

Events Management App

Consider the following website where the events has to be posted.

Website 1 : Facebook.com.

Website 2 : Eventbrite.com.

Website 3 : eventful.com.

Website 4 : Meetup .

Website 5 : evvnt.com .

Assumptions:

We have the API keys and scripts developed accordingly for the respective website.

Programming languages used: Python 2.7 and Flask.

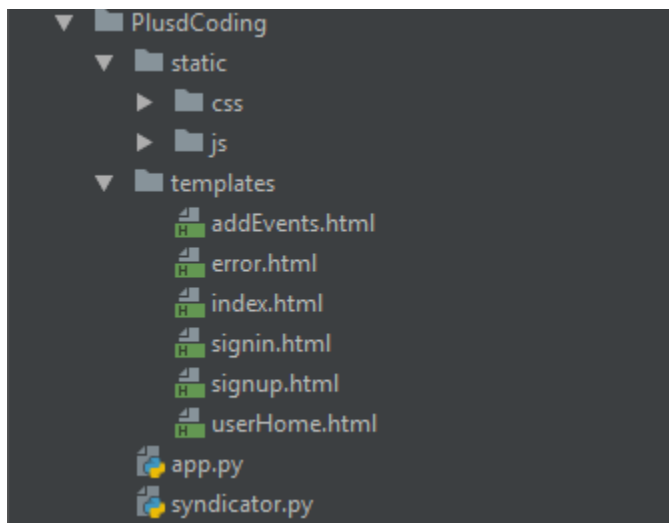
Database used: MySQL

Dev Tools: Py Charm

Setup:

First execute the DB_create.sql pointing to the respective database.

Project Directory:



Python Packages to be installed:

Please see the requirements.txt

Database Table Description

1. tbl_events_log: This contains the status of new and old events.
2. tbl_event : this contains information about all events
3. tbl_user : Table for login / signup.

Tbl_event

event_id	event_title	event_description	event_user_id	event_date	event_time	event_location
201801281932	Abhinav	Testing	5	2018-01-10	12am	Hvderbad

Tbl_events_log

event_id	status
201801281932	1
201801282049	1
201801282124	1
201801282126	1
201801282202	1
201801282204	1
201801290014	0
201801290804	0
201801290806	0

The status: 0 indicates Not Synced 1 indicated Synced

Tbl_user

User login details

user_id	user_name	user_username	user_password
1	Abhinav Gundlapalli	agundlapalli@uh.edu	Tempor@4593
2	viooi	viooi@uh.edu	Tempor@45
3	sindhur	SINDHUR@UH.EDU	tESET
4	abhinav4593	a@uh.edu	123456
5	abhinav4593	abhi@uh.edu	123456

A Stored procedure is used to create a new user each time from the app.py

App.py

This serves as the basic controller for the whole application which holds the routes for the Following pages

1. Home
2. Sigup
3. Signin
4. Logout
5. Create event/Add Event
6. User Home

Database Configurations are set here (Modify them based on your configurations):

```
# MySQL configurations
app.config['MYSQL_DATABASE_USER'] = 'root'
app.config['MYSQL_DATABASE_PASSWORD'] = 'root'
app.config['MYSQL_DATABASE_DB'] = 'plusd_db'
app.config['MYSQL_DATABASE_HOST'] = 'localhost'
app.config['MYSQL_DATABASE_PORT'] = 3306
mysql.init_app(app)
```

Cron job is initiated in app.py which will call execute the Syndicator.py file

```
# Cron job Function
def jobcron():
    execfile("syndicator.py")

sched = BackgroundScheduler(daemon=True)
sched.add_job(jobcron, 'interval', seconds=40)
sched.start()
```

The interval can be modified as required. Seconds or minutes or hours.

The functions in the App.py

```
@app.route('/')
def main():...

@app.route('/showSignUp')
def showSignUp():...

@app.route('/signIn', methods=['POST'])
def signIn():...

@app.route('/showSignIn')
def showSignIn():...

@app.route('/validateLogin', methods=['POST'])
def validateLogin():...

@app.route('/userHome')
def userHome():...

@app.route('/logout')
def logout():...

@app.route('/showAddEvents')
def showAddEvents():...

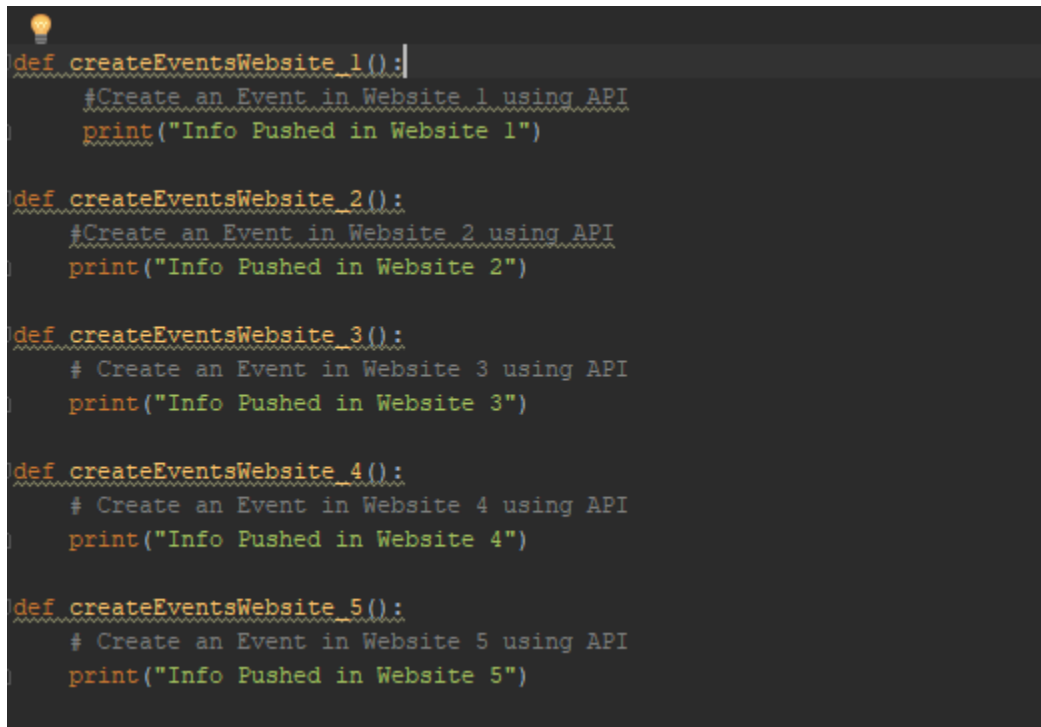
@app.route('/addEvent', methods=['POST'])
def addEvent():...

@app.route('/getEvent')
def getEvent():...
```

Sydicator.py

This file contains the functions for check in the databases for any new events and get the detailed information about that event and sync it across the websites.

The file contains 5 functions for each website.

A screenshot of a code editor with a dark background and a lightbulb icon in the top left corner. The editor displays five Python functions, each for creating an event in a specific website. The functions are: `createEventsWebsite_1()`, `createEventsWebsite_2()`, `createEventsWebsite_3()`, `createEventsWebsite_4()`, and `createEventsWebsite_5()`. Each function has a docstring indicating its purpose and a print statement that outputs a message like "Info Pushed in Website 1".

```
def createEventsWebsite_1():  
    #Create an Event in Website 1 using API  
    print("Info Pushed in Website 1")  
  
def createEventsWebsite_2():  
    #Create an Event in Website 2 using API  
    print("Info Pushed in Website 2")  
  
def createEventsWebsite_3():  
    # Create an Event in Website 3 using API  
    print("Info Pushed in Website 3")  
  
def createEventsWebsite_4():  
    # Create an Event in Website 4 using API  
    print("Info Pushed in Website 4")  
  
def createEventsWebsite_5():  
    # Create an Event in Website 5 using API  
    print("Info Pushed in Website 5")
```

The API code for respective website is injected in each of these functions and then processed.

The following functions are used to query the database.

`def getAllNewEvents()` : used to get the list of all new events and returns a list of ids.

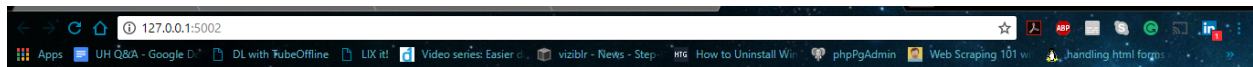
`def updateEventsLog(eventids)` : Update the processed ids post sync.

`def getEventDetails(eventids)` : get the details of each events for API Processing.

Execution order

```
list=getAllNewEvents()  
EventsList=getEventDetails(list)  
createEventsWebsite_1()  
createEventsWebsite_2()  
createEventsWebsite_3()  
createEventsWebsite_4()  
createEventsWebsite_5()  
updateEventsLog(list)  
print("Updation done")
```

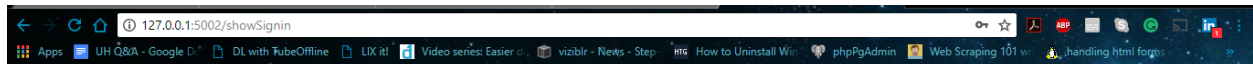
The App Home Page



Events Manager App

[Home](#) [Sign In](#) [Sign Up](#)

Sign In



Events Manager App

[Home](#) [Sign In](#) [Sign Up](#)

Sign up



Events Manager App

[Home](#) [Sign In](#) [Sign Up](#)

User Home

127.0.0.1:5002/getEvent

AppsUH Q&A - Google D...DL with TubeOfflineLIX.it!Video series: Easier v...vizibl - News - Step...How to Uninstall W...phpPgAdminWeb Scraping 101 w...handling html forms

Events Management

Add EventLogout

Event Id	Event Title	Event Desc	Event Date	Event Time	Event Location	Syndicate Status
201801281932	Abhinav	Testing	2018-01-10	12am	Hyderabad	Synced
201801282049	sfgsfghh	ggg	2018-01-17	gghrtsr	ffffff	Synced
201801282124	dafadfsgghn	ffff	2018-01-17	12am	Hyderabad	Synced
201801282202	suchi	suchi	2018-01-16	tesert	Hyderabad	Synced
201801290804	Abhinav	hello wort	2018-01-27	tesert	ffffff	Synced
201801290806	dafadfsgghn	hello wort	2018-02-10	gghrtsr	sfghdgh	Synced

Add new Event

127.0.0.1:5002/showAddEvents

AppsUH Q&A - Google D...DL with TubeOfflineLIX.it!Video series: Easier v...vizibl - News - Step...How to Uninstall W...phpPgAdminWeb Scraping 101 w...handling html forms

Create a New Event

HomeAdd EventLogout

Event Title

Enter event title

Event Description

Event Description

Event Date

mm/dd/yyyy

Event Location

Event Time

Add Event

New event not synced



Events Management

Add Event Logout

Event Id	Event Title	Event Desc	Event Date	Event Time	Event Location	Syndicate Status
201801290910	Test1	Test2	2018-01-18	12am	Hyderabad	Non Syncyed

