

nuPlanner

by Al Johri, Sai, Jerry Yang, Colin Diao, Li Wang, Xun

Wang, Sam Patti

Copyright 2013

Package controllers

controllers

Class Application

```
java.lang.Object
  |
+-Controller
  |
+-controllers.Application
```

```
public class Application
extends Controller
```

Constructors

Application

```
public Application()
```

Methods

events

```
public static Result events()
```

controllers

Class FB

```
java.lang.Object
└--controllers.FB
```

```
public final class FB
extends java.lang.Object
```

Fields

APP_ACCESS_TOKEN

```
public static final java.lang.String APP_ACCESS_TOKEN
```

Constant value: 524073037656113|11aTC3FhsPHJEeRZfWB9vk70nAk

USER_ACCESS_TOKEN

```
public static final java.lang.String USER_ACCESS_TOKEN
```

Constant value:

```
CAAHcpEMZB8DEBANQkZAyhJCXEYnAn2UW9paEZC7VWGJCb9a1ouGismLu8xKfCZCZBdtAl3Ld4iVC6u2Ha
XHqmknogy4RMuMswUMqc17PKSwAt4UmQoqDT4RIHKQaeHpnoXjXkCWTFlx7fNKKvDLUaauyS3KIuFsvziJ
aCphDuCgZDZD
```

CENTER_LATITUDE

```
public static final java.lang.String CENTER_LATITUDE
```

Constant value: 42.054581

CENTER_LONGITUDE

```
public static final java.lang.String CENTER_LONGITUDE
```

Constant value: -87.677192

CENTER_RADIUS

```
public static final java.lang.String CENTER_RADIUS
```

Constant value: 1000

FQL_EVENT_FIELDS

```
public static final java.lang.String FQL_EVENT_FIELDS
```

(continued from last page)

Constant value: `eid, name, creator, host, start_time, end_time, location, venue, description, pic, pic_big, pic_cover, pic_small, pic_square, timezone, update_time, all_members_count, attending_count, declined_count, unsure_count, not_replied_count`

FQL_EVENT_MEMBER_FIELDS

`public static final java.lang.String FQL_EVENT_MEMBER_FIELDS`

Constant value: `eid`

FQL_PLACE_FIELDS

`public static final java.lang.String FQL_PLACE_FIELDS`

Constant value: `page_id`

FQL_EVENT_TABLE

`public static final java.lang.String FQL_EVENT_TABLE`

Constant value: `event`

FQL_EVENT_MEMBER_TABLE

`public static final java.lang.String FQL_EVENT_MEMBER_TABLE`

Constant value: `event_member`

FQL_PLACE_TABLE

`public static final java.lang.String FQL_PLACE_TABLE`

Constant value: `place`

FQL_LIMIT

`public static final java.lang.String FQL_LIMIT`

Constant value: `LIMIT 51000`

FQL_ORDER

`public static final java.lang.String FQL_ORDER`

Constant value: `ORDER BY start_time ASC`

GRAPH_EVENT_FIELDS

`public static final java.lang.String GRAPH_EVENT_FIELDS`

(continued from last page)

Constant value:

`id,name,owner,description,start_time,end_time,location,venue,privacy,updated_time,
picture,ticket_uri`

GRAPH_SEARCH_TYPE

`public static final java.lang.String GRAPH_SEARCH_TYPE`Constant value: `event`

GRAPH_LIMIT

`public static final java.lang.String GRAPH_LIMIT`Constant value: `5000`

BLOCKED_ORGANIZATIONS

`public static final java.lang.String BLOCKED_ORGANIZATIONS`

BLOCKED_VENUES

`public static final java.lang.String BLOCKED_VENUES`

Constructors

FB

`public FB()`

controllers

Class Scrape

```
java.lang.Object
  |
+-Controller
  |
+-controllers.Scrape
```

```
public class Scrape
extends Controller
```

Constructors

Scrape

```
public Scrape()
```

Methods

scrape

```
public static Result scrape()
```

controllers

Class ScrapeFacebook

```
java.lang.Object
  |
  +--Controller
        |
        +--controllers.ScrapeFacebook
```

```
public class ScrapeFacebook
extends Controller
```

Constructors

ScrapeFacebook

```
public ScrapeFacebook()
```

Methods

scrape_graph

```
public static Result scrape_graph()
```

Scrapes events from Facebook using the Facebook Graph API. It searches for events radially outward from predefined center coordinates with a predefined radius.

This method requires a `USER_ACCESS_TOKEN` as opposed to the standard `APP_ACCESS_TOKEN`. Thus, this method can only be run upon user authentication. Thus, this method is currently not being used in the "scrape sequence," the sequence of functions run to scrape all events (defined in `Scrape.java`).

TODO: Enable this function after implementing Facebook User Authentication.

Returns:

Return a JSON string of all scraped events.

scrape_fql

```
public static Result scrape_fql()
```


(continued from last page)

Scrapes events from Facebook using the Facebook Query Language (FQL). It searches for events using the defined FQL_QUERY.

The FQL_QUERY searches for events that are created by Facebook Pages. It filters these events based on pages whose location is within CENTER_RADIUS meters from (CENTER_LATITUDE, CENTER_LONGITUDE).

As stated above, this query only searches for events that are created by a page. The event_member table in FQL contains uids and page_ids of, as the name suggests, members of events. The logic behind the query is that if a page is a member of an event, the page is the creator the event. Most pages have a location associated with them. If the location is within CENTER_RADIUS meters from (CENTER_LATITUDE, CENTER_LONGITUDE) the event will be returned.

However, one major pitfall is international/national organizations that have their headquarters within this radius. The page resides here, but the actual event may be in Washinton, DC or another country. Some examples of these are SAE and Rotary International. There is no obvious solution for filtering with these events, especially if they don't have a listed venue. The filter_events function in Utilities.java explores this in greater detail.

Returns:

Return a JSON string of all scraped events.

scrape_locations

```
public static Result scrape_locations()
```

Scrape Locations uses data already in the MyEvent table to search for events. It uses the getLocations function in Utilities.java to find a list of all locations where events currently reside based on the events in the database. It then uses the Facebook Graph API to perform a text based query of these locations. This is a powerful function and grabs a myriad of events. However, filtering is extremely important as the majority of these events will be outside of the city/state/country.

Returns:

Return a JSON string of all scraped events.

scrape_organizations

```
public static Result scrape_organizations()
```

[scrape_organizations description]

Returns:

[description]

controllers

Class ScrapeWildcatConnection

```
java.lang.Object
  |
+-Controller
  |
+-controllers.ScrapeWildcatConnection
```

```
public class ScrapeWildcatConnection
extends Controller
```

The ScrapeWildcatConnection class scrapes the events within WildcatConnection and stores them to the database.

Constructors

ScrapeWildcatConnection

```
public ScrapeWildcatConnection()
```

Methods

scrape_wildcatconnection

```
public static Result scrape_wildcatconnection()
```

controllers

Class Seed

```
java.lang.Object
  |
+-Controller
  |
+-controllers.Seed
```

```
public class Seed
extends Controller
```

The Seed class collects data from various sources on the web that will be saved to the database in order to aid the scrapers.

Constructors

Seed

```
public Seed()
```

Methods

seed

```
public static Result seed()
```

seed_organizations

```
public static Result seed_organizations()
```

controllers

Class Utilities

```
java.lang.Object
  |
  +--Controller
        |
        +--controllers.Utilities
```

```
public class Utilities
extends Controller
```

The Seed class collects data from various sources on the web that will be saved to the database in order to aid the scrapers.

Constructors

Utilities

```
public Utilities()
```

Methods

createEvent

```
public static MyEvent createEvent(com.restfb.json.JsonObject event)
```

getLocations

```
public static java.util.ArrayList getLocations()
```

saveOrUpdate

```
public static boolean saveOrUpdate(MyEvent event)
```

filter_event

```
public static boolean filter_event(MyEvent event)
```

save_events

```
public static java.util.List save_events(java.util.List events)
```

save_events

```
public static java.util.List save_events(com.restfb.json.JsonArray events)
```

facebook_fql_query

```
public static java.util.List facebook_fql_query(java.lang.String token,  
        java.lang.String query)
```

facebook_graph_query

```
public static com.restfb.json.JsonObject facebook_graph_query(java.lang.String token,  
        java.lang.String q,  
        java.lang.String type,  
        java.lang.String fields,  
        java.lang.String center,  
        java.lang.String distance,  
        java.lang.String limit)
```

Package models

models

Class MyEvent

```
java.lang.Object
  |
+-Model
  |
+-models.MyEvent
```

```
public class MyEvent
extends Model
```

Fields

eid

```
public java.lang.Long eid
```

name

```
public java.lang.String name
```

creator

```
public java.lang.String creator
```

location

```
public java.lang.String location
```

venue

```
public java.lang.String venue
```

start_time

```
public DateTime start_time
```

end_time

```
public DateTime end_time
```

(continued from last page)

pic

```
public java.lang.String pic
```

description

```
public java.lang.String description
```

find

```
public static <any> find
```

findLong

```
public static <any> findLong
```

findDate

```
public static <any> findDate
```

Constructors

MyEvent

```
public MyEvent(java.lang.Long eid,  
               java.lang.String name,  
               java.lang.String creator,  
               DateTime start_time,  
               DateTime end_time,  
               java.lang.String location,  
               java.lang.String venue,  
               java.lang.String description,  
               java.lang.String pic)
```

Methods

toJSONString

```
public java.lang.String toJSONString()
```


models

Class MyOrganization

```
java.lang.Object
  |
+-Model
  |
+-models.MyOrganization
```

```
public class MyOrganization
extends Model
```

Fields

fbid

```
public java.lang.Long fbid
```

name

```
public java.lang.String name
```

creator

```
public java.lang.String creator
```

url

```
public java.lang.String url
```

location

```
public java.lang.String location
```

description

```
public java.lang.String description
```

find

```
public static <any> find
```

(continued from last page)

findLong

```
public static <any> findLong
```

Constructors

MyOrganization

```
public MyOrganization(java.lang.Long fbid,  
                      java.lang.String name,  
                      java.lang.String location,  
                      java.lang.String url,  
                      java.lang.String description)
```

Methods

getJsonObject

```
public java.lang.String getJsonObject()
```

models

Class Person

```
java.lang.Object
  |
  +--Model
        |
        +--models.Person
```

Direct Known Subclasses:
[Student](#)

```
public class Person
extends Model
```

Fields

id

```
public java.lang.Long id
```

name

```
public java.lang.String name
```

Constructors

Person

```
public Person(java.lang.String name)
```

models

Class Student

```
java.lang.Object
├--Model
│   └--models.Person
│       └--models.Student
```

```
public class Student
extends Person
```

Fields

find

```
public static <any> find
```

Constructors

Student

```
public Student(java.lang.String name)
```

Index

A

APP_ACCESS_TOKEN 4

Application 3

B

BLOCKED_ORGANIZATIONS 6

BLOCKED_VENUES 6

C

CENTER_LATITUDE 4

CENTER_LONGITUDE 4

CENTER_RADIUS 4

createEvent 12

creator 15, 17

D

description 16, 17

E

eid 15

end_time 15

events 3

F

facebook_fql_query 13

facebook_graph_query 13

FB 6

fbid 17

filter_event 12

find 16, 17, 20

findDate 16

findLong 16, 18

FQL_EVENT_FIELDS 4

FQL_EVENT_MEMBER_FIELDS 5

FQL_EVENT_MEMBER_TABLE 5

FQL_EVENT_TABLE 5

FQL_LIMIT 5

FQL_ORDER 5

FQL_PLACE_FIELDS 5

FQL_PLACE_TABLE 5

G

getJsonObject 18

getLocations 12

GRAPH_EVENT_FIELDS 5

GRAPH_LIMIT 6

GRAPH_SEARCH_TYPE 6

I

id 19

L

location 15, 17

M

MyEvent 16

MyOrganization 18

N

name 15, 17, 19

P

Person 19

pic 16

S

save_events 12, 13

saveOrUpdate 12

Scrape 7

scrape 7

scrape_fql 8

scrape_graph 8

scrape_locations 9

scrape_organizations 9

scrape_wildcatconnection 10

ScrapeFacebook 8
ScrapeWildcatConnection 10
Seed 11
seed 11
seed_organizations 11
start_time 15
Student 20

T

toJSONString 16

U

url 17
USER_ACCESS_TOKEN 4
Utilities 12

V

venue 15