ago)   ☆ ↩ ⋮
to jshprentz, events ▾

Program Execution

Version	Count
meetup2xibo 3.0.0	145

Schedule Conflicts

Conference Room 2

2019-08-08 19:00 to 2019-08-08 21:00

- **Going Pro? Turning a Hobby into a Business**
  2019-08-08 19:00 – 2019-08-08 21:00
  at Conference Room 2
- **Introduction to Basic Electronic Circuits**
  2019-08-08 19:00 – 2019-08-08 21:00
  at Conference Room 2

Changes to Xibo Current and Future Event Listings

Enco Metal Lathe Red Sign Off
2019-07-27 14:00 – 2019-07-27 16:00
at Cancelled

> 2019-07-25 22:33 **Updated location**
>   From: Conference Room 1 and Metal Shop
>   To: Cancelled

Nova Labs Roundup - Volunteers aplenty Needed
2019-08-17 09:00 – 2019-08-17 11:00
at Nova Labs

> 2019-07-25 16:33 **Unknown Location**
>   Nova Labs Roundup - Volunteers aplenty Needed
>   2019-08-17 09:00 – 2019-08-17 11:00
>   Venue:
>   Find Us:

> 2019-07-25 16:33 **Inserted**

Figure 10.1.: The daily summary email message counts meetup2xibo runs, lists schedule conflicts, and lists Xibo event changes.

Location	Venue	Find Us	Example Meetup	Example URL
Blacksmithing Alley outside behind N	Nova Labs (Blacksmithing Alley)	[Outback in Blacksmithing alley, beh	Blacksmithing 101 Safety Sign off	https://www.meetup.com/NOVA-Makers/events/260057564/
CAD Lab	Nova Labs CAD Lab		CAD and 3D Modeling Help and Instruction	https://www.meetup.com/NOVA-Makers/events/pgqrtqyzhbvb/
Classroom A	*Nova Labs (Classroom A)		NoVA Maker Faire 2019 Organizing Team - Come	https://www.meetup.com/NOVA-Makers/events/tmnbrqyzhbzb/
Classroom A	*Nova Labs (Classroom A)	[]	Nova Labs Drone Meetup	https://www.meetup.com/NOVA-Makers/events/hrljnqyzhbgb/
Classroom A	Nova Labs	[Classroom A]	International Costumers' Guild: NoVA Chapter W	https://www.meetup.com/NOVA-Makers/events/258575608/
Classroom A	Nova Labs (Classroom A)		Introduction to KiCad & FreeCAD (Day 2 of 2)	https://www.meetup.com/NOVA-Makers/events/259916495/
Classroom A	Nova Labs (Classroom A)	[Classroom A]	Drone World Speed Record Build Team	https://www.meetup.com/NOVA-Makers/events/sbmmtqyzhbwb/
Classroom A/B	*Nova Labs (Classroom A/B)		FLL Skillbuilding Workshop #4	https://www.meetup.com/NOVA-Makers/events/258516983/
Classroom A/B	*Nova Labs (Classrooms A and B)		NoVA Hackers at NoVA Labs	https://www.meetup.com/NOVA-Makers/events/ktgjhqyzhbrb/
Classroom B	*Nova Labs (Classroom B)		WIT Meeting MB Private (Classroom A)	https://www.meetup.com/NOVA-Makers/events/tkhtpqyzgbfc/
Classroom B	*Nova Labs (Classroom B)	[ ]	Raspberry Pi Users Meetup	https://www.meetup.com/NOVA-Makers/events/gxlqjpyzgbfc/
Classroom B	*Nova Labs (Classroom B)	[]	3D Scanner Sign off - Create 3D models of physic	https://www.meetup.com/NOVA-Makers/events/259482614/
Conference Room 1	*Nova Labs (Conference Rm 1)		Make Your Own Kitchen Knife Day 2 of 2	https://www.meetup.com/NOVA-Makers/events/260123806/
Conference Room 1	*Nova Labs (Conference Rm 1)	Nova Labs is a short walk north from	Computational Mathematics: P=NP for students	https://www.meetup.com/NOVA-Makers/events/ftqfsqyzhbcc/
Conference Room 1	*Nova Labs (Conference Rm 1)	Parking Available; 6 minute walk to t	Thermoforming 101 - Making plastic parts from s	https://www.meetup.com/NOVA-Makers/events/260067815/
Conference Room 1	*Nova Labs (Conference Rm 1)	[ ]	Craig Trader's Open Office Hours	https://www.meetup.com/NOVA-Makers/events/snvpsqyzhbdc/
Conference Room 1	*Nova Labs (Conference Rm 1)	[]	RH team meeting (private)	https://www.meetup.com/NOVA-Makers/events/jzmqcqyzhbsb/
Conference Room 1	Nova Labs (Conf Rm 1)		Scythe Campaign (Private)	https://www.meetup.com/NOVA-Makers/events/lmpttqyzhbsb/
Conference Room 1 and Metal Shop	*Nova Labs (Conference Rm 1)	[Conference rm, followed by Metal s	Enco Metal Lathe Red Sign Off - Assocates and Ko	https://www.meetup.com/NOVA-Makers/events/259914012/
Conference Room 2	*Nova Labs (Conference Rm 2)		Build a humanitarian drone with Team RhinoHav	https://www.meetup.com/NOVA-Makers/events/rxsssqyzhbcc/
Conference Room 2	*Nova Labs (Conference Rm 2)	[]	Silhouette CNC Yellow Sign off	https://www.meetup.com/NOVA-Makers/events/260148733/
Conference Room 2 and Woodshop	*Nova Labs (Conference Rm 2)	[woodshop]	Woodshop Router Red Safety Sign Off	https://www.meetup.com/NOVA-Makers/events/260164588/
Conference Room 3	*Nova Labs (Conference Rm 3)		Rockets and Spacecraft MeetUp	https://www.meetup.com/NOVA-Makers/events/jgntmqyzhbzb/
Conference Room 3	*Nova Labs (Conference Rm 3)	Parking available. 6 minute walk fro	Laser Cutter 101 Red Safety Sign Off - Empower2	https://www.meetup.com/NOVA-Makers/events/260067320/
Conference Room 3	*Nova Labs (Conference Rm 3)	[ ]	Arduino group meeting	https://www.meetup.com/NOVA-Makers/events/zngfnqyzhbcc/

Figure 10.2.: The first several lines of location conversions list in CSV format and viewed as a spreadsheet in Microsoft Excel.

Cron Jobs

In practice, **meetup2xibo** and **summarize-m2x-logs** run unattended as cron jobs.

11.1 Cron Setup

Cron runs jobs listed in a *crontab* file. This file must be edited with the **crontab** program:

```
$ crontab -e
```

The example cron jobs below are written for the **bash** shell. Add this line to the crontab to select that shell:

```
SHELL=/bin/bash
```

When a cron job produces any output to *stdout* or *stderr* (standard output and standard error, respectively), cron can send those results via email. Add this line to the crontab to specify the email recipient:

```
MAILTO=chris.jones@example.com
```

The cron jobs run meetup2xibo, summarize-m2x-logs, and yesterday, all programs installed by the meetup2xibo package. The **which** command will reveal where meetu2xibo and the other programs were installed:

```
$ which meetup2xibo
/home/joels/.virtualenvs/meetup2xibo/bin/meetup2xibo
```

Add this line to the crontab to extend the system path with the path to the meetup2xibo programs:

```
PATH=~/.virtualenvs/meetup2xibo/bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

11.2 Meetup2xibo Cron Jobs

Cron runs meetup2xibo frequently to keep the agenda displays up-to-date with Meetup.com. Most events start on the hour, at a quarter past, or half past the hour, so run times are slightly offset ahead of event start times.

The following crontab line schedules ordinary meetup2xibo runs every 10 minutes:

```
3,13,23,33,43,53 * * * * cd ~/meetup2xibo && . ./meetup2xibo.env && meetup2xibo -w
```

The `-w` option directs warnings to standard error; cron will send them via email to the cron user. Meetup2xibo will log info and higher messages to the default `meetup2xibo.log` file.

Late each day, cron runs meetup2xibo with two extra options:

- `-c` to log schedule conflicts
- `-m` to log all location mappings from Meetup.com's venue name and how-to-find-us fields to Xibo's location

This run is scheduled to avoid the every-10-minute runs. The following crontab line schedules a schedule conflict and location mapping meetup2xibo run every night before midnight:

```
48 23 * * * cd ~/meetup2xibo && . ./meetup2xibo.env && meetup2xibo -w -c -m
```

11.3 Summarize-m2x-logs Cron Job

Cron runs summarize-m2x-logs early every morning to summarize the previous day's meetup2xibo activity and mail the results to interested people. The time was chosen to take place after daily log rotation, to have low impact on computer resources, and to provide a morning email to administrators. The following crontab line schedules summarize-m2x-logs while administrators are sleeping:

```
44 1 * * * YESTERDAY=`yesterday` ; summarize-m2x-logs
-t "jmith@example.com events@example.com"
-s "Meetup to Xibo Summary for ${YESTERDAY}"
~/meetup2xibo/meetup2xibo.log.${YESTERDAY}
| ssmtp -t
```

These five lines must be combined into a single crontab line.

Yesterday's date is saved in `YESTERDAY` for substituion later on the line. The `-t` option selects email-formatted output and provides a space-separated list of email recipients. The `-s` option provides the email subject with yesterday's date inserted. Summarize-m2x-logs takes yesterday's log file, `~/meetup2xibo/meetup2xibo.log.${YESTERDAY}`, with yesterday's date appended to the filename.

Cron pipes the email output to the **sSMTP**, a lightweight mail transport agent that handles only outbound emails. On a more capable computer, **sendmail** could be used instead.

meetup2xibo

12.1 Synopsis

meetup2xibo [-h] [-d] [-l *<LOGFILE>*] [-c] [-m] [-v] [-w]

12.2 Description

meetup2xibo downloads events from Meetup.com and updates the corresponding events stored in a Xibo CMS dateset.

meetup2xibo reads its configuration from *environment variables*. The *command line options* control only the message levels to log and where to send the logs.

12.3 Options

-h, --help
   Show a help message and exit.

-d, --debug
   Also log debug messages. If this option is omitted, log only info and higher level messages.

-l <LOGFILE>, **--logfile** <LOGFILE>
   Path to logfile (default: meetup2xibo.log).

-c, --conflicts
   Log conflict detection details about Meetup events and the places where they are scheduled. Default: do not log conflict details.

   New in version 3.0.

-m, --mappings
> Log location mappings from Meetup.com's venue name and find-us fields to Xibo's location field. Default: log mappings only with debug messages.

-v, --verbose
> Log to standard error. This is useful for debugging when running from the command line.

-w, --warnings
> Log warnings and higher level messages to standard error. This is useful when running in cron job because cron will mail any standard error output.

12.4 Environment

CONFLICT_PLACES
> A JSON array of places to check for scheduling conflicts.
>
> New in version 3.0.

CONTAINED_PLACES
> A JSON array of objects showing a place and the other places it contains For example:

```
export CONTAINING_PLACES='[
    {"place": "Ballroom", "contains": ["North Ballroom", "South Ballroom"]},
    {"place": "Ballroom Suite", "contains": ["Ballroom", "Dressing Room"]},
]'
```

> New in version 3.0.

DEFAULT_LOCATION
> The location to store in Xibo when Meetup's venue name and find-us fields contain no recognizable locations.

DEFAULT_PLACES
> A JSON array of places associated with the default location.
>
> New in version 3.0.

DELETE_AFTER_END_HOURS
> The number of hours after an event ends to wait before retiring the event from Xibo.

DELETE_BEFORE_START_HOURS
> The number of hours before an event starts required to delete from Xibo an event not listed by Meetup.

DELETE_UNTIL_FUTURE_DAYS
> The number of days in the future when an event not listed by Meetup may be deleted from Xibo.

END_TIME_COLUMN_NAME
> The name of the Xibo dataset column containing event end times in ISO 8601 format.

EVENT_DATASET_CODE
> The API code assigned to the Xibo event dataset.

IGNORE_CANCELLED_AFTER_DAYS
> The number of days in the future to ignore cancelled events and quietly delete them from Xibo.

LOCATION_COLUMN_NAME
> The name of the Xibo dataset column containing event locations.

MEETUP_API_KEY
> The API key that authenticates access to Meetup.com.

Deprecated since version 3.1: The Meetup API key no longer is sent to Meetup.com. This environment variable may be removed from configurations.

MEETUP_EVENTS_WANTED
> The number of events to request from Meetup.

MEETUP_GROUP_URL_NAME
> The group name for Meetup URLs. For example, in the URL https://www.meetup.com/NOVA-Makers/, the group name is *NOVA-Makers*.

MEETUP_ID_COLUMN_NAME
> The name of the Xibo dataset column containing Meetup event IDs.

MORE_PLACE_PHRASES
> A second list of phrases and places to try if *PLACE_PHRASES* failed to match. See *PLACE_PHRASES* for the JSON format.
>
> Changed in version 3.0: Was MORE_LOCATION_PHRASES with a different format.

NAME_COLUMN_NAME
> The name of the Xibo dataset column containing event names.

PLACE_PHRASES
> A JSON array of objects containing a phrase to match and a corresponding place. For example:

```
export PLACE_PHRASES='[
    {"phrase": "Conf Rm 1",         "place": "Conference Room 1"},
    {"phrase": "Conf Rm 2",         "place": "Conference Room 2"},
    {"phrase": "Conference room 1", "place": "Conference Room 1"},
    {"phrase": "Conference room 2", "place": "Conference Room 2"}
]'
```

> Changed in version 3.0: Was LOCATION_PHRASES with a different format.

SITE_CA_PATH
> The optional path to a self-signed certificate for Xibo.

SPECIAL_LOCATIONS
> A JSON array of objects that correct or override missing, incorrect, or verbose event locations from Meetup. For example:

```
export SPECIAL_LOCATIONS='[
    {"meetup_id": "zvbxrpl2", "location": "Orange Bay",
     "comment": "", "override": false, "places": ["Orange Bay"]},
    {"meetup_id": "lrzzfbhb", "location": "Private",
     "comment": "Private meeting", "override": true, "places": []}
]'
```

> List *meetup_id* to suppress warnings about missing locations. A non-blank *location* replaces the default location and can replace the computed location. Any *comment* helps document the special location. When true, the *override* flag forces a non-blank location to replace the computed location. Conflict resolution examines the list of *places* associated with the location.
>
> Changed in version 3.0: Added *places*.

START_TIME_COLUMN_NAME
> The name of the Xibo dataset column containing event start times in ISO 8601 format.

SUPPRESS_MEETUP_IDS
> A JSON array of Meetup IDs for events that Xibo should not display. For example:

```
export SUPPRESS_MEETUP_IDS='[
    "266192589",
    "tdwljryzqbjb"
]'
```

New in version 3.3.

TIMEZONE
The timezone database name of the timezone configured in Xibo. For example, `America/New_York`. See the list of timezones for timezone database names.

XIBO_CLIENT_ID
The client ID that identifies this application to Xibo.

XIBO_CLIENT_SECRET
The client secret that authenticates this application to Xibo.

XIBO_HOST
The hostname or IP address of the Xibo CMS server.

XIBO_ID_COLUMN_NAME
The name of the Xibo dataset column containing Xibo event IDs.

XIBO_PORT
The port number of the Xibo CMS server, usually 443.

summarize-m2x-logs

13.1 Synopsis

summarize-m2x-logs [-h] [-m] [-s *<EMAIL_SUBJECT>*] [-t *<EMAIL_TO>*] [*<INFILE>*] [*<OUTFILE>*]

13.2 Description

`summarize-m2x-logs` summarizes logs generated by `meetup2xibo`.

INFILE is the path to a log file. If omitted, it defaults to standard input. *OUTFILE* is the path to the output file. If omitted, it defaults to standard output.

The HTML-formatted default summary report contains three sections:

Program Execution Executions counts of various versions of `meetup2xibo`.

Changes to Xibo Current and Future Event Listings Descriptions of inserted, updated, and cancelled events.

Changes to Xibo Past Event Listings Descriptions of retired events.

The `--mappings` option replaces the default summary report with a CSV-formatted list of location mappings. Each mapping shows how Meetup.com's venue name and find-us map to a Xibo event location field value. The mappings include links to example events at Meetup.com.

The `--to` option sends output via email instead of to a file. The email message body contains the default summary report. An attachment containing a CSV-formatted list of location mappings will be sent if any mapping log lines are found. The `--to` and `--subject` options provide the email message destination and subject.

13.3 Options

-h, --help
    Show a help message and exit.

-m, --mappings
    Summarize location mappings in CSV format. (default: summarize logs in HTML format)

-s <EMAIL_SUBJECT>, **--subject** <EMAIL_SUBJECT>
    Email subject. (default: Meetup to Xibo log summary)

-t <EMAIL_TO>, **--to** <EMAIL_TO>
    Generate an email message to this address or to space separated addressses. Overrides –mappings option. (default: no email message)

yesterday

14.1 Synopsis

yesterday

14.2 Description

Display the yesterday's date in ISO 8601 format.

14.3 Example

On April 1, 2019,

```
$ yesterday
```

displays this output:

```
2019-03-31
```

Frequently Asked Questions

15.1 How is meetup2xibo pronounced?

Meetup to Xibo

15.2 How is Xibo pronounced?

See Introducing Xibo.

15.3 Explain the feature "Remove accounting codes from event titles"

Most Nova Labs event names begin with a two-character accounting code such as *EL* for electronics. Meetup2xibo removes accounting codes while converting events from Meetup.com format to Xibo format. For example, "EL: Electronics 101" becomes "Electronics 101." This feature is not configurable.

Credits

16.1 Development Lead

- Joel Shprentz (@jshprentz).

16.2 Contributors

None yet. Why not be the first?

16.3 Python Packages

Meetup2xibo builds upon these excellent library packages:

- jinja2, a template engine
- oauthlib, OAuth request-signing logic
- parsley, a parsing library that uses the PEG algorithm
- pyahocorasick, multi-pattern string search using the Aho-Corasick algorithm
- pytz, the Olson timezone database
- requests, HTTP requests for humans
- requests_oauthlib, OAuth library support for Requests, OAuth for humans
- requests-toolbelt, utilities for requests

16.4 Testing Packages

These Python packages help test **Meetup2xibo**.

- hypothesis, property based testing

- pytest-mock, mock objects for pytest

- pytest, a testing framework

16.5 Development Tools

These tools help build, document, test, and validate **Meetup2xibo**.

- bumpversion updates version numbers in source code.

- Cookiecutter and the audreyr/cookiecutter-pypackage project template provided the Python package structure.

- coverage measures code coverage during tests.

- flake8 checks code style.

- Sphinx renders documentation in HTML, PDF, and other formats.

- tox automates testing with multiple Python versions.

- twine publishes Python packages to PyPi.

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

17.1 Types of Contributions

17.1.1 Report Bugs

Report bugs at https://github.com/jshprentz/meetup2xibo/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

17.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

17.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

17.1.4 Write Documentation

Meetup to Xibo could always use more documentation, whether as part of the official Meetup to Xibo docs, in docstrings, or even on the web in blog posts, articles, and such.

17.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/jshprentz/meetup2xibo/issues.

If you are proposing a feature:

- Explain in detail how it would work.

- Keep the scope as narrow as possible, to make it easier to implement.

- Remember that this is a volunteer-driven project, and that contributions are welcome :)

17.2 Get Started!

Ready to contribute? Here's how to set up *meetup2xibo* for local development.

1. Fork the *meetup2xibo* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/meetup2xibo.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv meetup2xibo
$ cd meetup2xibo/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 meetup2xibo tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

17.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.

3. The pull request should work for Python 3.5, 3.6, and 3.7. Check https://travis-ci.org/jshprentz/meetup2xibo/ pull_requests and make sure that the tests pass for all supported Python versions.

17.4 Tips

To run a subset of tests:

```
$ py.test tests.test_meetup2xibo
```

17.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

History

18.1 Future History

- Support Xibo 2.

18.2 Recent History

18.2.1 3.3.1 (2019-12-02)

- Report unneeded suppressed Meetup event IDs when summarizing logs.

18.2.2 3.3.0 (2019-11-19)

- Suppress processing and display of configured events.
- Add SUPPRESSED_EVENT_IDS environment variable.
- Emphasize today's schedule conflicts.

18.2.3 3.2.4 (2019-11-04)

- Highlight today's conflicts in red in log summaries.

18.2.4 3.2.3 (2019-10-01)

- Accept special location places when summarizing logs.

18.2.5 3.2.2 (2019-08-01)

- Improve reporting of missing environment variables.

18.2.6 3.2.1 (2019-08-01)

- Improve reporting of JSON configuration errors.
- Log errors during initialization.

18.2.7 3.2.0 (2019-07-29)

- Document quick start import of dataset structure.

18.2.8 3.1.0 (2019-07-27)

- Meetup API key no longer is sent to Meetup.com.
- Environment variable `MEETUP_API_KEY` may be removed from configurations.

18.2.9 3.0.1 (2019-07-27)

- Document logging and reporting schedule conflicts.
- Fix documentation issues.

18.2.10 3.0.0 (2019-07-25)

- Detect and report room conflicts.

18.2.11 2.4.10 (2019-06-12)

- Report reversion to earlier unknown locations in log summaries.
- Improve reporting of OAuth2 errors.

18.2.12 2.4.9 (2019-05-30)

- Improve reporting of Xibo dataset discovery errors.

18.2.13 2.4.8 (2019-05-29)

- Improve reporting of Meetup.com server errors.
- Add Python 3.7 tests to Travis CI.
- Change development status from beta to production.

18.2.14 2.4.7 (2019-05-27)

- Add missing modules to distribution manifest.

18.2.15 2.4.6 (2019-05-27)

- First release on PyPI.
- Document location conversion spreadsheet

18.2.16 2.4.5 (2019-05-26)

- Complete user manual.

18.2.17 2.4.4 (2019-05-25)

- Expand documentation.
- Test with Python 3.7 also.

18.2.18 2.4.3 (2019-05-08)

- Add documentation.

18.2.19 2.4.2 (2019-04-25)

- Correct conversion of cancelled events.
- Add man pages to documentation.

18.2.20 2.4.1 (2019-03-26)

- Ignore "3D" accounting code in event names.
- Summarize past and current changes in separate lists.

18.2.21 2.4.0 (2019-03-21)

- Report event updates with human-friendly field names (e.g., start time instead of start_time).
- Add TIMEZONE environment property.
- Make Meetup to Xibo event conversions timezone aware.

18.2.22 2.3.2 (2019-03-18)

- Correct filename travis.yml to .travis.yml.

18.2.23 2.3.1 (2019-03-18)

- Add Travis CI configuration.

18.2.24 2.3.0 (2019-03-18)

- Summarize location mappings in CSV format.
- Describe past deleted events as "retired."
- Improve email header generation.

18.2.25 2.2.2 (2019-03-15)

- Add configuration parameter for the number of Meetup events to request.
- Improve reporting of HTTP errors.

18.2.26 2.2.1 (2019-03-14)

- Add special locations no longer needed to log summary reports.
- Improve formatting of log summary reports.
- Move xibo_id to end of XiboEvent tuples to ease visual comparison with PartialEvent tuples in logs.

18.2.27 2.2.0 (2019-03-13)

- Add comand line arguments to log summarizer: input, output, email headers.
- Add yesterday script.
- Simplify all times to HH:MM, omitting seconds.
- Report unknown location warnings.

18.2.28 2.1.0 (2019-03-13)

- Add tools to summarize meetup2xibo logs

18.2.29 2.0.1 (2019-03-06)

- Improve location analysis and overrides.
- Track cancelled events.
- Add documentation with Sphinx and ReadTheDocs.
- Replace MySQL access with Xibo web API.
- Replace Python configuration file with environment based configuration.
- Restructure code with dependency injection.
- Add Python package support.

- Reformat to conform to PEP-8.

- Test with multiple Python versions (3.5 and 3.6).

18.3 Ancient History

18.3.1 1.0.1 (2017-12-12)

- Download events from Meetup web API.

- Insert/update/delete corresponding Xibo events directly in Xibo's MySQL database.

CHAPTER 19

Indices and tables

- genindex

- search

Index

Symbols