Django Major Event Log Documentation

Release 3.0.0

University of North Texas Libraries

Sep 19, 2023

Contents

This Django application is designed to keep track of digital preservation events using the PREMIS Event model. The app takes care of both creating and viewing these events. The admin site is responsible for managing the actual events, which includes creating them as well as making modifications to the event date, contact info, description, etc. The other pages are responsible for displaying events with various levels of detail.

CHAPTER 1

Table of Contents

1.1 Installation

1.1.1 Requirements

- Django 4.2
- Python 3.8 - 3.10

1.1.2 Installing

1. Install the app

```
$ pip install major-event-log
```

2. Add app and all dependencies to INSTALLED_APPS.

```
INSTALLED_APPS = (
  'django.contrib.admin',
  'django.contrib.auth',
  'django.contrib.contenttypes',
  'django.contrib.sessions',
  'django.contrib.messages',
  'django.contrib.staticfiles',
  'major_event_log',
)
```

3. Include the URLs

```
urlpatterns = [
  path('admin/', admin.site.urls),
  path('major-event-log/', include(('major_event_log.urls', 'major-event-log'),
```

(continues on next page)

(continued from previous page)

```
        namespace="major-event-log"))
]
```

4. Migrate/sync the database

```
$ python manage.py migrate
```

1.2 Developing

Install the requirements

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

Note: lxml, used in the tests, requires that libxslt and libxml2 be installed on the system.

Clone the git repository:

```
$ git clone https://github.com/unt-libraries/django-major-event-log
$ cd django-major-event-log
```

Initialize the database

```
$ python manage.py migrate
```

Create a superuser so you can create test events

```
$ python manage.py createsuperuser
```

Start the development server

```
$ python manage.py runserver
```

You should now be able to access the admin portion of the app at 127.0.0.1:8000/admin/ and the public-facing side of the app at 127.0.0.1:8000/major-event-log/.

1.2.1 Testing

To run the tests in the development environment:

```
$ python manage.py test ./tests
```

You can also run the tests with Tox:

```
$ [sudo] pip install tox
$ tox
```

1.3 Model

1.3.1 Event

Fields

`id` - The primary key for each event. Uses the UUID 4 standard.

`title` - The short, descriptive title for the event.

`detail` - The longer, more detailed description of the event.

`outcome` - The event outcome. Two choices are available here: success or failure, as defined here.

`outcome_detail` - A more detailed description of the outcome of the event.

`date` - The event's date and time of occurrence.

`entry_created` - The date and time that the event was created in the database.

`entry_modified` - The date and time that the event was last modified in the database.

`contact_name` - The name of the individual or organization who created the event, or which is responsible for managing the events.

`contact_email` - The email address for that individual or organization.

Methods

`get_absolute_url()` - This method returns the absolute URL which points to the event's own details page, minus the domain.

`is_success()` - This method returns a Boolean (True or False) which indicates if the chosen value for `outcome` was success. So, if the event outcome was a success, then this method would return `True`.

CHAPTER 2

Contributors

- Gio Gottardi
- Damon Kelley
- Madhulika Bayyavarapu
- Gracie Flores-Hays